Freie Universität Berlin

Department of Mathematics and Computer Science
Institute of Computer Science

Doctoral Dissertation

Object Trust: From Trustworthy Hosts to Trustworthy Data

Dissertation zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.) am
Fachbereich Mathematik und Informatik der Freien Universität Berlin

vorgelegt von

M.Sc. Pouyan Fotouhi Tehrani

Berlin 2024

verteidigt am 28. März 2025

First Examiner
Prof. Dr.-Ing. Jochen Schiller
Freie Universität Berlin

Second Examiner
Prof. Dr. Ulrich Meissen
Hochschule für Technik und Wirtschaft Berlin

Abstract

We rely on the Internet as a ubiquitous communications network for critical functions ranging from financial transactions to emergency and crisis alerts. Organizational, policy, and technical efforts have been made to enable users to assess the trustworthiness of the hosts with which they communicate. However, this host-centric approach to trust fails if a network path to the host in question cannot be established. Our research aims to understand trust, decouple it from hosts, and bind it to the data itself to achieve object trust, which ensures trustworthy data exchange even when the source is unavailable.

In this thesis, construct a trust framework for the Web, measure the effectiveness and deployment of security and trust mechanisms on the Web, and assess the suitability of these mechanisms for information-centric networks. Our contributions have operational implications for today's Internet and pave the way for the integration of trust mechanisms into other network paradigms.

Zusammenfassung

Als allgegenwärtiges Kommunikationsnetz vertrauen wir dem Internet kritische Aufgaben an, sei es bei der Abwicklung von Finanztransaktionen oder bei der Alarmierung in Notfällen und Krisen. Dazu wurden organisatorische, politische und technische Anstrengungen unternommen, um die Nutzer in die Lage zu versetzen, die Vertrauenswürdigkeit der Hosts, mit denen sie kommunizieren, zu beurteilen. Dieser host-zentrierte Ansatz scheitert jedoch, wenn kein Netzwerkpfad zu dem betreffenden Host aufgebaut werden kann. Unsere Forschung zielt darauf ab, Vertrauen zu verstehen, es von den Hosts zu entkoppeln und an die Daten selbst zu binden, um Objektvertrauen (object trust) zu erreichen, das einen vertrauenswürdigen Datenaustausch gewährleistet, selbst wenn der Urheber der Daten nicht verfügbar ist.

In dieser Arbeit konstruieren wir ein Framework für vertrauenswürdige Kommunikation über das Web, messen wir die Wirksamkeit und den Einsatz von Sicherheits- und Vertrauensmechanismen im Web, und bewerten die Eignung dieser Mechanismen für Information-centric Networks. Unsere Beiträge haben einen operativen Einfluss auf das heutige Internet und ebnen den Weg für die Integration von Vertrauensmechanismen in andere Netzwerkparadigmen.

Statement of Integrity

I hereby declare to the Freie Universität Berlin that I have prepared this dissertation independently and have done so without using sources or aids other than those indicated. This work is free of plagiarism. All excerpts taken verbatim or in content from other publications have been clearly identified as such. This dissertation has not been submitted in the same or similar form to any previous doctoral program. I agree to have my work checked by plagiarism detection software.

Selbstständigkeitserklärung

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die Arbeit ist frei von Plagiaten. Alle Stellen, die anderen Werken wörtlich oder inhaltlich entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch nicht in einem früheren Promotionsverfahren vorgelegt. Ich erkläre mich mit einer Überprüfung meiner Dissertation durch ein Plagiatserkennungsprogramm einverstanden.

Berlin, den 17. Dezember 2024, Pouyan Fotouhi Tehrani

Acknowledgments

I wish to express my deepest gratitude to all those who supported and guided me throughout the journey of completing this thesis.

First and foremost, I would like to thank my advisors for their invaluable guidance and constructive feedback. Your expertise and mentorship have been a cornerstone in shaping this work. **Matthias**, thank you for your unwavering encouragement over the years, it was with you that I learned the fundamentals of sound research. **Thomas**, I have always appreciated your support, which has more than often gone beyond mere research and academics. **Eric**, I am indebted to you for letting me be a part of your research, introducing me to critical working groups and showing me how to present my research. **Jochen**, thank you for giving me the opportunity to start my doctoral journey when I thought that my academic career was over. **Ulli**, please know that I am and always have been deeply grateful to you for being a mentor and a friend who has supported me personally, professionally and throughout my academic journey. **Frank**, I appreciate and will never forget your help in publishing my first paper outside the university, which opened the door to my dissertation. Dear **Professor Hoppe**, I truly value the opportunity you gave me to enter the world of research and academia, even though I did not have much to offer at the time.

Special thanks go to my colleagues and peers at the Chair of Distributed and Networked Systems at the Technische Universität Dresden, and the Chair of Computer Systems and Telematics at the Freie Universität Berlin for their collaboration, stimulating discussions, and moral support. Their camaraderie made this experience both intellectually enriching and enjoyable. **Cenk**, your eye for aesthetics taught me that good content is always accompanied by good visualization. **Marcin**, your precision and goal orientation always helped me to get out of spirals of chaos. **Raphael**, the way you structured the work, went through the RFCs thoroughly and presented the results accurately is an inspiration I'll take with me for the rest of my career. **Jonas**, thank you for reviewing my work, your objective assessment and feedback has always been invaluable. **Maynard**, your moral support, especially as this work was coming to an end, helped me over the last mile, and I am really grateful for that. Last but not least, **Peter**, thank you for keeping me sane.

I am also deeply grateful to my family for their unconditional love and patience throughout this process. **Katharina**, I can never repay you for the strength you gave me during my dissertation. All my love goes to you, **Minou**, and **Ava**, my daughters. Your belief in me has been my greatest source of strength. My parents, **Hossein** and **Sedigheh**, and my siblings, **Nima** and **Nastaran**, made it possible for me to be who I am and where I am now. And finally, my sincere appreciation to **Arsalan**, **Shabnam**, **Fabio**, **Iris**, **Paula**, and all my dearest friends who have unconditionally supported me to get where I am today.

„So eine Arbeit wird eigentlich nie fertig, man muß sie für fertig erklären, wenn man nach Zeit und Umständen das Möglichste getan hat.“

– Johann Wolfgang von Goethe

Inhaltsverzeichnis

Introduction

This thesis lays the groundwork for trustworthy data exchange when the data source in unavailable, *i.e.,* in face of intermittent connections, unreliable links, or fragmentation. We first motivate and provide a context for this work through a brief background. We then draw a research road map to addresses relevant aspects to reach our goal. Three milestones set overarching objectives. To reach each objective, we define smaller targets, put each target in context of related work and highlight uncharted research territories through concrete research questions. The remainder of this work reflects a similar structure with each part corresponding to a research endeavor and chapters tackling subordinate targets. The summary of our main contributions, at the end of this chapter, provides an overview of our findings and insights from each chapter.

Trustworthy Data Exchange

The Internet is arguably the most widely used and accessible communications network of our time. We use the Web for basic tasks like online shopping, but we also trust it for essential purposes like emergency communications. Our trust in the Internet is based on its transcendence into the real world. In fact, many technical aspects of the Internet are closely related to real-world attributes and entities. Most commonly, we use domain names that resemble real names (*e.g.,* trademarks), and we view Web certificates as identification documents that can be used to authenticate a real person or organization. With such identifying information linking an online entity to its real-world counterpart, we can appropriately assess its trustworthiness based on our preexisting real-world trust relations.

The first identity cue about the endpoint we are trying to connect to is its domain name. We resolve the domain name to an IP address and initiate a secure connection using Transport Layer Security (TLS). During the TLS handshake, the endpoint provides the second identity cue in the form of a Web certificate. The certificate authenticates the endpoint, *i.e.,* proves that we have reached the correct host for the given domain name, and may contain additional identifying information such as the owner's name, address, *etc.*. At this point, we can identify the communication partner and set our trust level accordingly. Cryptographic keys are then negotiated, and the communication channel is secured for subsequent data transmission.

In the model described above, the building blocks of online trust, *e.g.,* the identifying information, are associated with an endpoint, a host, where the data resides. We then consider the data itself to be trustworthy *because* we have authenticated the host and established a secure data channel with it. This approach, however, fails to deliver if we are unable to contact the host, for example, due to intermittent connectivity, high latency, low bandwidth, partial isolation, or any other issue that challenges end-to-end connectivity. To address shortcomings of host-based trust, we set the stage in this thesis for what we call *object trust* by extending the concept of object security.

Research Road Map for Object Trust

Our goal in this thesis is to directly associate measures of trust with data and achieve object trust. The idea is to decouple data from its producer, in both temporal and spatial dimensions, while maintaining security and trustworthiness. For this we unify notion of trust from the Web with the concept of object security. And as the foundation model for our object security we rely on Information-Centric Networking (ICN). In ICN, data producers secure data packets directly, allowing data to be stored and transmitted independently of any particular network host. We extend object security with a focus on trustworthiness. The basic premise is to make data a first-class citizen, embedding everything needed to evaluate its trustworthiness, regardless of its producer and host.

Adapting trust mechanisms from the host-based Internet approach to information-centric networks calls for research on three dimensions. We need to (*i*) understand what amounts to trusted and trustworthy communication, (*ii*) comprehend trust means on the Internet and measure their efficacy, and (*iii*) assess suitability of information-centric approaches in integrating these means. In the following, we first elaborate on each aspect and then relate them to the existing literature, highlight blind spots in research, and formulate concrete research questions to achieve our goal.

Trusted Communication. Trust is a human condition. It is challenging to translate this rather subjective state into technical means. In communication networks, trust and trustworthiness (commonly used interchangeably) are evidence-based. For example, a digital signature allows authentication of data origin and gives the recipient that chance to assess the trustworthiness of data. Not the cryptographic primitives, *e.g.,* the signature itself, is relevant for trust relations but its association with the real-world, *e.g.,* the owner of that signature. So catering for trustworthy communication first requires understanding how trust relations are formed among humans and how can we realize relevant attributes from the non-technical realm using technical means.

Trust on the Web. The need for trusted communication on the Internet, or more specifically, the Web, was recognized early on, especially as more services such as e-commerce were delivered online. Among other things, domain names were recognized as intellectual property, the DNS was secured with DNSSEC, and the Web PKI was established to provide authentication. Thus, the foundation of trust was based on authenticating hosts, establishing secure connections, and, in turn, establishing key trust relations based on the fact that the host is known and the communication channel is secure. The effectiveness of these efforts depends on their deployment and actual use. While each approach has been studied in isolation, a comprehensive understanding of the entire ecosystem is lacking. This is also a prerequisite for identifying shortcomings and practical obstacles before adapting to alternative networking paradigms such as ICN.

Object Trust. The current paradigm of trust on the Internet revolves around hosts. In many cases, however, we are more interested in the data itself and less invested in where it comes from. Email security, Certificate Revocation Lists (CRL), and the like are approaches that assume insecure communication channels and instead secure the data so that it can be delivered by any network host over insecure channels. Such an approach to object security is a well-known concept that secures the data rather than the communication channel by embedding security measures. However, security alone does not guarantee trustworthiness. We introduce the concept of *object trust* as an extension of object security that also binds information relevant for trust with the data. To do this, we use ICN, which already implements object security. While ICN security measures do not depend on end-to-end connectivity, it lacks the non-technical aspect, the human core, that makes up trust. Since ICN shares the same cryptographic primitives with DNSSEC, Web PKI, and other relevant aspects of the Internet trust ecosystem, our goal is to conceptualize an approach that consolidates the well-established organizational efforts of the Internet with the data-centric paradigm of ICN.

Research Endeavor 1: Modeling Trust for Online Communication

When we talk about trust online, it refers to the user perception of Web services. A common understanding is that security is an objective attribute of trustworthiness [26], [29], [75], [129], [406], [180]. For example, a given service is regarded trustworthy, as we can prove that the data has remained intact during transmission, or the domain name has not been spoofed.

Reducing trust to security, however, fails to grasp the way humans establish trust relations. So, various studies investigate the impact of anthropic factors, such as familiarity, reputation, and alike [27], [49], [442], [216], [218], [254]. Integrating these factors would then enable users to assess trustworthiness on their own.

The inertia of the Internet as a multi-stakeholder, complex sociotechnical system [351], however, makes it difficult to introduce new processes that enjoy widespread adoption. The challenge that we face here is to use established infrastructure and existing methods in catering for a trustworthy communication. We pursue this on two grounds by (*i*) conceptualizing an integrative trust model, and (*ii*) adapting that trust model for the Web.

Concept of an interdisciplinary model of trust for digital communication

Trust is primarily a subjective perception that is formed within a social context [309], [395], [209], [216], [246], [257]. Translating trust into technical means is rather non-trivial. The first step is to understand how people perceive trust, both psychologically and socially. That is to go beyond trust models based merely on technical attributes [129], [406], [207], *e.g.,* security and correctness, and consider models that account for the human factor [49], [218], [474]. The next step is to contemplate on how the medium of digital communication affects trust relations. This opens up the following research questions:

Research Questions
❓ How can factors impacting trust relations be captured in an integrative model?
❓ What are the practical limits in applying such model?

Addressing the first question requires an investigation of trust beyond the technical community and into the realms of psychology and sociology. Our goal is to develop a model of trust that integrates personal and societal aspects of trust with technical characteristics. By conceptualizing trust as a multidimensional space, we can show how online service providers can move within this space to enforce trust relations for their digital communications. The second question is dedicated to the practical applicability of this model. That is, how can this model be applied to online communication and what are its limitations, especially with regard to the non-technical aspects.

Construct a trust framework for the Web

Studies of trust on the Web typically either focus on non-technical aspects such as design and structure [49], [218], [254] or conflate trustworthiness with security [29], [406], [180]. Both views have pitfalls when it comes to trust. Non-technical aspects can be imitated without being detected and are therefore less effective, and mere security measures are insufficient for trust relations because anyone can build and deploy them, as numerous spoofing campaigns in the past have shown. We want to take a practical approach that considers both aspects by putting security in context and relating it to the real world, specifically using existing tools and infrastructures.

Relevant to our investigation are feature that represents real-world properties, and protocols that secure them. The most common and widely available approaches for the Web are the Web PKI, TLS, DNSSEC, and related security protocols that build on DNS(SEC), namely DANE and CAA. The Web PKI evolves around X.509 certificates and authentication using digital signatures [297]. Transport Layer Security (TLS) provides integrity and confidentiality [439], [440]. And DNSSEC [444]–[446] secures the global domain namespace that is the starting point for virtually every online transaction. CAA and DANE link domain names and the Web PKI. Certification Authority Authentication (CAA) [357] allows name owners to select CAs authorized to certify names from their namespace, and DANE [363] allows users to signal which certificates to expect for a given service at a given domain name. So, to build a trust framework using these technologies, we need to address the following questions

Research Questions
❓ Which features from Web PKI, DNS, and related technologies amount to trust?
❓ How can security and trust be captured algorithmically?

The concern of the first research question is how the technologies discussed above interrelate and what their mutual impact on trust is. Focusing on features that are under the control of endpoints, we consider every step involved in Internet communication, from resolving a domain name to establishing a secure connection and exchanging data. We extract those features that play a role in authentication, integrity, and confidentiality. In a further step, we consider how trust can be expressed in quantifiable and qualifiable security assurances. This allows us to compare security deployments and provide users with an understandable assessment of trustworthiness without going into technical details.

Research Endeavor 2: Assessing Trust Measures on the Web

Service providers secure their infrastructure as they see fit. We want to understand how different trust measures are deployed and configured on the Web. Understanding actual deployment configurations and trends is also useful for identifying shortcomings and pitfalls in existing approaches. We do this through active measurement.

We start with DNS, the starting point for almost all web-based transactions. Domain names bring familiarity to the Internet and are known to contribute both positively and negatively to trust relations [121], [201], [205], [206], [251]. However, plain DNS is vulnerable to spoofing and manipulation [288], [156], [157], [255]. We focus on how service providers secure their names with DNSSEC and how well cryptographic hygiene is maintained.

Next, we look at Web certificates and the Web PKI as a broader context. Certificates enable authentication at various levels, from the domain name owner to the real-world entity behind a server. The latter in particular allows users to adjust their trust relations based on their real-world experiences. Misissued or illegitimate certificates jeopardize authentication on the Internet [10], [25], [30], [405], [217], so security measures such as CT logs, DANE, and CAA have been proposed. We are interested in understanding how each technology is used on its own and how it interacts with the others.

Monitor and validate DNSSEC key transitions

As part of security hygiene, cryptographic material need to be renewed regularly. Renewing cryptographic keys in a distributed system such as DNS is an intricate task, specially while maintaining high availability requirements and considering omnipresent caches. DNSSEC key rollovers are not a new topic to the research [45], [97], [98], [106], [173], [187], [256], however, existing literature on key transitions is either limited in scope, *e.g.,* focuses only on the root zone [174], [203], or imposes restrictive assumptions, *e.g.,* on the number of involved keys [45], [173]. By aligning our measurement method to the deployment practices in the wild, we aim to get a better picture of DNSSEC key transitions and answer the following questions:

Research Questions
❓ How can key life cycle and transitions be formally characterized and classified?
❓ How can discrete points be fitted into this model?

As RFC guidelines and existing literature commonly model DNSSEC key transition under oversimplified assumption, *e.g.,* only two keys being involved in a transition, we first need to establish a model that is capable of capturing complex key management and transitions in the wild. The advantage of such model is twofold. First, it can be used to evaluate security implications, and second, it can be used to compare and evaluate a given implementation with guidelines such as RFCs. The second research question addresses the challenge we face when applying the aforementioned model to measured data, *i.e.,* creating a continuous temporal model from discrete measurements.

Defining a framework for systematic measurement of TLS and Web PKI

Despite extensive measurement studies, there is no common consensus on how to measure Web PKI and TLS and subsequently interpret the data, so similar studies reach different or even contradictory conclusions about the state of Web PKI or TLS. For example, measurement studies by Chung *et al.* [42] and Oakes *et al.* [184] both investigate invalid certificates in the wild, but reach to contradictory results. Before conducting our own measurement of TLS and Web PKI, we set out to answer the following questions:

Research Questions
❓ What are essential measurable features of TLS and Web PKI?
❓ How do these characteristics affect each other and what are the confounders?

Our goal is to establish a systematic framework for TLS and Web PKI measurements by conducting a thorough review of related work. This should enable comparison of different studies and provide a basis for future measurement studies. In addition, we are interested in confounding factors that may bias the interpretation of the data, in order to answer why similar studies may reach different conclusions.

Assess how Web security measures interlink in practice

In addition to domain name resolution security, the transactional and authentication security provided by TLS and the Web PKI, respectively, amount to trust relations. In addition, protocols such as CAA and DANE provide protection against illegitimate certificates.

Related work study X.509 certificates in general [53], [67], [111], with a focus on validity and revocation status [10], [42], [73], [149], the trustworthiness of CAs [37], [71], [108], or DNS-related protocols such as DANE [380], [139], [140], [491] and CAA [213]. Based on insights from related work and previous research efforts, in particular our integrated trust model, we are interested in the interplay in deployments in the wild in the context of the following research questions:

Research Questions
❓ How DNS(SEC) and Web PKI ecosystems are utilized in trust-critical scenarios?
❓ Is the Web PKI ecosystem adequately protected against certificate misuse?

From related work, we already have a general overview that reveals an encompassing reliance on the Web PKI for authentication purposes using X.509 certificates, and a comparatively low penetration rate of DNS-related protocols, *i.e.,* DNSSEC, DANE, and CAA. We first want to see if these insights still hold, specifically using our own measurement methods. Not only each technology on its own, but their interplay is integral in assessing the overall security and impact on trustworthiness. For example, in case of the Web PKI, how CAA, DANE, and CT logs interlink in real deployments against misissuance and misissued certificates. In addition to the global view, we are also interested in scenarios where trustworthy communication is essential, for example, in emergency management, online financial services and alike.

The second research question aims to understand why specific technologies are used only reluctantly. The comparative analysis above lays the groundwork. However, a close look at the specifications, dependencies, and actual deployments is required to cover both technical and non-technical aspects that might hamper popular adaption.

Research Endeavor 3: Decoupling Trust Measures from the Infrastructure

On the Web, data is bound to a specific host, identified by an IP address. Both data retrieval and its security (*i.e.,* through TLS) require an end-to-end connection. Without a network path, trusted data exchange is not guaranteed. An alternative to this host-oriented paradigm is proposed by Information-Centric Networking (ICN), which names data instead of hosts. ICN decouples data both spatially, *i.e.,* from its host, and temporally, *i.e.,* from the availability of its origin, while introducing its own object security model. We are interested in (*i*) knowing its sustainability for practical use, and (*ii*) the possibility of adapting existing trust-enforcing approaches to ICN.

Assess ICN sustainability

The basic premise of ICN, *i.e.,* naming data instead of hosts, is decades old [311], [229]. However, ICN has not gained enough momentum to become a viable alternative to IP networking. To identify possible causes, related works have analyzed ICN with a focus on tussles among potential stakeholders and non-technical aspects (social, economic, *etc.*) [61], [382], [223], [244]. We take a comparative approach to assessing the viability and sustainability of the ICN and raise the following questions:

Research Questions
❓ Can ICN be compared to the Internet as an ecosystem?
❓ How ICN names can achieve the same value as domain names?

The first question positions ICN in relation to the Internet as a complex sociotechnical ecosystem, to understand what we can learn from success of the Internet beyond mere technical aspects. We then set the focus on names and namespace management at the core of ICN and inquire into using DNS(SEC) in ICN.

Bringing Trust from the Web to the ICN

On the Web, trust mechanisms are tied to hosts. With the exception of host-based security, which is replaced by object security in ICN, naming and authentication retain their relevance for trust relations through ICN. The challenge is to integrate domain names and X.509 certificates in ICN to take advantage of the non-technical aspects of DNS and Web PKI that make Web-based communication trustworthy. However, approaches in ICN that address naming and authentication often reinvent the wheel by proposing parallel methods, procedures, and even infrastructures [4], [105], [155], [486], [262], [268]. This situation leads us to the following research questions:

Research Questions
❓ What are existing ICN approaches to naming and authentication?
❓ How existing Web trust solutions can be integrated into ICN?

We first survey existing ICN approaches to namespace management and key management. Doing so, we aim to detect naming and authentication characteristics that are specific to ICN and do not have a one-to-one counterpart on the Internet. We then apply insights from our previous research endeavors in an effort to integrate DNS(SEC) and the Web PKI certificates into ICN.

Key Contributions

We address the research question proposed above through a series of publications that comprise the upcoming chapters of this thesis. Tabelle 1 provides an overview. In the following, we provide a brief overview of our main contributions.

An Integrative Trust Framework. In Kapitel 1 we review seminal work on trust and trustworthiness in the fields of psychology, sociology and computer science [80] and proposed a framework for trustworthy digital communication. Our framework encompasses trust in digital communication in

Tabelle 1: An overview of the structure of this thesis and research questions addressed in each chapter

Teil I: Conceptualizing a Trust Framework for the Web		
Kapitel 1	❓ How can factors impacting trust relations be captured in an integrative model? ❓ What are the practical limits in applying such model?	[84] [80]
Kapitel 2	❓ Which features from Web PKI, DNS, and related technologies amount to trust? ❓ How can security and trust be captured algorithmically?	[84][86]
Teil II: Conceptualizing a Trust Framework for the Web		
Kapitel 3	❓ How can key life cycle and transitions be formally characterized and classified? ❓ How can discrete points be fitted into this model?	[186]
Kapitel 4	❓ What are essential features of TLS and Web PKI measurements? ❓ How do these characteristics affect each other and what are the confounders?	[342]
Kapitel 5 Kapitel 6 Kapitel 7	❓ How DNS(SEC) and Web PKI ecosystems are utilized in trust-critical scenarios? ❓ Is the Web PKI ecosystem adequately protected against certificate misuse?	[84][86][79][341]
Teil III: Decoupling Trust Measures from the Infrastructure		
Kapitel 8 Kapitel 9	❓ Can ICN be compared to the Internet as an ecosystem? ❓ How ICN names can achieve the same value as domain names?	[87][83]
Kapitel 10 Kapitel 11	❓ What are existing ICN approaches to naming and authentication? ❓ How existing Web trust solutions can be integrated into ICN?	[85][82]

terms of *context* and *evidence* and guides service providers on which factors (online and offline) are relevant for trustworthy communication and how they can be used to maintain trust relations.

Algorithmic quantification and qualification of Web Security and Trust. In Kapitel 2, we develop a method to define trustworthiness in context of Web communication with a focus on critical services [86]. We anchor trust on three dimensions of (*i*) identifiers and names, (*ii*) authentication and identification, and (*iii*) transaction, while considering both technical and non-technical (organizational, political, *etc.*) aspects. We show how these dimensions can be realized using accessible and widespread technologies, *i.e.,* DNS(SEC), Web PKI, and TLS. Finally, we introduced a method to algorithmically quantify and qualify these security assurances.

A Temporal Model for DNSSEC Key Life Cycle and Transition. In Kapitel 3, we develop a novel model for DNSSEC key life cycles and transitions that is capable of capturing real-world transitions beyond the theoretical one-to-one key rollovers postulated by RFCs and related work [186]. We apply this method to 15 years of DNSSEC data and analyze key life cycle management practices as well as key rollover procedures and their security implications. For our analysis, we develop a novel method capable of reconstructing complete key life cycles with high accuracy using discrete measurement data points. We find that key management errors have decreased over the years, while a significant proportion of key rollovers still fail to comply with RFC specifications. To gain a better understanding, we apply our own holistic ‘behavior-based’ classification and observe a clear trend towards key rollovers, which we call 'cut-over'. Both our data corpus and our analysis cover aspects that go beyond related work.

A Framework for TLS and Web PKI Measurements. In Kapitel 4, we propose a survey-based and data-driven systematic approach to TLS and Web PKI measurements [342]. Our approach is based on our own survey of 36 measurement studies over the past decade. We focused on the measurement and analysis methods. The contribution is twofold: (*i*), we have created a basis for making measurement studies comparable, and (*ii*), we have provided a starting point for future measurements. We have used our system to classify related work and to explain the reasons for discrepancies in similar studies.

Analysis of DNS(SEC) and Web PKI deployment on a global scale and in trust-critical scenarios. In chapters 5 and 6, we measure and analyze deployment of Web PKI, DNSSEC, CAA, and DANE based on insights from aforementioned contributions. Our findings in Kapitel 5 reveal that measures relevant to securing certificate issuance are not widespread [79]. While around half of 4M analyzed domains support either CAA or DNSSEC, only about 7% support both. DANE is barely used ($\approx$ 1%) and is plagued by misconfigurations. We also detect shortcomings in practices of CAs regarding deployment of CAA. Overall we observe that measures against certificate misissuance are not widely deployed. In Kapitel 6 we apply our security model on critical infrastructure using examples of US Alerting Authorities, German emergency management organizations, and foreign representation of UN member states [86]. The findings are aligned with deployment trends on the global scale, yet we find that identification, an integral requirement of trustworthiness, is widely neglected due to shared certificates and a constant trend away from extended validation certificates. In Kapitel 7 we show deficiencies in CAA specifications and deployment. We identify design flaws, such as implicit semantics, loose policy scooping, and lack of verifiability, as well as temporal constraints that undermine CAA effectiveness in mitigating risk of misissuance. Using a novel approach based on CT logs we also analyze CAA deployments and discover faulty implementations by a prominent CA. Insights from this work lay the basis for design of future DNS-based security protocols.

Sustainability Assessment of Information-Centric Networks. In chapters 8 and 9 we discuss fundamental economic, societal, and political aspects of a potential ICN Internet, with an emphasis on the role of non-technical issues in the acceptance and deployment of ICN on a global scale [83], [87]. In Kapitel 8, we highlight that power structures, trust, accountability, and privacy in the ICN are fundamentally different from their counterparts in the Internet. Most importantly, we point out the lack of organizational, political, and other non-technical aspects that prevent the ICN from reaching a state comparable to the Internet or the Web. In particular, in Kapitel 9 we show how naming and namespace management, which are at the core of the ICN, require organizational and political efforts to achieve practical values similar to domain names [83]. We propose a solution based on DNSSEC that integrates ICN namespace management into the DNS ecosystem, taking advantage of the established legal, organizational, and political aspects of the DNS.

Trustworthy Communication using ICN. In Chapters 10 and 11 we show how in practice Web technologies relevant to our trust model can be brought into ICN. First, we survey and evaluate key and namespace management of over 30 proposed approaches for ICN in Kapitel 10. We find that none of the namespace management solutions are designed for global namespaces, and the proposed key management solutions require an organizational infrastructure that does not yet exist [85]. Therefore, in Kapitel 11 we propose a concept that leverages the existing ecosystems of DNS and Web PKI in ICN. Our solution uses ICN to enable communication in challenged networks, while integrating DNS and Web PKI to provide assurances that amount to trust relationships [82].

How to Read This Thesis

This thesis is divided into three parts that reflect the aforementioned research efforts: (*i*) modeling trust, (*ii*) trust on the Web, and (*iii*) object trust. Each part is divided into several chapters, each dedicated to a set of research questions. The chapters are self-contained and can be studied independently. The only exception is Kapitel 6, which is actually the continuation of Kapitel 2. Note also that these chapters, along with Kapitel 8, are extended versions of the corresponding publications (see Tabelle 2).

Tabelle 2: Research publications that comprise this thesis, sorted by publication date

[83]	P. Fotouhi Tehrani, E. Osterweil, J. H. Schiller, T. C. Schmidt und M. Wählisch, "The Missing Piece: On Namespace Management in NDN and How DNSSEC Might Help," in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, Ser. ICN '19, Macau, China: ACM, Sep. 2019	Kapitel 9
[82]	P. Fotouhi Tehrani, E. Osterweil, J. H. Schiller, T. C. Schmidt und M. Wahlisch, "Authenticated Communication in Crises: Toward an Infrastructureless Trust Model for Challenged Networks," in *2019 International Conference on Information and Communication Technologies for Disaster Management*, Ser. ICT-DM '19, Paris, France: IEEE, Dez. 2019	Kapitel 11
[80]	P. Fotouhi Tehrani, N. von Kalckreuth und S. Lamprecht, "Toward an Integrative Model of Trust for Digital Emergency Communication," in *ISCRAM 2020 Conference Proceedings–17th International Conference on Information Systems for Crisis Response and Management*, Ser. ISCRAM '20, Mai 2020, S. 1012–1021	Kapitel 1
[87]	P. Fotouhi Tehrani, J. H. Schiller, T. C. Schmidt und M. Wählisch, "On economic, societal, and political aspects in ICN," in *Proceedings of the 7th ACM Conference on Information-Centric Networking*, Ser. ICN '20, ACM, Sep. 2020	Kapitel 8
[84]	P. Fotouhi Tehrani, E. Osterweil, J. H. Schiller, T. C. Schmidt und M. Wählisch, "Security of Alerting Authorities in the WWW: Measuring Namespaces, DNSSEC, and Web PKI," in *Proceedings of the Web Conference 2021*, Ser. WWW '21, Virtual Event, Ljubljana, Slovenia: ACM, Apr. 2021	Kapitel 2 Kapitel 6
[85]	P. Fotouhi Tehrani, E. Osterweil, T. C. Schmidt und M. Wählisch, "SoK: Public key and namespace management in NDN," in *Proceedings of the 9th ACM Conference on Information-Centric Networking*, Ser. ICN '22, New York, NY, USA: ACM, Sep. 2022	Kapitel 10
[186]	E. Osterweil, P. Fotouhi Tehrani, T. C. Schmidt und M. Wahlisch, "From the Beginning: Key Transitions in the First 15 Years of DNSSEC," *IEEE Transactions on Network and Service Management*, Jg. 19, Nr. 4, S. 5265–5283, Dez. 2022, ISSN: 2373-7379	Kapitel 3
[86]	P. Fotouhi Tehrani, E. Osterweil, T. C. Schmidt und M. Wählisch, "A Security Model for Web-Based Communication," *Communication of the ACM*, 2024	Kapitel 2 Kapitel 6
[79]	P. Fotouhi Tehrani, R. Hiesgen, T. Lübeck, T. C. Schmidt und M. Wählisch, "Do CAA, CT, and DANE Interlink in Certificate Deployments? A Web PKI Measurement Study," in *Proc. of Network Traffic Measurement and Analysis Conference (TMA)*, Dresden, Germany: IEEE, Mai 2024, S. 1–11	Kapitel 5
[342]	P. Fotouhi Tehrani, E. Osterweil, T. C. Schmidt und M. Wählisch, *How to Measure TLS, X.509 Certificates, and Web PKI: A Tutorial and Brief Survey*, 2024	Kapitel 4
[341]	P. Fotouhi Tehrani, R. Hiesgen, T. C. Schmidt und M. Wählisch, *A Call to Reconsider Certification Authority Authorization (CAA)*, 2024	Kapitel 7

Teil I

Modeling Trust for Online Communication

Kapitel 1

Toward an Integrative model of Trust in Digital Emergency Communication

Abstract

Digital technologies have become an integral enabler of communication during various phases of emergency management (EM). A crucial prerequisite of effective communication between authorities and the public during EM is the establishment of adequate mutual trust. Trust, however, is an elusive concept which is not easily translatable into technical settings. In this paper we propose an integrative model of trust in digital communication and show how such model can be advantageous in assessing and improving trust relations in context of EM. Our interdisciplinary model, which is based on findings from psychology, sociology and computer sciences provides an abstraction which not only seizes both subjective and objective as well as personal and non-personal, *e.g.,* institutional or cultural, aspects of trust but at the same time is concrete enough to be applicable to real-life scenarios.

1.1 Introduction

The Munich rampage of 2016, leaving nine people dead and many injured, is a prominent example of how vulnerable our emergency communication infrastructure truly is: while the high amount of data traffic at that time rendered communication over the Internet practically impossible, online social media platforms, such as Twitter, were being flooded with false information regarding the nature of attacks and possible targets [289]. Even for those at site, who were able to communicate through overloaded channels, it remained a nontrivial task to decide whom or what to trust.

The penetration of digital communication in nearly every aspect of our private and public life raises the question of how to integrate or realize trust over technical infrastructure. At the same time, the multi-faceted nature of trust poses a real challenge to its translation in its broader sense into technical context. Whatever factors may be influential in building trust, mediated communication, *e.g.,* digital communication, almost always leads to loss or at least inhibition of cues or signals that are crucial to the construction of trust relations [442]. The interleaving of digital and real world aspects in emergence or suppression of trust in the context of digital communication highlights the necessity of an interdisciplinary study of the subject matter.

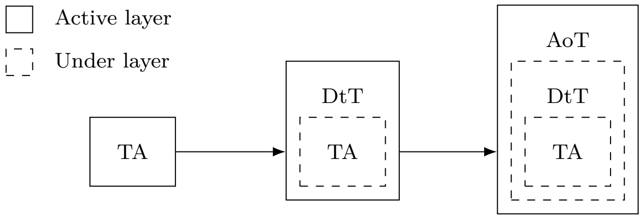


Abbildung 1.1: Trust stages and layers: Trust Attitude (TA), Decision to Trust (DtT), and Acting of Trusting (AoT)

By combining different understandings of trust from psychology, sociology, and computer sciences, we propose an integrative model which can lead to a better understanding of trust in digital communication, specifically in context of emergency communication. We understand trust as both cause and effect which emerges within a given *context* and is based on presence of some *evidence*. Our main goal is to provide a concrete basis upon which emergency management practitioners can better comprehend their relationship with their audience, and to equip them with a framework that, in contrast to existing approaches, proves to be practical beyond experimental or research settings.

The rest of this paper is structured as follows: in Abschnitt 1.2 an overview of trust among various disciplines are provided. Our integrative trust model is introduced in Abschnitt 1.3, and Abschnitt 1.4 discusses how this model can be utilized for emergency communication. Finally, in Abschnitt 1.5 we conclude our work and draw a brief outlook for future work.

1.2 Trust among Various Disciplines

Trust is an elusive concept which has been subject of study in various disciplines (see [41], [406]. In the following we give a brief overview of how trust is conceptualized in fields of psychology, sociology, and computer science.

1.2.1 Psychology

In psychology the most common and favored definition considers trust as "a psychological state of a trustor comprising the intention to accept vulnerability in a situation involving risk, based on positive expectations on the intentions or behavior of the trustee" [209]. The emphasis here lies on the assumption that trust is a psychological condition which causes a certain behavior or choice and presupposes a) expectation of beneficial outcome and b) existence of risks and uncertainty [216]. This definition can be formalized as "X trusts Y in context C for performing task τ (with the action α and the result P), where τ corresponds to goal g of X" denoted as g_X [309]:

$$\mathbf{Trust}(X, Y, C, \tau, g_X)$$

In this model the establishment of trust shows some parallels with cognitive skill acquisition processes, particularly by uniting dispositional and experiential determinants. The model postulates that the trust level of an individual in a certain situation emerges from situational context, experience of that individual, cognition state (attention, memory, decision-making processes, etc.), and social contact behavior, in which the context has the major influence ([237]. Castelfranchi und Falcone 2010 further elaborate this concept of trust using a three-layered model (Abbildung 1.1), which postulates that trust is a process of three steps:

- **Trust Attitude (TA)**: the mental attitude, *i.e.,* disposition, towards an entity.
- **Decision to Trust (DtT)**: the decision to rely on another entity, which makes the trustor 'vulnerable'.
- **Act of Trusting (AoT)**: the behavior, *i.e.,* the act of trusting.

Besides trust, *trustworthiness* is being defined is an important part of the process of trust establishment, but is not equal to trust itself. While trust is a psychological state, which involves two entities (the trustor and the trustee, which can be individuals, groups or institutions), trustworthiness is just a characteristic of the trustee which is signaled to and interpret by the trustor [216]. This signal has the meaning that the trustee is worthy of being trusted [459]. In interpersonal trust there are three characteristics, which influence the perceived trustworthiness of an individual: ability, benevolence and integrity [216]. In a digital context, such characteristics are rather generic terms to describe the antecedents of the perceived trustworthiness, for example, there is evidence that in context of an online environment the structural design (usability), the content design (information quality), the social-cue design (indicators of social presence), and the reputation of the system influence the perceived trustworthiness [343], [254], [259]. Usability plays an important role in this. So, participants are more dedicated to use applications, which are easy to use, well-suited to their purposes and well-designed. By aesthetics and a useful design, users can be convinced, that a lot of time and effort had been invested into producing the application. These investments are perceived as serious and trustworthy approach by the user [103].

1.2.2 Sociology

Sociological studies in the past decades have dealt with the concept of trust under a variety of terms such as *solidarity* [328], *social ties* [457] or *reciprocity* [95]. In contrast to psychology there is, however, no consensus to define and understand trust as an individual state of mind. In this context social trust often puts the emphasis on relational and situational aspects [246], [257].

Luhmann recognizes trust as a mechanism of reducing complexity which keeps us capable to (inter)act even in situations of uncertainty [395]. Only with uncertainty – as the end of rational prediction [346], [395] – a window of opportunity opens and one can act in faith [28], [144], [170], [457]. Going one step further, Luhmann distinguishes between confidence and trust concluding that a situation of trust is based on the awareness of possible disappointment and the decision to trust anyway [396]. Luhmann's definition of trust resonates some aspects widely agreed by others, namely the willingness to become vulnerable while having positive expectations [24], [209].

An important notion that sociologists add is that quantification as well as a finite list of factors of trust stand contradictory to the idea of trust itself. Trust is not rational [359], rather it goes beyond mere economic rationality [277], [70], [488], which is constrained to calculation and prediction of trust.

1.2.3 Computer Science

In computer sciences, trust and trustworthiness are often used interchangeably and generally formulated as a quantifiable value on a continuous or discrete scale based on some evidence either provided actively or collected passively. A piece of evidence asserts a claim either about an entity, *e.g.,* trustee, or an action or more specifically the correctness of an action. The most prominent examples are the public-key infrastructure (PKI) (see [473]), Web-of-trust (WOT) [492], blockchains, and reputation systems (see [29]).

In PKI and WOT the notion of trust is based on binding identities to cryptographic keys through *digital certificates*. A digital certificate is provided by an entity as a proof of identity (see [297]). Such certificate is comparable to an identity document in the real world, such as a passport. The idea is to prove that behind a digital representation lies a real-world entity; if a trustor is provided with adequate proof of identity, then it can decide either to trust or not to trust that entity based on its preexisting trust relation in the real world. In PKI, trust anchors, *i.e.,* entities which are already identified and marked as trusted by a trustor, vouch for correctness of such real/virtual binding by cryptographically signing the respective certificate just a national government would vouch for correctness of a passport. As a generalization of the PKI concept, WOT resembles a social network where every entity can vouch for the identity of another. To decide if an unknown entity can be trusted or not, one checks how many of its already trusted or semi-trusted peers can vouch for that unknown entity. In contrast to PKI, where trust is a binary attribute, WOT introduces *marginal trust*, in addition to *complete trust* and *untrusted*; an enhancement that does not necessarily simplify the assessment of trust relations.

Trust in the context of blockchain technologies is reduced to verifiability of implemented protocols (see *proof of work* in Bitcoin protocol in [415]. In this sense trust(worthiness) is not defined in relation to any specific entity, rather in regard to an action or process. Here, immutable transactions are bundled into blocks which are then chained together, while it is computationally possible to verify each and every transaction from the very first to the last one and ensure that none has been tampered with. Under the assumption that no involved entity in the distributed ecosystem of blockchain is inherently trustworthy, as long as all transactions and processes are verifiable, the overall system is considered to be trustworthy, *i.e.,* acting in an expected manner.

Finally, reputation systems bring temporal aspects of trust into account by collecting information about or observing the behavior of an entity in a given period: to estimate the trustworthiness of an unknown entity, one can either observe the behavior of that entity through time, or rely on statements

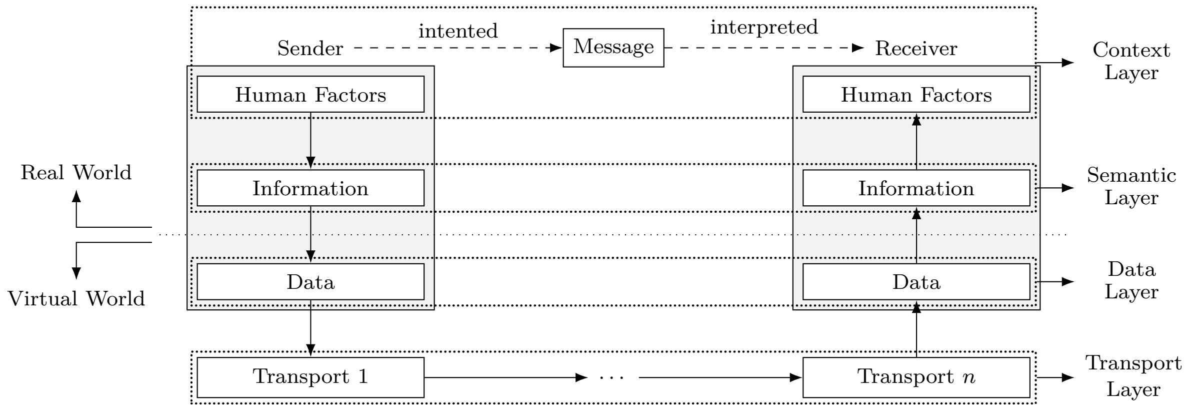


Abbildung 1.2: Digital Communication Model

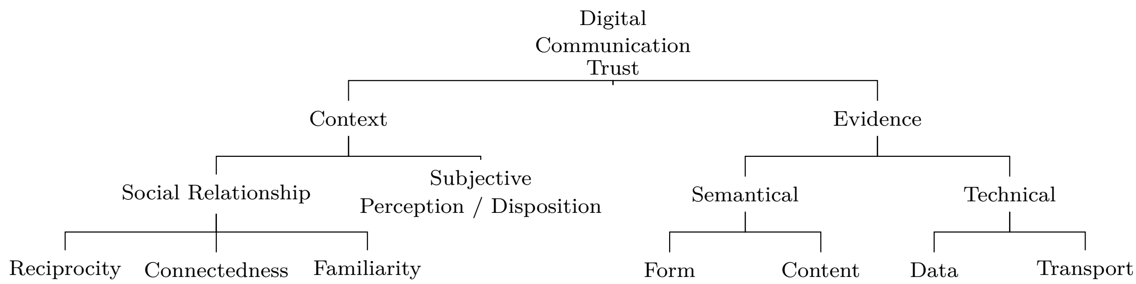


Abbildung 1.3: Taxonomy of Trust in Digital Communication

from other trusted entities. The observation can either be on a local scale, e.g., watching neighboring nodes in an ad hoc network, or on a global scale, e.g., ranking of sellers on an e-commerce website. Whereas in general only a single factor is observed, introducing multiple factors can improve the precision and reliability of calculated trustworthiness (see Trapp, Wählisch und Schiller [474]; *cf.,* Granovetter [96]).

1.3 An Integrative Trust Model

In contrast to vis-à-vis interaction, the biggest obstacle in establishing trust over digital communication is inhibition or at least distortion of signals necessary to evaluate a trust relation (*cf.,* [442]). This effect is depicted in Abbildung 1.2 (*cf.,* [216]): the sender formulates information which is then represented as data packets which subsequently are transmitted over a transport layer, reconstructed as data packets, interpreted as information and finally perceived by the receiver. The transition between each layer causes loss or transformation of signals that could be integral for evaluation of trust by the receiver, *i.e.,* the trustor. Additionally, there is a common context that influence the communication between the sender and the receiver. Adequately modeling trust, thus, entails addressing and understanding of processes occurring at each layer in both real and virtual spheres.

Based on the findings presented in the previous section, we consider trust to be a subjective experience which emerges in face of uncertainty under positive expectation regarding the qualities of an entity. In context of digital communication, trust relations are constrained and impacted by the digital medium, through which the interaction between trustor and trustee succeeds. In other words, the mediated communication between the entities is a decisive factor for establishment of trust[1]. Although there is no consensus as to which factors amount to experience of trust (see [29] for an overview), we assume that the following holds: i) trust is context dependent, and ii) trust is evaluated based on available signals or cues which we henceforth refer to as *evidence*. Based on existing research and related works, we have developed a taxonomy (depicted in Abbildung 1.3) which summarizes our abstraction of trust relevant for digital communication. In the following we provide clarification for each point and give an overview of related works.

1.3.1 Context

Context comprises both subjective and objective aspects of trust. Despite existing efforts on formalizing trust using mathematical tools and notation[2], we tend to believe that such formalisms, *e.g.,* trust as a quantity between -1 and 1, fail to grasp and reflect the concept of trust in its totality, thus, limiting our scope to qualitative factors of trust.

Disposition to Trust

Trust is fundamental to building social bonds, thus, risks and uncertainties in social interaction appear more acceptable through trust [331]. These uncertainties are rooted in the fact that in the fewest decisions, complete knowledge or complete information is available [41]. Trust can be viewed from two different perspectives of the disposition and experience [60], which are not clearly distinguishable from one another. The former sees trust as a deeply rooted disposition, which is conveyed from generation to generation in the context of early childhood socialization and which can be seen as a cultural feature of a society [60], [323], [231]. The latter, *i.e.,* the perspective of experience, sees trust as a result of formative experiences such as active participation in society by means of social interactions, upbringing and education, as well as an institutional context as a possible 'safety net' [323], [237]. These approaches are difficult to distinguish from one another, since early childhood socialization is also an experience. In fact, the relationship between disposition and experience is reciprocal, as formative childhood experiences and following education are of utmost importance for the disposition to trust.

Social Relationship

Multiple aspects and characteristics of social relationships are emphasized to influence trust: many studies distinguish between different levels of society (micro, meso, macro) and specialize on certain types with respect to this differentiation. For example, in organizational sociology much attention is

[1] See Riegelsberger [442] for an exhaustive investigation and Bos, Olson, Gergle u. a. [27] for concrete examples.
[2] See Marsh [400] as one of the very first pioneering works in formalizing trust.

given to trust between individuals, between individuals and organizations [248], and among organizations [62], [433]. Fewer but still a wealth of approaches are concerned with trust between state actors and organizations and/or individuals [344], [435]. Even though most studies focus on one layer and aspect, there is an agreement that these levels intervene with each other and can not be separated in practice [59] while certain variables influencing trust levels can be found on every layer. Due to the focus on digital communication we highlight three of these relational aspects for building trust, which are addressed as influential in psychological, sociological and computational research: reciprocity, familiarity and connectedness.

Reciprocity is understood as an aspect of the power structure between trustor and trustee that evolves around the issue of who is (able to) send and receive impulses: the trust received by an institution from individuals is not necessary reciprocated, for example, Tapia u. a. [239] studies how various NGOs initially leveraged Twitter to disseminate information but did not use it to collect back data as they did not perceive senders (presumably recipients of their messages) as trustworthy. Nevertheless, reciprocity is commonly seen as an important factor and basis of trust [316], [275] (as organic solidarity, based on 'exchanging' even found by [328]).

Familiarity is understood as the historical and temporal dimension of trust which denotes the commonly expressed expectation that trust evolves over time. Concepts like *trust dynamics* try to grasp this general understanding and to bring it in line with observations and practical experiences [72], [344], [209]. This, however, fails to be an eligible case of compelling causality: familiarity is rather a variable for the trustor to calculate the risk of disappointment due to previous experiences with the trustee [346], [163]. This understanding includes the difficulties to rebuild broken trust [225].

Connectedness captures the relationship of entities within their social networks. The premise of connectedness is the fact that just like any kind of other of information being exchanged within ones social network, information about trust relations, *i.e.,* trustworthiness, credibility or reputation (*cf.,* Mui u. a. [172]), are also passed on from one entity to another and can be integral to building own trust relations.

1.3.2 Evidence

Evidence is the signal or cue that amounts to establishment of trust relations. Here, we investigate two type of evidence that can be transmitted over digital infrastructure: semantical and technical.

Semantical

Content and form are qualitative types of evidence that can influence the trustworthiness perception of a message and its sender. Content, ranging from textual body to seals of approval and third-party certificates [343], [254], has a major influence on the perceived trustworthiness of a technical system, *i.e.,* its reliability and correctness of provided information [48], [143], [259]. Whereas precise and well-formulated content can enforce trustworthiness, other factors, such as advertisement on a website or impolite and non-constructive messages have a negative influence on the perceived trustworthiness [48], [103].

Graphical attributes also have influence on the perceived trustworthiness of a technical system. For example, Miran u. a. [164] studies the effect of color usage in combination with background maps "on users' perception, interpretation, and reaction to threat information" in context of probabilistic hazard information. Similarly, the effect of form on trustworthiness has also been studied on the world wide web: while the impact of the choice of colors to achieve a high trustworthiness (postulated by Wang u. a. [254]) is negligible [343], the general aesthetics of the website have a big influence on the perceived trustworthiness. Professionalism and usability of a website are associated with the perceived aesthetics [18], [78].

Technical

Under technical evidence, we subsume information security measures, such as confidentiality, integrity, authentication and alike (cf., CIA triangle [481]). Such measures can either be applied directly on data or on transport layer, for example, data can be encrypted and transmitted over insecure transport channels or the channel itself can be secured and plain data is transmitted securely. The decision whether to secure data, transport layer or both depends on the given situation. Protocols, such as SSL/TLS, which depend on an end-to-end connectivity between communication partners, are not applicable where networks are fragmented and data is transmitted using data mules. On the other hand, if the communication succeeds over multiple transport protocols, or over insecure channels, protocols such as OSCORE [450], can be leveraged to secure data directly regardless of the transport layer capabilities.

Technical evidence can either be self-sufficient or third-party-dependent. Measures to secure data integrity, such as checksums, for example, are self-sufficient as the trustor can verify that the data has not been tampered without reliance on any external party. Other measures, such as X.509 certificates [297] which bind names to cryptographic keys and are used for authentication, depend on trusted third parties (TTP) to vouch for the binding by signing a digital certificate. Integral aspect in verification of evidence of the latter type is a bootstrapping phase to mark which TTP is trustworthy.

1.4 Shaping Trust in Emergency Communication

Emergency management is an enterprise encompassing policies, activities, and measures with the aim of preparing for, mitigating, responding to and recovering from emergency situations (*cf.,* Blanchard [296]). In every phase of EM, communication (between and among) authorities and the public, plays an integral role [200], not just as a functional necessity, but as "the most important and least understood role for policy makers" amounting to reinforcement of social resilience [150].

Effective communication relies on presence of trust among communication partners, however, depending on involved actors and each phase of EM, factors introduced in the previous section (see Abbildung 1.3) may have a stronger or lighter impact on trust relations: for example, whereas trust in technology in virtual emergency response teams may have a higher priority (*cf.,* Büscher u. a. [32]), trusting crowdsourced data over social media may rely heavily on semantic evidence and connectedness of actors within the network (*cf.,* Mehta u. a. [162]). Specially during response and recovery phases, dynamics of trust relations are changed: whereas on the one hand the time pressure to act and react does not leave enough development room for temporal aspects of trust building, network infrastructure may also be fragmented, isolated or overloaded so that technical trust mechanisms are not applicable anymore. In short, during response and recovery "sharing and dissemination of information is both critical and problematic, beginning with whom to trust in unfamiliar settings." [160].

According to our proposed model, both context and evidence are integral in maintaining trust relations. There is, however, a reciprocal excretion of influence between these factors: one can imagine how context is decisive for choosing proper evidence and how evidence shapes the context in a persisting cycle of trust formation. For the sake of simplicity, we limit ourselves here to response and recovery phases where the context is given and cannot easily be changed but the evidence needs to selected accordingly. To show how our proposed model can be advantageous for strengthening trust in emergency communication, we appeal to example of *emergency alerting applications* such as FEMA's app[3] in the U.S. or KATWARN[4] [403] in Germany. The point of departure is the systematic analysis of the context between trustees, *i.e.,* alerting authorities, and possible trustors, *i.e.,* citizens receiving alerts: in this case the relationship is not reciprocal so that the communication succeeds from alerting authority to citizens; the recipients might be in a vulnerable emotional and psychological state, and may have not yet familiarized themselves with responsible authorities. Considering the communication context and the constraints of given situation, an alerting authority can then choose proper evidence modalities to enforce the establishment of a trust relation: the better the understanding of authorities regarding the context, the more appropriate evidence can be chosen. Proper language, form, and content needs to be chosen to invoke the maximum perception of trustworthiness in recipients. Finally, the decision on technical evidence is highly dependent on the state of communication infrastructure: authorities should be aware that in case of disaster or catastrophes communication networks may be fragmented, ad hoc networks emerge as alternatives, data is relayed from producers to consumers over (possibly malicious) third party nodes, *e.g.,* data mules, and that common security measures are not applicable anymore. Since no assumptions can be made about the available transport layer, technical evidence should be limited to the data itself: on the one hand the data must be *self-certifying*, *i.e.,* provide evidence that it hasn't been tampered with, and on the other hand it should be *self-authenticating*, *i.e.,* carry a digital signature to authenticate its origin. It should be noted that self-authentication presupposes a *trust bootstrapping phase* in which the certificate of origin is retrieved and marked as trusted by the trustee, *i.e.,* alert recipient.

[3] `https://www.fema.gov/mobile-app`
[4] `https://katwarn.de/`

1.5 Conclusion and Outlook

Trust remains an attractive subject of study among various disciplines. In this paper we balance abstract notions of trust and practicability by introducing an integrative model of trust, which comprises both quantitative and qualitative aspects of trust in digital communication. Our model serves organizations and individuals involved in emergency communication as a point of departure to assess their policies, procedures, and technical infrastructure with the aim of improving trust with their audience.

Although in our use case scenario, the focus lies on response and recovery phases and trust is understood as effect of available evidence, in future work we will investigate trust not just as cause but also as a cause which can be attended to during preparedness and mitigation phases. In other words, trust will be integrated into the perpetual cycle of emergency management; partly as a cause and partly as effect.

Kapitel 2

A Security Model for Web-Based Communication

Abstract

Web access involves various protocols to resolve domain names to IP addresses, establish data exchange channels with Web servers, and to authenticate communication partners. Each protocol has its own set of requirements and security measures. In addition to technical features, operating the Web also introduces organizational and political aspects which are important to consider when deploying a secure basis for Web-based communication.

In this paper, we propose an algorithmic security model based on the widely deployed technologies DNS(SEC) and Web PKI to cover the three dimensions *identification*, *resolution*, and *transaction*. Our model enables quantification and qualification of the security assurance provided by an online service provider.

2.1 Introduction

Online media have been proven to be rather effective in communicating with and providing services to the public. An ever-growing number of US Americans prefer to get their news online [348], in Germany, the Internet is cited as the most popular information medium [463]—just to mention two examples. Exceptional events such as the Ukraine war or COVID19 pandemic led to worldwide significantly increase in Web-based services [171] and made clear that the Web is used to communicate important information. The United Nations, for example, launched *Diplomatic Pulse* [371], a comprehensive interface to official statements of UN Member States, which is fed by scraping information from at least one official Web presence per state. Parallel to the valid use of Web communication to inform the public, the amount of fraudulent services has also been on a constant rise. During the COVID19 pandemic an overwhelming number of instances of an *infodemic* [482] were launched and continued to prevail [298] despite efforts from top tech companies [455].

The fundamental base for leveraging Web content is trustworthy communication with the content endpoint. Assessing the level of provided security is challenged, though, due to the complexity of the current Web, which involves multiple protocols and stakeholders when accessing a Web page.

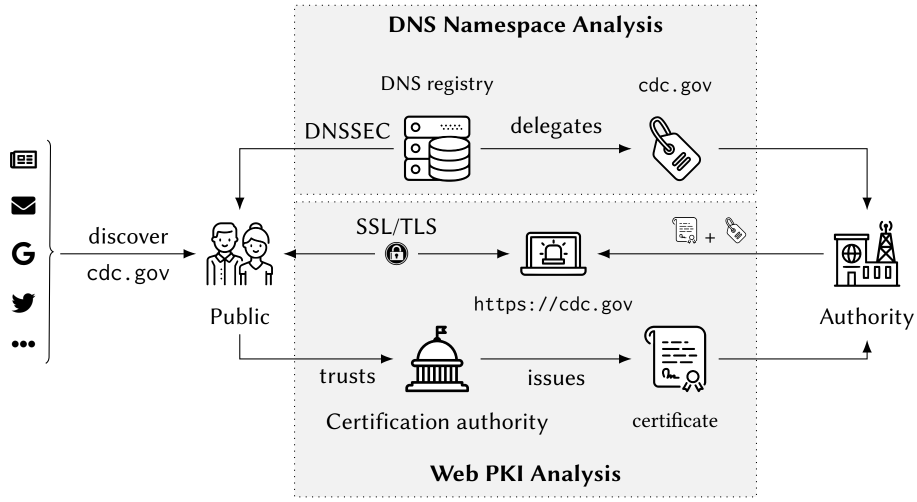


Abbildung 2.1: Accessing data via the Web

In this paper, we introduce a generic security model for Web services based on three dimensions of resolution, transaction, and identification. Using this model, we provide a framework to understand whether and how specific integration of a service provider in the domain namespace and its use of DNSSEC [444]–[446] and X.509 certificates impact secure communication (see Abbildung 2.1). We calculate an *Assurance Score* for a given service provider and qualify its provided level of security as having either inadequate, weak, or strong *Assurance Profile*. Through this lens, we systematically investigate the digital representations of organizations active in emergency management or diplomatic relations.

2.2 Background

While establishing a communication channel between two endpoints involves a number of different protocols, securing that channel as a whole requires not only securing each protocol but also their overlaps, *i.e.,* the intersection between the involved protocols. The two integral protocols of web communication are DNS, a distributed database mainly used to resolve domain names to IP addresses, and HTTP for actual data exchange. To initiate communication, the legacy insecure procedure is for a client to first have a recursive resolver (RR) resolve a domain name to an IP address, and then communicate with the web server behind that address over HTTP protocol. In this procedure the clients are left with no choice but (*i*) to assume that DNS records have not been spoofed midway, and (*ii*) to assume that data has not been tampered with and originate from the authentic server. To address these security weaknesses, the *DNS Security Extensions* (DNSSEC) [444]–[446] and transport layer security protocols (SSL/TLS) were introduced. DNSSEC provides otherwise insecure DNS resolution protocol with data origin authentication and data integrity [445], and the *Transport Layer Security* (TLS) protocol (successor of SSL) brings identification, and authenticated integrity and confidentiality (using X.509 certificates [297]) to the HTTP protocol in context of the Web PKI ecosystem. These new extensions, however, also introduce new attack surfaces. DNSSEC can be enabled for a domain name only if the chain of preceding zones are secure all the way up to a trust anchor so that if a preceding zone is compromised, the entirety of its namespace can face validation problems. Within the Web PKI ecosystem, certification authorities (CAs) are generally not constrained in certificate issuance and a single compromised CA can issue certificates for any arbitrary domain name; moreover, automated domain validation certificates can be misissued if DNSSEC is missing. Advantageously, each protocol provide measures that can counteract the shortcoming of the other: the *DNS-based Authentication of Named Entities* (DANE) allows domain name owners to describe what (type) of certificate is to be expected from the respective web server through `TLSA` records, thus, extending the DNSSEC trust chain [376] to the Web PKI while neutralizing certificate misissuance. X.509 certificates incorporate domain names either as common name (although deprecated [438]) or Subject Alternative Name (SAN) so even if resolution is compromised, a specific certificate cannot be used to authenticate any other arbitrary name. Beside technical procedures, organizational aspects of DNS and Web PKI ecosystems are also relevant for securing web communication. Specifically, restricted namespaces with eligibility requirements, *e.g.,* `.gov`, and organization or extended validation (OV/EV) [299] certificates, which are only awarded to legal or natural persons under more strict conditions, can help clients to better identify their communication partners and respectively assess their trust relations with them.

2.3 A Security Model for Web-Based Communication

We define a security model for web-based communication based on three dimensions: (*i*) securely authenticating the authoritative service ("identification" of the person, organization *etc.* behind the service name), (*ii*) securely verifying that users have not been misdirected and are transacting with the service name they have identified ("resolution" of name to network service), and (*iii*) ensuring that the content was not altered, leaks privacy, *etc.* during the session ("transaction" security). Although different methods can be utilized to realize such a secure workflow, here we focus on those technologies that are most accessible to (and deployable by) users and service operators in today's Internet, namely the DNS and the Web PKI ecosystems.

An illustrative example. To demonstrate the communication workflow and respective security pitfalls, we consider the simple case of inquiring information about COVID-19 guidance as a resident of Jackson County in Missouri. Through a search engine, an online ad, a recommendation from friends, *etc.* the URL is quickly discovered: `https://jacohd.org`. When visiting the website, the presence of a green padlock in the address bar indicates *only* the confidentiality and integrity of data exchange, but does not indicate whether the website belongs to the supposed service (*i.e.,* Health Department of Jackson County in Missouri instead of one of the other 22 Jackson Counties). The generic domain name could have been registered and operated by anyone. An attacker could have published a forged website implementing the look and feel of the real health department. At no stage is the user given the chance to authenticate the identity (*i.e.,* identification) of the service provider because the provided DV certificate does not include any identification information. In contrast, the health department of

Jackson County in Michigan, reachable under `www.co.jackson.mi.us`[1], is under a restricted TLD indicating that it belongs to Jackson County (`co.jackson`) in Michigan (`mi.us`), while accompanied by an EV certificate which serves as a definitive proof of identity.

2.3.1 Secure Resolution

Web-based communication starts with resolving a domain name. There are two methods to assert that a name has been resolved correctly: either through DNSSEC cryptographic assurances or through X.509 bindings in a certificate's common name (CN) or subject alternative names (SAN). Whereas in the former case a name is resolved to an IP address of a server (through `A`/`AAAA` records) *before* communicating with that server, in the latter case the name binding can be verified only *after* the server has delivered a certificate. As such, the latter approach provides only an a posteriori assurance, and only under the assumption that the certificate is not misissued. Furthermore, in contrast to DNSSEC, the assurance that a certificate provides in terms of name binding is not necessarily backed by any cryptographic measures: a certification authority might allow domain validation per E-Mail or telephone or cater for automatic validation through DNS or HTTP challenges (see RFC8555 [292]). This process is carried out once and considered valid throughout the lifetime of the certificate. Especially in case of DV certificates, if resolution is compromised, *e.g.,* through DNS poisoning, and as a result attackers were granted a DV certificate (see Brandt *et al.* [30]), there is no way to verify the integrity of the resolution process. Thus, the only effective solution in securing name resolution and deterring collateral damages such as DV certificate misissuances is deployment of DNSSEC.

The case of secure resolution becomes complicated if a name is an alias for another, *i.e.,* through a DNS `CNAME` record. In such cases, the name itself might be under a secure zone with the `CNMAE` record signed and secure, but its canonical name might be positioned under an insecure zone. For example, `www.whitehouse.gov`, which is an alias for `wildcard.whitehouse.gov.edgekey.net.` is properly secured, whereas its canonical name (under Akamai CDN namespace) is not. In such cases, not only resolving an IP address is insecure, but no DNSSEC-dependent service, such as DANE which is critical for secure transaction as discussed bellow, can be implemented.

2.3.2 Secure Transaction

After successful name resolution, a client initiates the actual data transaction. The most common practice in securing web transactions is through transport security protocols (TLS/SSL) which enable server (and less commonly client) authentication using X.509 certificates and cater for confidentiality and integrity of data exchange. It is worth noting that authentication during data transaction is a crucial aspect, because encryption and integrity checks alone can also be performed by a malicious actor using monkey-in-the-middle attacks. Authentication in this context is the process of verifying that communication is taking place with the server discovered through the resolution process. Just as a DNS binds a name to an IP address, a certificate binds a cryptographic key among others to at least

[1] The complete URL is `https://www.co.jackson.mi.us/276/Health-Department`.

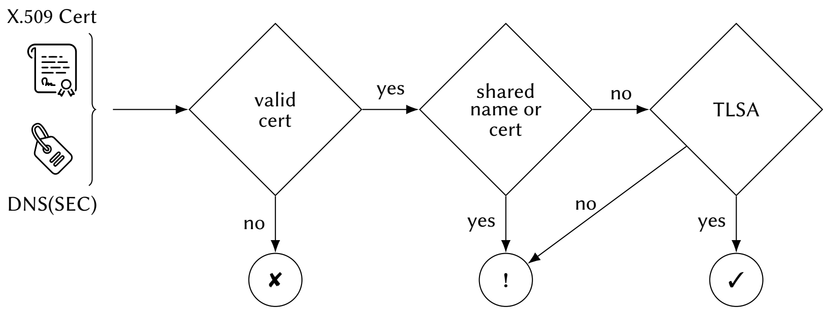


Abbildung 2.2: Classifying transactions using certificate, name, and handover characteristics (✘ Insecure ! Partial ✓ Secure)

one domain name. If a name is resolved to an address and a server behind that address provides a certificate bound to that name while cryptographically proving the ownership of that certificate, then the server is authenticated. This, however, only holds if the certificate itself is authentic, *e.g.,* not misissued. To counter this threat, the *DNS-based authentication of Named Entities* (DANE) [363] through TLSA records can be used to extend the DNSSEC trust to the Web PKI by signaling clients what (type of) certificate to expect from the server. This is an extra security measure to prevent adversaries from presenting bogus certificates to clients in case of server or CA compromises *even* if name resolution succeeds securely.

The transaction security depends on the certificate and name characteristics alongside the existence of a respective TLSA record. *Secure* transaction can be established if a dedicated certificate is provided and the expected certificate is described through DANE TLSA records. If a certificate is shared, *i.e.,* bound to multiple names (using SAN) or used by different entities under the same name, only *partial* security can be reached. Finally, If none or only an invalid certificate is used, we consider the transaction as *insecure.* It should be noted that since DANE relies on DNSSEC, secure transaction also implies secure resolution. Our classification workflow is depicted in Abbildung 2.2.

2.3.3 Secure Identification

By identification, we refer to the process of uniquely recognizing the communication partner; an entity in the real world, *e.g.,* a legal or natural person. In a secure setting, it would be possible to identify and authenticate the communication partner *before* initiating the transaction. Yet, the point of departure for Web communication are domain names, which cannot be used for secure identification, while Web PKI certificates (as proofs of identity) are provided only *after* resolution succeeds and transaction is initiated. This implies that targeting authentic names and subsequent secure resolution are necessary (yet insufficient) conditions of identification. Respectively, simple name spoofing, *e.g.,* through typosquatting [119], DNS cache poisoning [255], or other DNS hijacking attacks, which can mislead users to

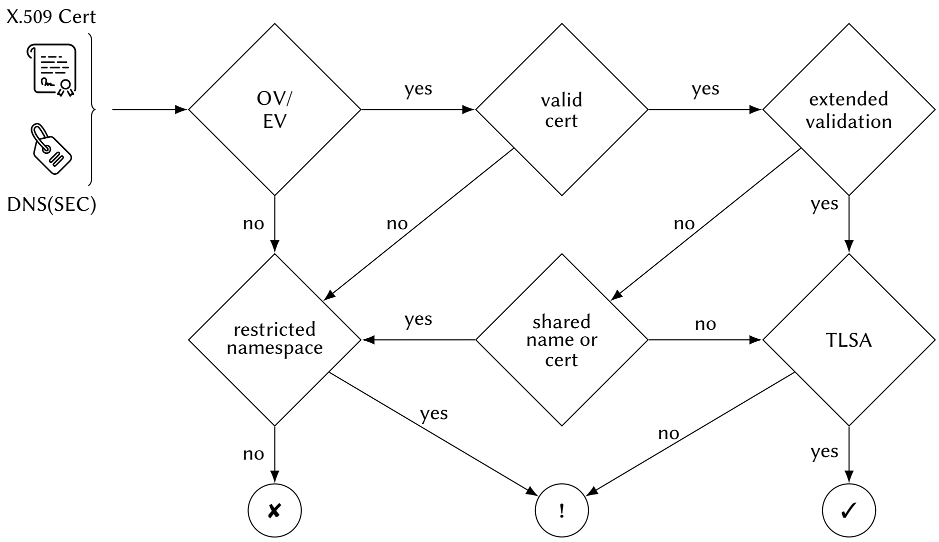


Abbildung 2.3: Classifying identification based on name, namespace, certificate, and trust chain characteristics (✘ Insecure ! Partial ✓ Secure)

malicious services, can act as a precursor for impersonation attacks, especially if subsequently only a DV certificate with no identity information is presented. A viable countermeasure is the use of restricted namespaces so at least the affiliation or identity of the service provider can be inferred directly from its respective name. Governmental organizations in the US, for example, educate visitors that domain names of federal government agencies most commonly end in `.gov` or `.mil`. Subsequently, an OV/EV certificate provides direct elements of proof of identity. When considering using approaches like this, exceptions may serve to help prove the rule: consider that the United States Post Service's (USPS's) official website is `usps.com`, *i.e.,* not under .gov. This, then, necessitates additional knowledge or verification before users of *that* government agency can be assured that they are transacting with the official authority online. Proper identification, thus, involves both selection of proper domain names, secure resolution, and identity information from OV/EV certificate.

We classify identification in three groups of insecure, partial, and strong: if an entity does not provide any identification information in terms of an OV or EV certificate and is not located under a restricted namespace, identification is impossible (*insecure*). If the available information does not suffice to uniquely identify the entity, only a *partial* identification can succeed: that is if the only piece of information at hand is a domain name under a restricted namespace, or if other information is available from an OV certificate but the certificate itself is shared or the respective TLSA records are missing (see above). Note that we explicitly consider EV certificates as dedicated by definition (see EV issuance and management guidelines [305]). Finally, we only consider the presence of a dedicated OV/EV certificate accompanied by a TLSA record as *strong* identification. The workflow for identification classification is

Tabelle 2.1: Assurance Scores ($resolution + transaction + 2 \times Identification$), and respective Assurance Profiles (● : strong, ◐ : weak, ○ : inadequate) based on the interplay of resolution, transaction, and identification dimension (✘/0 : insecure, !/1 : partial, ✓/2 : secure).

	Security Dimension			Assurance	
#	Resolution	Transaction	Identification	Score	Profile
1	✘	✘	✘	0	○
2	✘	✘	!	2	○
3	✘	!	✘	1	○
4	✘	!	!	3	○
5	✓	✘	✘	2	○
6	✓	✘	!	4	○
7	✓	!	✘	3	○
8	✓	✓	✘	4	○
9	✓	!	!	5	◐
10	✓	✓	!	6	◐
11	✓	✓	✓	8	●

depicted in Abbildung 2.3. Note that strong identification implies secure transaction and in turn secure resolution.

2.3.4 Assurance Profiles

To provide a comprehensive view on all three security dimensions, we introduce *Assurance Scores* and *Assurance Profiles*. An Assurance Score defines a measure that combines the amount each security dimension contributes to web-based communication. Each Score is then mapped to a Security Profile, which groups multiple deployment options into strong, weak, or inadequate.

In detail, we first codify each possible label of insecure, partial, and secure as 0, 1, and 2, respectively. Since resolution can either be secure or insecure it is codified with 0 or 2. Then, we calculate the *Assurance Score* as $resolution + transaction + 2 \times Identification$. We add extra weight to identification because this dimension is integral in secure communication—even if resolution and transaction are both secure, there is still a need to at least partially identify the communication partner.

Tabelle 2.1 provides the various possible combinations of all three dimensions alongside the calculated Assurance Score and Profile. To achieve *strong assurance*, a service provider must provide secure resolution, transaction, and identification (row 11). It is worth noting that secure identification implies secure transaction and, in turn, secure resolution. *Weak assurance* can be reached if resolution is secure, identification is partial, and transaction is at least partial (rows 9 and 10). Any other combination leads to an insecure Assurance Profile (rows 1–8).

To demonstrate how Assurance Scores and Profiles can be useful for service providers, we consider the example of `nrw-corona-soforthilfe.de`, a phishing campaign that successfully defrauded about 4000 applicants of the corona relief program in one of Germany's federal states. Before being confiscated, the domain name under the unrestricted `.de` namespace resolved securely to a web server outside Germany, providing a DV certificate issued by Let's Encrypt[2]. The calculated Score of this website would have been 3, corresponding to row #7 in Tabelle 2.1. The authentic counterpart, `soforthilfe-corona.nrw.de`, was located under the restricted namespace (`.nrw.de`) of the state of North Rhine-Westphalia, providing a shared DV certificate, without supporting DNSSEC, leading to score 3 (row #4). The score of the authentic website could have been upgraded to a weak assurance profile just by enabling DNSSEC (row #9).

2.4 Related Work

To the best of our knowledge, this is the first study proposing a security model for web services investigating how widely deployed technologies of DNS(SEC) and Web PKI can be leveraged to cater for trustworthy communication. Since the combined aspect of these technologies has not been studied in previous work, we limit ourselves to an overview of related work which studies DNS(SEC) and Web PKI on their own.

Domain Namespace and DNSSEC. The influence of a domain name on authenticating or at least recognizing the real-world entity behind that name has been investigated in terms of general trustworthiness associated with TLDs and impersonation of trusted entities through domain name masquerading. Walther, Wang, and Loh [251] examine how choice of TLD can positively impact the credibility of health websites. Seckler *et al.* [218] investigate how a relevant domain name, *e.g.,* a known TLD, can positively enforce familiarity and in turn increase trust. Similarly, a yearly report [426] commissioned by the Public Internet Registry examines the trustworthiness of select TLDs among NGO donors.

A closely related topic is how the domain namespace of malicious websites is structured and operated. Korczynski *et al.* [126] show how low pricing and registration barriers alongside the possibility of bulk registration is an enabler for malicious actors to migrate to new gTLDs. In a longitudinal study of typosquatting, Agten *et al.* [6] reveals how registration fees and registry policies can attract or deter malicious actors; practically determining the credibility of such TLDs (the top three most abused TLDs in the world are new gTLDs [461]). And Antonakakis *et al.* [12] introduce a reputation system for DNS to detect malicious domain names. Different studies show how scammers try to impersonate other entities by partly or fully integrating legitimate domain names in their own domain names [6], [124], [205], [241] or even by using homonymous names using internationalized domain names [232].

With regard to namespace security, studies in the past pinpoint a relatively low DNSSEC penetration due to various factors ranging from lack of support by local resolvers to server misconfigurations [45], [106], [146], [188] despite more than 90% of all TLDs being signed and supporting DNSSEC [369]. The

[2]Original certificates can be observed under `https://crt.sh/?q=nrw-corona-soforthilfe.de`

prevalence of DNSSEC among various types of organizations, such as educational, military, commercial, *etc.* has not been subject of study to determine if there is a correlation between field of operation and sensibility for DNS security measures. The only exception is the fine-grained, *i.e.,* including second level domains, regular analysis of DNSSEC deployment among select governmental agencies within the `.gov` namespace, educational institutions, and industry in the US [425], [208]. The prevalence of DANE [291], or more specifically the deployment of `TLSA` records [363], has been studied by Zhu *et al.* [491]. A more recent study takes a closer look at usage of DANE among E-Mail servers [139]. Nonetheless, if and how DANE is used to extend the DNSSEC trust chain to the Web PKI has not been extensively studied.

Web PKI. Throughout the years, various measurements have characterized X.509 certificates in use over the Internet in terms of validity, issuing CAs, key strength, *etc.* [42], [67], [111], [166]. Among these, Mishari *et al.* [166] investigate the difference between certificates of legitimate and fraudulent websites. The study by Holz *et al.* [111] has the advantage of being performed from different vantage points spread over the world. The measurements by Durumeric *et al.* [67] is noteworthy as it goes beyond mere X.509 certificate analysis and investigates the dependencies among root and intermediate CAs, their market share, and the characteristics of respective certificates. And finally, the measurements performed by Chung *et al.* [42] aim to understand why a majority of certificates advertised over IPv4 are invalid. It should be noted that except the last study, the others have been carried out before major changes in the Web PKI occurred, such as various mergers and the public launch of Let's Encrypt [1]. Furthermore, the findings from these studies exhibit different characteristics of sample sets, which are either too limited (*e.g.,* Alexa Top 1M) or too broad (*e.g.,* IPv4 space). Those differences do not allow for statistical inference and comparison with our observations.

The practice of sharing private keys of an X.509 certificate – defined as a third party, *e.g.,* a CDN provider, having access to private keys of a service provider– is investigated by Liang *et al.* [147] and extended by Cangialosi *et al.* [33]. Such definition relates to our concept of shared certificates in context of secure transaction, however, mere delegation of services to third parties is not considered as disadvantageous or insecure in our model.

Specifically related to the topic of our work are studies which investigate the trustworthiness of CAs in general and their policies specially in enabling fraud and impersonation. Delignat-Lavaud *et al.* [53], for example, investigate the conformance of CAs to the CA/Browser Forum guidelines, which in turn can influence trustworthiness of a CA. Others have defined various metrics to qualify [37], [71] or quantify trustworthiness of CAs [108] beyond technical measures. In a recent study Schwittmann, Wander and Weis [217], similar to Brandt *et al.* [30], exhibit how various CAs are susceptible to attacks on DV certification processes that can practically lead to domain impersonation. Roberts *et al.* [205] study which CAs are responsible for issuing DV certificates to malicious target-embedded domains; and similarly Kim *et al.* [123] investigates the practices of CAs in issuing and handling certificates for phishing websites. Stojmenović and Biddle appeal to using OV/EV certificates for identification and to rely on the Web PKI ecosystem to handle identification of online service providers [230].

2.5 Conclusion and Outlook

In this article, we conceptualized a security model to algorithmically quantify and qualify trustworthiness in Web communication. This security model covers three security dimensions (resolution, transaction, identification), which are combined into Assurance Scores grouped into Assurance Profiles. Assurance Profiles provide a lean, yet powerful view on the security support of a specific Web service. They may serve as an advancement in future Web browsers to hide the complexities in secure Web communication by comprehensively notifying end users whether the level of security is strong, weak, or inadequate.

In the future, when orthogonal security paradigms arise, such as content object security, or when emerging but fragmented edge networks, such as the Internet of Things, gain importance, the security model proposed in this article shall be extended, adapted, and applied. We urgently remind our community that complexity will always arise and demands for an automated and comprehensive security assessment integrated in end user software.

Teil II

Assessing Trust Measures on The Web

Kapitel 3

From the Beginning: Key Transitions in the First 15 Years of DNSSEC

Abstract

When the global rollout of the DNS Security Extensions (DNSSEC) began in 2005, a first-of-its-kind trial started: The complexity of a core Internet protocol was magnified in favor of better security for the overall Internet. Thereby, the scale of the loosely-federated delegation in DNS became an unprecedented cryptographic key management challenge. Though fundamental for current and future operational success, our community lacks a clear notion of how to empirically evaluate the process of securely transitioning keys.

In this paper, we propose two building blocks to formally characterize and assess key transitions. First, the *anatomy of key transitions*, *i.e.,* measurable and well-defined properties of key changes; and second, a novel *classification model* based on this anatomy for describing key transition practices in abstract terms. This abstraction allows for classifying operational behavior. We apply our proposed transition anatomy and transition classes to describe the global DNSSEC deployment. Specifically, we use measurements from the first 15 years of the DNSSEC rollout to detect and understand which key transitions have been used to what degree and which rates of errors and warnings occurred. In contrast to prior work, we consider all possible transitions and not only 1:1 key rollovers. Our results show measurable gaps between prescribed key management processes and key transitions in the wild. We also find evidence that such noncompliant transitions are needed in operations.

3.1 Introduction

Key transitions are critical for cryptographically enhanced infrastructures at the Internet-scale. The Internet is composed of loosely-federated administrative domains, and managing cryptographic keys under those conditions raises operational challenges in particular whenever domains depend on one another. Mismanagement of cryptography can lead to security shortfalls in those systems and infrastructures that depend on them (*e.g.,* [380], [387], [434], [475]). Regarding DNS, Paul Mockapetris and Kevin J. Dunlap (1988) wrote, "Distributing authority for a database does not distribute a corresponding amount of expertise" [167]. DNSSEC, which began its global deployment in 2005, implicitly raised the

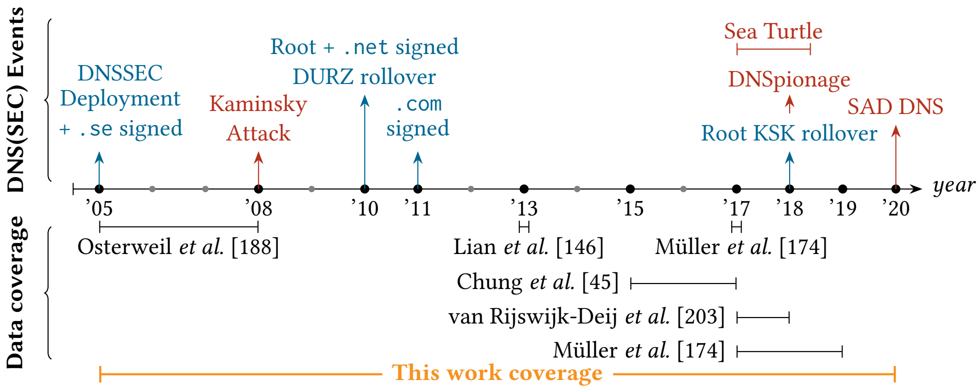


Abbildung 3.1: Notable DNS(SEC) deployment events (blue) and security incidents (red) during the measurement periods of related work (black) and this work (orange).

question: *Can* distributing responsibility to manage cryptographic material *teach us* the corresponding amount of expertise?

In DNSSEC it is common to periodically change the cryptographic keys in use. In some cases this is done as a hygienic prescription, in others it can be an emergency response to a security event (such as a key compromise or a cryptosystem weakness). The changing of keys is a process called key transitioning (or key rollover). In this process, a system gracefully transitions from using keys that are *departing*, *i.e.,* being removed, to using keys that are *remaining*, *i.e.,* kept unchanged or being newly added, while ensuring continuity of protections during the change. Following structured and validated process models for key transitions is critical for maintaining the security assurances of the overall system. Different infrastructures dictate different processes and prescribe them in different ways [20], [391], [413], [464]. While guidance for DNSSEC key life cycle management and timing exists in RFCs, a more foundational evaluation framework is missing so that the community can quantify and evaluate *all* operational aspects. Such a framework would not only allow us to compare real deployments to prescribed guidance but to each others as well.

In this work, we propose a novel method to precisely model DNSSEC key transitions and apply this to analyze and classify the data from the first 15 years of DNSSEC deployment (2005–2020). We define the composing elements required by our model as the *anatomy of DNSSEC key transitions.* Furthermore, we propose a measurement methodology to quantify key transitions observed in the wild. By using our anatomy and transition model, we are able to model key transition behaviors in the wild from both RFCs [413], [464] and from related work in the literature [256]. Our measurements cover $\approx$ 19 million key transitions. They reveal a significant amount of operational heterogeneity, many of which deviate significantly from standards without necessarily degrading security. Our contributions in this work are threefold:

1. **Anatomy:** We examine the timing features of keys while transitioning and propose an *anatomy of DNSSEC key transitions*, which defines a candidate set of measures that are necessary to measurably characterize key transitions.
2. **Transition classification:** Based on our proposed anatomy, we present a novel methodology, which we use to concisely quantify and analyze key transitions.
3. **Longitudinal study:** To illustrate the generality of this work, we present measurements of operational key transitions that span 15 years of DNSSEC deployment, covering a number of notable events that have not been fully analyzed by related works (see Abbildung 3.1).

The remainder of this paper is organized as follows. Abschnitt 3.2 summarizes background about DNSSEC. In Abschnitt 3.3, we propose our key transitions anatomy and explain how we construct our model of transition. We follow this by proposing a security analysis of DNSSEC key life cycle and key transitions in Abschnitt 3.4. We present the measurement corpus used in this work in Abschnitt 3.5. Abschnitt 3.6 details our methodology to derive a continuous model of the DNSSEC key lifetimes from discrete observations. Abschnitt 3.7 introduces our approach to classify life cycle management and transitions of keys. Based on that, we proceed to our extensive use case study in Abschnitt 3.8. We cover related work in Abschnitt 3.9 and discuss our results in Abschnitt 3.10. Finally, we conclude in Abschnitt 3.11.

3.2 Background

The DNS is a hierarchically administered global database of Resource Records (RRs), which are inserted and removed whenever operators choose. The design of the DNS allows the administrator of any sub-tree (called a DNS *zone*) to delegate the management authority of any branch under their zone to another authority. Delegations are implemented when a zone parent adds Name Server entries, *i.e.,* `NS` RRs, that point to the DNS name servers of that sub-zone. This hierarchical delegation allows administrators to operate their zones independently, and requires only a one-time coordination as long as the name servers remain the same.

To compensate for a number of security threats (see [288], [22]), DNS evolved to have the *DNS Security Extensions* (DNSSEC), whose specifications underwent their final round of standardization [444]–[446] in 2005. DNSSEC overloads the hierarchical namespace of DNS with cryptographic key learning and verification. By design, DNSSEC-enabled zones generate and manage their own cryptographic key pairs using any set of DNSSEC standardized cryptosystems. Operators then encode the public portion of their key pair in a new RR type, `DNSKEY`. A DNSSEC signature, an `RRSIG`, is generated by the respective private portion of a `DNSKEY` over each set of same-type RRs, called an *RRSet*, and is always returned with each DNSSEC response. An `RRSIG` specifies its *inception* and *expiration* times to limit its period of validity and to resist replay attacks. As these dates are defined in absolute values, DNSSEC implicitly requires "loose time synchronization" between authoritative name servers and validating resolvers [288].

DNSSEC specifies that zones should manage two separate classes of `DNSKEY`s: Zone Signing Keys (ZSKs) and Key Signing Keys (KSKs). While the cryptographic material used for these keys is fundamentally identical, their key life cycle management and roles are distinct: whereas ZSKs are used to sign all of a zone's contents (*e.g.,* `A` and `NS` records), KSKs are only used to sign `DNSKEY` RRSets. The root zone uses a well-known, self-signed KSK. All other zones need to have their KSKs authorized by a Delegation Signer (`DS`) record in their parent zone so that keys at each zone can be globally verified by Relying Party (RP) software (also called validating recursive resolvers, or just validators). In this way, validators use the KSK of the DNS Root zone as a Trust Anchor (TA) and point of departure to construct unambiguous verifiable paths to any DNSSEC zone in the hierarchy through recursive tracing of secure delegations. This follows the same way DNS zones are normally resolved recursively via `NS` records. The entire secure delegation chain from the DNS root to delegated zones, the *chain of trust*, is the element that creates operational dependence between the cryptographic management of zones and their parents.

To validate a single RRSet, an RP relies on various pieces of information (*i.e.,* `DNSKEY`s, `RRSIG`s, `DS` records) from possibly different zones and sources (*i.e.,* authoritative servers or caches). Validation of an RRSet in a given zone succeeds if (*i*) a verifiable path to that zone exists and (*ii*) the signature over that RRSet is valid. Breaking the first requirement causes an RP to determine the state of data as insecure, while neglecting the second ends with a bogus state [444]. These two requirements also constrain when and how operators can insert and remove RRs without defeating the protection provided by DNSSEC. Note that a valid signature presupposes that formal cryptographic requirements are met (*e.g.,* digests are correctly calculated), the key generating the signature is included in the `DNSKEY` RRSet, and the set is available and valid for at least the total validity period of the signature. We provide a thorough discussion on temporal constraints with a focus on key transitions in Abschnitt 3.4.

3.3 Modeling Key Transitions

Key transition refers to the procedure of modifying the set of valid DNS keys over time. The growing literature discusses DNSSEC key transitions in terms of key "rollovers" [173], [174], *i.e.,* a single new key replacing the only existing key. Based on a consistent monitoring of the first 15 DNSSEC years, we find evidence that the global DNSSEC deployment follows a different reality as is depicted in Abbildung 3.2. Besides the expected one-to-one transitions, a notable portion (13%) of transitions involve more than two keys and require a more expressive model. To reflect this reality, we introduce a generic model of key transitions.

In our generic model, a transition is characterized by an effective change of the DNSSEC key set, *i.e.,* a transition succeeds only if the removal of one or more *departing keys* results in an altered set of *remaining keys*. It should be noted that the set of remaining keys includes both newly added keys (if any) and keys which existed throughout the transition. Our model allows us to evaluate both simple key rollovers involving a single departing and a single new key as well as more complex key transitions involving multiple keys, also seen in the wild.

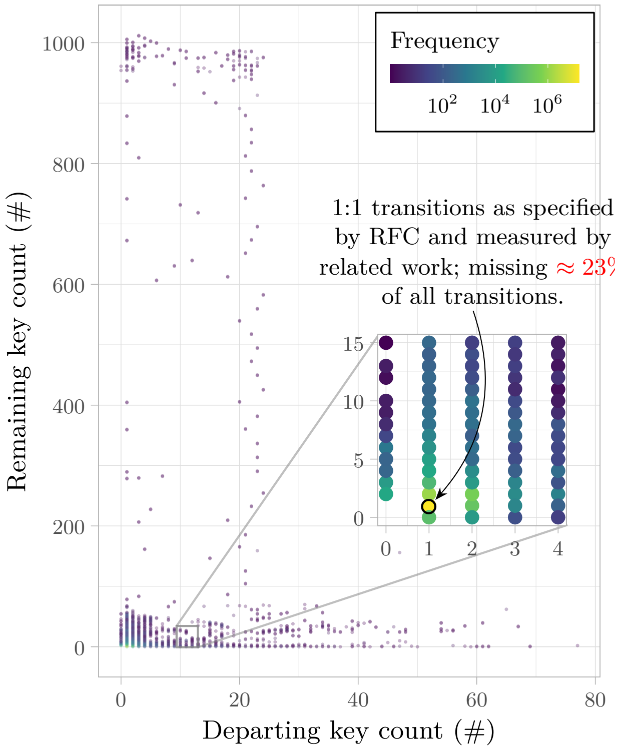


Abbildung 3.2: Frequencies of key transitions with different cardinalities between 2005–2020

In the following, we introduce a temporal model of DNS keys, which we use to define a *transition anatomy* and provide a method to capture the semantics of key transitions in terms of *transition classes*.

3.3.1 Anatomy of a Key Transition

Key transitions are measurable through changes in `DNSKEY` resource records, and their respective `RRSIG` records as published by the authoritative servers of the records. In the case of KSKs, changes in `DS` records require monitoring in the parent zone. While such changes are present, we consider a transition as ongoing.

Before we describe the temporal aspects of `DNSKEY` transitions, we first need to define a life cycle model for `DNSKEY`s, which adequately describes a key existence from its inception throughout its usage, run-out, and its expiration. These four phases are depicted in Abbildung 3.3.

The only temporal information that is explicitly expressed in DNSSEC about resource records is encoded in (*i*) `TTL` values and (*ii*) the validity period as defined by `RRSIG`s. `TTL` values are indicators used by caching resolvers to *locally* determine a time window after which a record should be considered stale and flushed from the cache. The main purpose of TTLs is to establish and maintain eventual consistency in caches [444, §8.1]. In contrast, the validity of `DNSKEY`s as denoted through `RRSIG`s provides information that can be used to reconstruct the life cycle of any key. Accordingly, we create a life cycle model of keys using (*i*) signatures *over* and (*ii*) signatures *generated by* those keys. Using

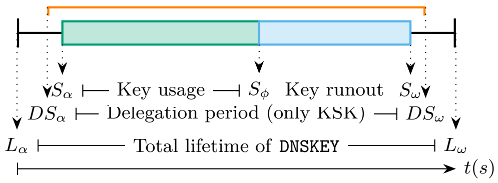


Abbildung 3.3: A temporal model spanning the *total lifetime* of a DNSKEY. RRSIGs generated by the key define active *key usage*, during *key runout* no new signatures are created while existing ones remain valid, and RRSIG(s) covering respective DS records define the key *delegation period*.

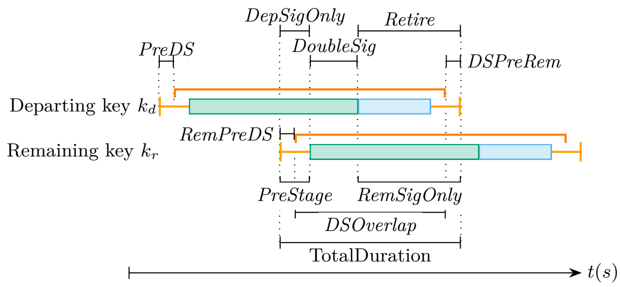


Abbildung 3.4: Anatomy of a 1 : 1 key transition.

RRSIGs *over* a key (recall that a single DNSKEY can be signed multiple times throughout its lifetime) the total lifetime of that key can be measured. We denote the earliest and latest point in time when the key was singed as L_α and L_ω, respectively. The RRSIGs *generated* by a DNSKEY, in turn, can be used to determine when the key was active and in use. Formally we denote the interval from earliest and latest times that data was verifiable by this key $[S_\alpha, S_\omega]$, the time when the key stopped generating new signatures S_ϕ, and subsequently the duration in which the key was only used to verify existing signatures $[S_\phi, S_\omega]$. Additionally, for KSKs only, the signatures over DS records of the parent zone are used to infer the period in which the parent zone was securely delegating to this key ($[DS_\alpha, DS_\omega]$).

We now use this temporal model to characterize a simple 1:1 key rollover. To this end we define a *transition anatomy* based on the following ten features, which include measures regarding a remaining key k_r and a departing key k_d, as visualized in Abbildung 3.4:

1. *PreDS*: When a corresponding DS record was introduced for k_d in the parent zone.

2. *DoubleSig*: The period during the transition when both departing and remaining keys were *actively* signing.

3. *PreStage*: The time during which the remaining key was valid, but *before* being used to verify zone data.

4. *DepSigOnly*: The duration during the key transition when *only* the departing key was in use.

Tabelle 3.1: Temporal features of Transition Anatomy and their respective intervals.

Feature	**Interval**
PreDS	$[L_\alpha(k_d), DS_\alpha(k_d)]$
DoubleSig	$[S_\alpha(k_d), S_\phi(k_d)] \cap [S_\alpha(k_r), S_\phi(k_r)]$
PreStage	$[L_\alpha(k_r), S_\alpha(k_r)]$
DepSigOnly	$[S_\alpha(k_d), S_\phi(k_d)] \cap [T_\alpha, T_\omega] - \mathit{DoubleSig}$
Retire	$[S_\phi(k_d), L_\omega(k_d)] \cap [T_\alpha, T_\omega]$
DSOverlap	$[DS_\alpha(k_d), DS_\omega(k_d)] \cap [DS_\alpha(k_r), DS_\omega(k_r)]$
RemSigOnly	$[S_\alpha(k_r), S_\phi(k_r)] \cap [T_\alpha, T_\omega] - \mathit{DoubleSig}$
DsPreRem	$[DS_\omega(k_d), L_\omega(k_d)]$
RemPreDS	$[L_\alpha(k_r), DS_\alpha(k_r)]$
TotalDuration	$[T_\alpha, T_\omega]$
∟ for ZSK	$[L_\alpha(k_r), L_\omega(k_d)]$
∟ for KSK	$[\mathit{min}(L_\alpha(k_r), DS_\alpha(k_r)), \mathit{max}(L_\omega(k_d), DS_\omega(k_d))]$

5. *Retire*: The duration during the key transition when signatures generated by departing key run out (run out described in Figure 3.3).

6. *DSOverlap*: The duration (if at all) that `DS`(es) for the departing and remaining keys overlapped.

7. *RemSigOnly*: The duration during the key transition when *only* the remaining key was in use.

8. *DsPreRem*: If the departing key was covered by a `DS`, the amount of time that the key was valid *after* `DS`(es) no longer delegated to it.

9. *RemPreDS*: When a corresponding `DS` record was introduced for the remaining key k_r in the parent zone.

10. *TotalDuration*: Duration of the entire transition.

Tabelle 3.1 describes these features, using our life cycle notation of involved keys presented in Abbildung 3.3.

Based on our anatomy, we are able to describe arbitrary transitions of n departing keys to m remaining keys, using $\binom{n+m}{2}$ pairwise $1:1$ transitions. Our measurements of operational zones over 15 years indicate that this distinction is important. For example, consider a zone of n keys (with possibly different initial inception times) and 1 to n keys in use to sign data. If that zone transitions to m keys (where 1 to m are used to sign data), a number of unknowns arise: Which key(s) rolled over to which other keys? Did all the departing keys roll over to all of the remaining keys? If some keys persisted, are they (partially) replaced, as well?

Abbildung 3.2 clearly shows that while the majority of observed key transitions change by one key (gaining or losing), many transitions cause alterations of $|m - n| > 1$. These fine granular observations of real deployments illustrate why additional expressiveness is needed, and why many of the previous discussions and characterizations of "rollovers" in the literature apply only to cases in which a single *active* key is rolling over to another single active key.

We extend these discussions by observing that if more than one key is added or departed at the same time, these are multiple concurrent transitions at the same timestamp. The intuition here is that no single departing key is measurably more pivotal than another. Thus, we define each departing key as transitioning to any of the keys that remain. This definition of key transitions allows us to measure operational behavior and answer questions such as: How many active keys are in use? When are transitions aborted or rolled back? When are secure delegations (from `DS` records) correct? Or, does a zone remain secure (see Abschnitt 3.4) as transition is ongoing. In addition, our transition model measures the relative ages of (the remaining to departing) keys: *newer*, *older*, or the *same* age. These semantics, although unmentioned in the RFCs, are useful in some process models (*e.g.,* [256] discussed below).

Our proposed anatomy is a fine-grained description of the atomic timing components that are necessary and sufficient to fully characterize key transitions. Whether key transitions are being performed manually, part of a process, fully automatized, or result from unsupervised ad hoc changes in a DNSSEC zone, the diversity of their activities is concisely represented by this anatomy.

3.3.2 Transition Classes

Our transition anatomy allows for the precise reconstruction and description of any pairwise key transition. Special care needs to be taken when *characterizing* transitions abstractly, as RFC 5011 [464] or RFC 7583 [413] do for example. We discretize the value of each transition feature (see Tabelle 3.1) instead of using their absolute values from empirical measurements. Different combinations of the resulting discretized feature set then represent transition classes. Transition classes allow us to compare and to assess whether prescribed security guarantees are preserved while keys are changing (*e.g.,* by adhering to specifications such as RFCs).

To classify a given transition, we first calculate the interval for each of its features using empirical measurements, *e.g.,* $PreDS = [1618016262, 1618048662]$. We then discretize the features (1) through (9) based on whether their interval widths (see Tabelle 3.1) are < 0, $= 0$, > 0, *e.g.,* $PreDS > 0$. For intervals defined as intersections of other intervals, *e.g.,* *DoubleSig*, the respective interval widths are always non-negative, whereas other measures can assume negative values, *e.g.,* the width of *PreDS* interval can be negative when a KSK is securely delegated *before* being signed and included in the `DNSKEY` RRset ($L_\alpha(k_d) \geq DS_\alpha(k_d)$). For cases in which an interval is undefined, *e.g.,* measurements of `DS` records for ZSKs that have no `DS` record to measure, we use the *N/A* placeholder. In Abschnitt 3.7, we will see that N/A cases do not have an impact on transition classes.

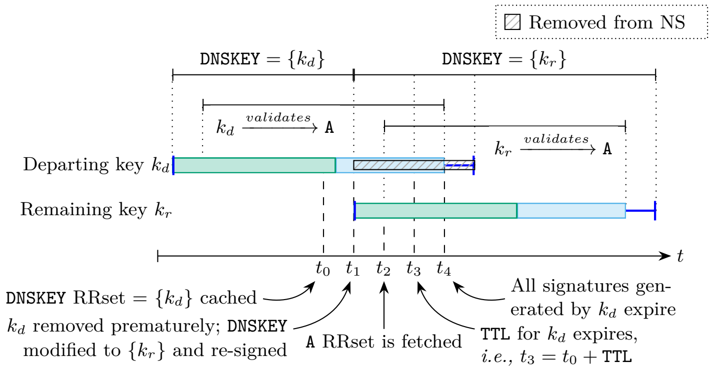


Abbildung 3.5: An example of how modifying and re-signing of an RRset (DNSKEY here) can cause failed validation in caching resolvers, while a valid signature over the RRset still exists.

Discretization facilitates an empirically simple comparison of completely independent key transitions. For example, if two keys in a transition are both observed to be signing data at the same time, their observed *DoubleSig* interval width would be a finite value. This would then be discretized as > 0. Two other keys, in another transition (possibly in another zone) would likely have different interval widths, but would be assigned to the same discrete classification value, and would thus enable comparisons between these transitions. As a result, every pairwise key transition can be represented as an ordered set of discretized features.

3.4 Security Analysis of Key Transitions

In this section, we discuss the temporal constraints that must be satisfied during key transitions to avoid zone data to be invalidated as insecure or bogus (see Abschnitt 3.2). Recall that a single record is considered valid if all RRsets that are necessary for the validation of that record are also valid at the time of verification. To maintain this temporal requirement, each zone operator must ensure that at any point in time (*i*) secure delegations from the parent zone cover at least one KSK that signs the DNSKEY RRset, and (*ii*) all records are signed by at least one valid ZSK that is included in the DNSKEY RRset. Breaking the first condition causes zone data to be invalidated as insecure, and ignoring the second leads to a bogus state. This temporal requirement must also be maintained during key transitions, just as DNSKEY, DS RRsets, and RRSIGs are undergoing changes. The main challenge, however, is not only to maintain a secure state on authoritative nameservers but to ensure that (*i*) RPs can base validations on RRsets that are cached somewhere in the network and still carry valid signatures, and (*ii*) changes at primary authoritative name servers are not instantaneously observed by relying parties (RPs) [381]. Given a secure data state and a well-functioning RP, we consider a transition as secure if its modifications to keys, signatures, and secure delegations prevent a bogus or insecure state during and after the transition. This definition also accounts for adversaries that can replay valid RRsets that might already be removed from the authoritative name servers and purged from caches. Our following

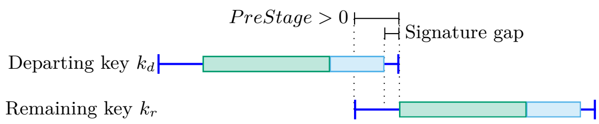


Abbildung 3.6: An example of improperly timed ZSK transition with pre-staged remaining key but belated activation causing bogus validation due to a signature gap.

security analysis considers the relation of a key to its generated signatures, and its relation to other keys and delegation signers.

3.4.1 Temporal Relationship of a Key and its Signatures

Based on our temporal model of DNSSEC keys (see Abbildung 3.3), the following condition applies for a key and its generated signatures: $L_\alpha \leq S_\alpha \leq S_\omega \leq L_\omega$. This condition states that no signature validity period should precede or exceed the total (non-zero) lifetime of the key that generated it. By extension, `DNSKEY` sets that have a valid signature may not be prematurely removed from authoritative name servers without the risk of introducing inconsistencies through caching or replay attacks. The following example illustrates such a situation: an RP caches the `DNSKEY` RRset of a zone and its `RRSIG`(s) at t_0. Before the signature over the key set is expired, the authoritative name server replaces the ZSK at t_1, uses the KSK to generate a new signature over the `DSNKEY` RRset, and finally re-signs all other RRsets in the zone with the new ZSK. At t_2, before either the signature or the `TTL` of the old `DNSKEY` expires, the recursive resolver fetches, for example, `A` records and their signatures generated with the new ZSK from the authoritative name server. At this point, the resolver considers the cached key set as valid (`TTL` spans at t_3) yet fails to validate the retrieved RRset and concludes a bogus state (see Abbildung 3.5). Even if the `TTL` was expired on all caches and the old `DNSKEY` RRset was purged at t_2, an adversary could still poison RP caches with the old key set and cause a failed validation. This clearly illustrates that `TTL`s do not affect the foundational security analyses.

The scenario of failed validation because of bad timing becomes more severe in case of `DS` records. When a newly signed `DS` record is fetched from an authoritative server but the RP uses a departing, cached `DNSKEY` for validation, a complete child zone becomes insecure. This case, however, is more complicated because the only requirement to maintain the chain of trust is to have at least one active KSK securely delegated even if multiple KSKs are present and are actively signing the `DNSKEY` RRset.

3.4.2 Temporal Relationship among Keys and Delegation Signers

The second temporal constraint states that involved keys in a transition must have overlapping lifetimes ($TotalDuration > 0$) as changes on primary authoritative name servers are not instantaneously observable by RPs [381]. Changes to delegation signers must also account for additional delays because as the operator of a zone can only request a modification to `DS` records to a parent zone, with no control over the timing when exactly changes are applied by that parent zone.

To avoid going bogus, a zone operator must guarantee the continuity of signatures over zone data. As DNSSEC allows for multiple signatures over the same RRset at the same time, it suffices that for each RRsets in a zone to have at least one signature that is valid at all times. Transitions that involve multiple keys must make sure that for any RRset signed by a departing key, a remaining key exists which signs that RRset no later than its signatures expire. The most basic solution for this is to activate keys, *i.e.,* create `RRSIG`s, as soon as keys are introduced in the zone ($PreStage = 0$). This way, any active departing key can stop generating new signatures (runout, *i.e.,* $Retire > 0$) and be removed after its generated signatures are all expired ($L_\omega \geq S_\omega$). This approach, however, expands the zone size as multiple valid signatures are present simultaneously during the transition. To address this, keys can be introduced to a zone and start signing later ($PreStage > 0$) to reduce the time window of overlapping signature. Here, the downside is the more challenging timing in making sure that there is no signature gap for any available RRset. Abbildung 3.6 depicts an example where the remaining key is pre-staged but activated too late, thus, leaving a gap during which zone data cannot be validated.

With KSKs, additional care must be given to the timing of `DS` records in order to maintain the chain of trust during the transition and avoid going insecure. In details, this depends on how the KSK is being transitioned: if $PreStage > 0$, the new `DS` record can be added to the parent zone *after* the key is included ($RemPreDs > 0$) but no later than expiration date of previous DS, while the old delegation can be removed before the departing key expires ($DSPreRem > 0$), yet no later than its runout period ends. This approach can be used to minimize the interaction between the zone operator and its parent zone by combining the request to remove the old delegation and adding the new one ($DSOverlap = 0$). If $PreStage = 0$, any combination of *RemPreDS* and *DSPreRem* that does not cause a gap between delegation signers ($DS_\alpha(k_r) \leq DS_\omega(k_d)$) can be applied. This implies that when all signatures of the departing key expire, one remaining active key must still be securely delegated.

3.5 Monitoring System and Data Corpus

In this section, we introduce the DNSSEC measurements taken from our monitoring system [428] that we used to evaluate key transitions in the wild. We give an overview of our monitoring system, describe its operational aspects, and discuss the characteristics of the resulting data corpus, which covers the first 15 years of the global DNSSEC rollout.

3.5.1 Monitoring System

Our monitoring system collects DNSSEC records (`DNSKEY`s, `DS`es, `RRSIG`s) alongside other types of RRsets, network PMTU measurements, name server versions, and many other relevant measurements by polling *all* of *every* zone name servers (as specified by both the zone and predecessor `NS` records) from distributed vantage points across the Internet [428], [188]. This comprehensive polling is a critical feature for observing key transitions with complex process models such as those specified in RFC 8901 [368].

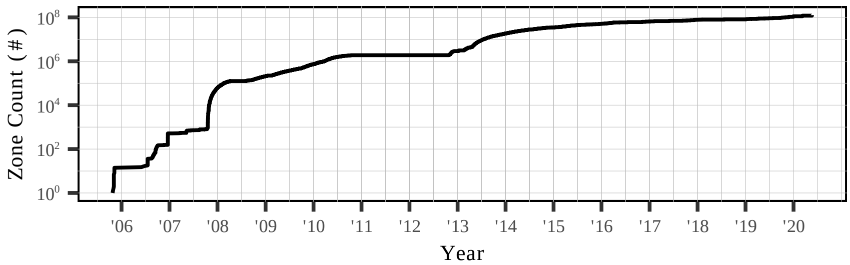


Abbildung 3.7: Total number of monitored DNSSEC-enabled zones.

In order to capture the holistic status of the global DNSSEC deployment, we broadly define zones as being *DNSSEC-enabled* if they have deployed one or more DNSKEY records. The set of DNSSEC zones in this corpus was learned from proactive crawling of online sources, NSEC-walking [379], [187], user-submissions, selected top-lists (Alexa Top One Million [282], the Majestic Million [397], and the Cisco Umbrella Popularity [312]), and other techniques. The development of the number of monitored zones throughout the years is depicted in Abbildung 3.7, and the detailed numbers are presented in Appendix 3.A.

As our automatic discovery methods might catch non-production zones, *i.e.,* secure zones which are deployed for testing purposes, we apply a conservative heuristic to keep only production zones. First, all zones under arpa TLD are marked as production, then any zone that points to an active web or mail server is added to an include-list [188]. The remaining zones are considered as non-production. This way, we keep only zones for which we have positive indications of actually being in production. It should be noted that since there is no systematic mechanism to discover all zones in the global DNSSEC, measurements can be subject to sample bias; nonetheless, we argue that our measurements of over 9.5 million DNSSEC zones at the time of this writing represent a relevant set on which to measure general behaviors of key transitions in the wild.

Our monitoring system polls DNSSEC zones concurrently in four ways:

1. The monitoring system repeatedly queries all zones from each of its remote polling locations. These full measurements poll hundreds of zones concurrently, and when finished, the system starts again. As the number of DNSSEC zones has increased, the full-corpus polling period has grown from taking days to taking weeks, making the periodicity of polling a varying interval.

2. In parallel, it polls the DNS Root zone and all Top-Level Domains (TLDs) once per day, which is half the TTL period of those zones.

3. In addition, it polls popular DNSSEC-enabled domains from our top-list collection every two days.

4. Finally, the interactive website of the monitoring system allows users to poll any zone on-demand.

Noteworthy, though, is that zones that delegate to others are implicitly queried and measured multiple times beyond individual schedule. For example, the `.com` zone is not only monitored once but its keys and their *usages* are also re-observed for every delegated domain below, in order to assess `NS` and `DS` records. For example, the root zone is measured over 1,300 times every day, once for every Top-Level Domain polled. This greatly increases the frequency of observation, albeit in an aperiodic way.

Our monitoring system has occasionally undergone migration to new hosting infrastructures, database backups, and other operational maintenance events during the 15 years, leaving irregular gaps in our data corpus. We control for gaps in our methodology, for details see Abschnitt 3.6.

3.5.2 Data Corpus

This work makes use of data from October 1, 2005 to August 31, 2020. The resulting data corpus encompasses over 30.8 billion DNSSEC measurements from 9,535,615 DNSSEC-enabled zones with 35,882,395 distinct `DNSKEY`s. We observed 58,193,197 points in time when keys were either added or removed from zones. Of those changes, 17,965,575 key transitions were detected (see Unterabschnitt 3.3.1) and analyzed, which we will discuss in Abschnitt 3.8.

Our data corpus spans several events that are notable for the global deployment of DNSSEC. First, multiple Top-Level Domains (TLDs) such as `.com`, `.edu`, and 145 country-code TLDs (ccTLDs) deployed DNSSEC for the first time. Second, the announcement of a crucial large-scale DNS cache poisoning attack vector called the Kaminsky Attack [377], [478], whose remediation was publicized to be the deployment of DNSSEC. Third, the DNS Root zone enabled and rolled out DNSSEC for the first time using a Deliberately Unvalidatable Root Zone (DURZ) key. Fourth, in 2010, the DNS Root zone performed its first ever Key Signing Key (KSK) transition, from the DURZ key to the 2010 KSK. Fifth, over 1,200 DNSSEC-enabled new generic Top-Level Domains were added since 2013. Finally, the Root KSK was transitioned for the second time ever in 2018, after being started and paused during 2017. Abbildung 3.1 depicts a number of these notable events and incidents and by overlaying the measurement periods of related work and this study, shows this work has a uniquely complete longitudinal dataset to draw conclusions from.

3.6 Methodology

The initial step in our methodology is to reconstruct the lifetime of `DNSKEY`s according to our discrete measurements and in accordance with our proposed temporal model (see Unterabschnitt 3.3.1). This is a deceptively challenging step because when keys are provisioned into zones there are no semantics to express (or for zone administrators to even know) life cycle information. In general, key lifetime management and changes such as re-signings or deployment of new keys may occur at varying times between polling cycles. This will lead to gaps in time between three events: (*i*) when these changes happen on the authoritative name servers, (*ii*) when we poll the zones, and (*iii*) when we observe

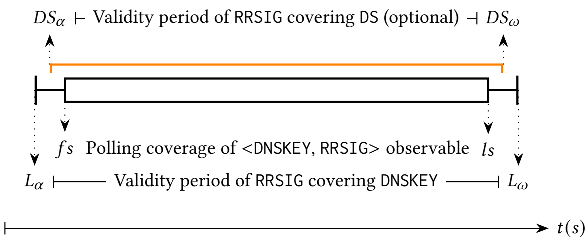


Abbildung 3.8: Visualization of an observable for a given <DNSKEY,RRSIG> tuple.

them in use. This may lead to changes that a measurement system completely misses regardless of the polling frequency. A simple example is the replacement of a DNSKEY multiple times between two polls.

We now give a brief overview of our methodology and then present the key building blocks in more detail.

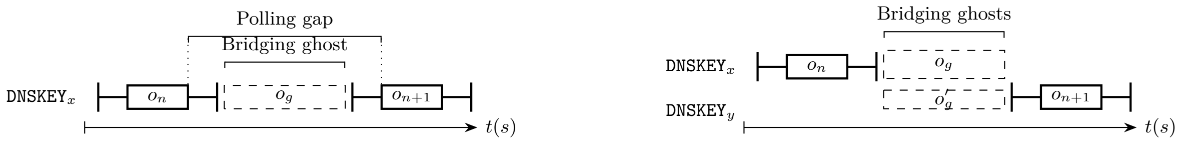
3.6.1 Overview

Our first step in reconstructing the complete and continuous lifetime of keys from our discrete measurements is to infer a so-called *observable* for any unique <DNSKEY, RRSIG> tuple and any respective secure delegation period (if applicable) while preserving points of time when this specific observable was seen during our measurements. If the lifetime of a key is extended by re-signing (*i.e.,* generating a new RRSIG over the same DNSKEY), our methodology creates a new observable tuple. In this sense, an observable can be considered as a piece of evidence based on a given RRSIG denoting that a key has been valid from the inception to expiration dates of that RRSIG. Abbildung 3.8 depicts *one* observable for key k_x. The lifetime of this single observable duration $[L_\alpha, L_\omega]$ is calculated from the signature covering k_x; its delegation period $[DS_\alpha, DS_\omega]$ is inferred from the signature in the parent zone covering DS record(s). fs and ls denote the times when this single observable was first and last seen during the measurements. Note that fs is recorded once per observable, and never changed again, while ls is updated when the same <DNSKEY, RRSIG> tuple is encountered again in a subsequent measurement.

Individual observables expand the information about a key lifetime from a discrete snapshot to a continuous timeline within the validity period of its covering RRSIG. Such extrapolation of observables, however, might still leave some *gaps* in our continuous lifetime model. Furthermore, other life cycle information, such as key usage (see Abbildung 3.3), cannot be inferred from RRSIGs over keys but must be measured from signatures *generated* by those keys when they are in use. To address this we introduce a novel three-step methodology that we call *Bridging, Busting,* and *Binding*:

Bridging Extend observables by filling in measurement gaps with place-holder observables, which we call bridging "ghosts".

Busting Use collected evidence, *e.g.,* RRSIGs over non-DNSKEY records, to remove (or "bust") incorrect ghosts.



(a) Bridging between observables of the same `DNSKEY`

(b) Bridging between observables of different `DNSKEY`

Abbildung 3.9: Filling in observation gaps with bridging ghosts.

Binding For any given `DNSKEY` combine remaining contiguous observables into a continuous holistic life cycle model.

This process extends sets of observables into full key life cycles and builds a basis to calculate related statistics such as signing frequency, measure management errors, and compute other aggregate behaviors. It also accounts for gaps in our data corpus.

3.6.2 Bridging

Bridging begins by time-ordering observables for each zone. If the maximum time of a single observable (*i.e.,* $max(L_\omega, ls)$) is less than the minimum time of the next observable (*i.e.,* $min(L_\alpha, fs)$, then a *ghost record* is inserted, which exactly covers the missing time. Note that a ghost does not necessarily fill the whole gap between two polling cycles, but only the time for which no temporal information can be inferred (see Abbildung 3.9a). If the ghost observable covers a gap between the observables of the same `DNSKEY` (but different `RRSIG`s), then the ghost observable proposes that the key existed continuously during the gap, as shown in Abbildung 3.9a. If, on the other hand, the gap exists between observables of two different `DNSKEY`s, then the process cannot know precisely when the older `DNSKEY`$_x$ stopped being present and the newer `DNSKEY`$_y$ was deployed, if or when they may have overlapped, or if there was a period of no key(s) being present. Therefore, at this stage, a bridge is *temporarily* used, whereby a trailing ghost for `DNSKEY`$_x$ is inserted until both the next observable and a leading (overlapping) ghost for `DNSKEY`$_y$ are inserted, starting just after the previous observable, as seen in Abbildung 3.9b. Ghosts represent optimistic assumptions about consistency between observations, but in the next phase we *bust* ghosts if additional evidence proves they are incorrect or need to be adjusted.

3.6.3 Busting

Ghost observations model place-holders of inferred data that may have existed between the data we observed. For each zone, after the initial optimistic Bridging phase, our process begins to examine keys in relation to each other and incorporating additional evidence to detect if a ghost assumption can be refuted (and thereby *busted*). For this, additional measured data (such as `NS`, `SOA`, `A`, and associated `RRSIG`s) allow the process to determine if, or how, a ghost should be busted. For example, if a ghost bridges a key, but another key was seen during that time, we can determine the ghost-key was not

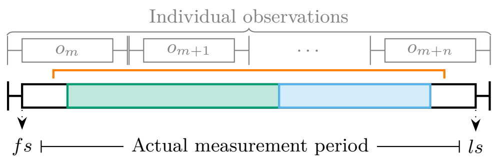


Abbildung 3.10: Observations are bound together to find minimum and maximum timestamps for inception, expiration, first seen, and last seen. Measuring bounds also seek evidence of when keys were used to generate signatures (green region) and how long their signatures were valid for (blue region).

present for the period when the other key was seen. The ghost is, then, busted by truncating it to the time interval(s) that are not covered by the other key. Alternately, if ghosts for two keys overlap between a transition (see Abbildung 3.9b), information about which of them was able to verify signatures over other measured data is used to determine when one key was removed and the other was present, and truncate ghost overlaps accordingly. Overall, ghosts may be truncated if a zone's data was observed and the key was absent, and affirmed if a ghost's key was seen to be signing other records. Ghosts which may have been causes by measurement outages are removed altogether. Gaps caused by outages are distinguished through their relatively long duration. For our purposes, we take the yearly statistics of our measurement system crawl times, *i.e.,* a complete round of polling all zones, to bust gaps caused by outages: for any given year all gaps that are larger than the mean crawl time plus four standard deviations are considered as an *outage gap*, are busted and removed. While the removal of long-ghosts could result in missed key transitions and key management behaviors, not removing them could alternately enshrine inaccurate assumptions.

3.6.4 Binding

For every key, all the contiguous observations (including unbusted ghosts) are used to *Bind* observables into one single continuous key, as seen in Abbildung 3.10. Bound keys describe the life cycle of a DNS key in terms of our temporal model (Abbildung 3.3) with two additional features. The first timestamp that the key was measured (fs) and the last time a key was observed (ls). Here, all the observed `RRSIG` records that could be verified by each key, regardless of which DNS record type they covered, are used to quantify whether keys were in active use. The inception timestamp of each `RRSIG` is used as evidence to indicate when a key pair was signing data ($[S\alpha, S\phi]$). The expiration of that signature specifies the run-out, or the duration that a key's data was verifiable while no new signatures were detected ($[S\phi, S\omega]$).

3.7 Classifying Keys and Transitions

Using our model of continuous life cycles for `DNSKEY`s, we move on to classification of `DNSKEY` life cycle states and key rollovers and more complex key transition types.

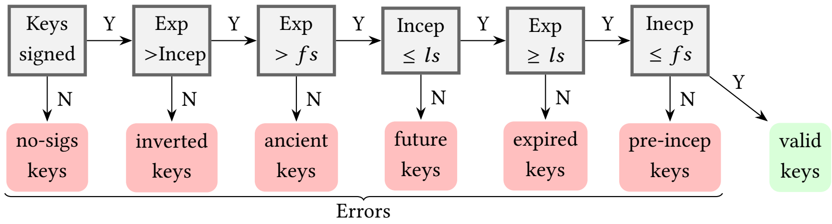


Abbildung 3.11: Classification of key life cycle management states. Each state results from an observable property (rectangle) that is either fulfilled (Y) or not fulfilled (N). Properties include expiration (Exp) and inception (Incep) dates, as well as first (fs) and last seen (ls) of operational usage.

3.7.1 Key Life Cycle Classification

We present our novel classification scheme of measurable errors in key life cycle management in Abbildung 3.11. Based on this model we observe previously undetected errors in live deployments, which we explain in more detail in Abschnitt 3.8.

This classification scheme defines six types of key life cycle management errors depending on when a key is introduced in the zone and how inception and expiration dates of the respective `RRSIG` are defined. For example, if the expiration predates the inception date we classify the key as *inverted*, or if key was observed before its actual inception date, we label it as a *future* key. Avoiding these errors is a necessary (yet insufficient) precondition for *valid* cryptographic protections.

3.7.2 RFC-Based Classification

The IETF has specified many aspects of how DNSSEC zones and data should be configured and maintained across numerous RFCs [350], [381], [413], [464]. Among those are several processes that model ways in which `DNSKEY`s should be rolled over. In RFC 5011 [464], Trust Anchor (TA) rollover is specified for zones whose predecessor zones do not securely delegate `DS` records. Additionally, in RFC 7583 [413], processes are described for how zone administrators should transition their ZSKs and KSKs.

First we characterize RFC guidelines in terms of our transition model and then investigate how transitions in the wild conform to these guidelines (in Abschnitt 3.8). We classify deviations as either warnings, *i.e.*, the behavior does not strictly follow the RFC guidelines, yet, does not disturb validation; or as errors, which render the validation as bogus or insecure. It should be noted that while conventional IETF parlance (*e.g.*, `MUST` vs. `SHOULD` in RFC 8174 [390]) often makes this distinction explicit, the measurable quantities in RFC-7583 [413] do not use `MUST, MAY, SHOULD`, etc. We, therefore, semantically assign these values based on the overall processes. For example, if a transition was specified as needing to have $Retire > 0$, but it was 0, this would only be a warning because the key transition would still allow keys to verify data for a zone. In contrast to this, if a KSK must be present *before* a

Tabelle 3.2: Mapping of key transition processes specified in DNSSEC RFCs and other literature to Transition Anatomies. In each row, gray cells show the mandatory features that must be fulfilled to map a transition to a predefined class, whereas the other cells describe soft constraints, which only cause warnings if not followed exactly. Cells showing "–" indicate wildcards.

		Features of the Transition Anatomy									
	Transition Class	*PreDS*	*DoubleSig*	*PreStage*	*DepSigOnly*	*Retire*	*DSOverlap*	*RemSigOnly*	*DSPreRem*	*RemPreDS*	*TotalDuration*
Based on RFCs	*ZSK Pre-Pub*	–	$= 0$	> 0	> 0	> 0	–	> 0	–	–	–
	ZSK Double-Sig	–	> 0	$= 0$	$= 0$	$= 0$	–	$= 0$	–	–	–
	KSK Double-DS	< 0	$= 0$	$= 0$	$= 0$	$= 0$	> 0	> 0	< 0	< 0	–
	KSK Double-KSK	> 0	> 0	$= 0$	$= 0$	> 0	$= 0$	> 0	> 0	> 0	–
	KSK Double-RRset	> 0	> 0	$= 0$	$= 0$	> 0	$= 0$	> 0	$\neq 0$	–	–
	KSK Update-TA	–	–	–	–	–	–	–	–	–	$\geq 30d + AvgSig$
Based on [256]	*KSK Emergency 2*	–	$= 0$	$= 0$	$= 0$	$= 0$	$= 0$	> 0	–	≥ 0	–
	KSK Emergency 3	–	$= 0$	> 0	–	> 0	–	–	> 0	> 0	extended
	ZSK Emergency 2	–	$= 0$	$= 0$	$= 0$	$= 0$	–	> 0	–	–	–
	ZSK Emergency 3	–	–	> 0	–	–	–	–	–	–	extended

DS record in order to let resolvers securely verify the KSK during a *Double-DS* transition, then *PreDS* ≤ 0 is a critical error.

RFC 5011 (Update of Trust Anchors). This transition is specified by a Finite State Machine with timers and a rigorous process model [464]. The process-model, however, is specified from a resolver's perspective (*i.e.,* timers that a resolver should set internally), which do not always directly correlate to observable timing of DNSKEYs in authoritative zones. Additional guidance [358] was written for authoritative zone administrators, which lends itself more directly to being measured. These two publications [358], [464] define the specification for RFC 5011 transitions as those whose *TotalDuration* $\geq 30days + min(15d, \frac{1}{2} \times TTL)$, where 30 days are specified in [358] and the key's Time-To-Live (TTL) is specified as an additional component of the period. Additionally, those keys that are being removed must be revoked *and* be used to sign their own revoked DNSKEY set. In order to be conservative and permissive, we model the upper-bound (*i.e., max* instead of *min*) from each key's own average signature period. We conservatively modeled those zones that did adhere to RFC 5011 timing, but *did not* revoke as still being RFC 5011 compliant, but flagged them with a warning.

RFC 7583 (DNSSEC Key Rollover Timings). RFC 7583 specifies the process models for both ZSKs and KSKs, and they are more stringent. For ZSK, these transitions are defined as either being "Pre-Publication" or "Double-Signature"; and for KSK as one of "Double-DS," "Double-KSK," "Double-RRset," or "Double-RRSIG". Note that the final type is identified as unrealistic in the RFC, and not even fully described there; therefore, we also omit it here.

Tabelle 3.2 summarizes how RFC-based classifications are translated into our transition anatomy from Abschnitt 3.3. An illustrative example is provided in Abbildung 3.12, depicting an excerpt of transitions for a representative zone, `4d.cz.`. As is shown, the zone performs a correct ZSK pre-publication transition while (at other times) performing double-KSK transitions *with warnings* (see Tabelle 3.2). Other examples of key transitions are visualized in Appendix 3.C.

3.7.3 Non-IETF Prescriptions: Emergency Key Transitions

In addition to the above guidance from RFCs, other prior work by Wang *et al.* [256] proposed an approach for conducting emergency key transitions. There, the authors present 10 candidate process prescriptions for emergency key transitions, which are distinct from conventional RFC guidance. It is noteworthy that this prior work considers keys transitions to stand-by keys which are envisioned to be perpetually present before the emergencies. This, therefore, does not precisely specify the *TotalDuration* threshold, but suffices to make the "extended" duration feature a necessary detectable discriminator in these emergency transitions classification. Table 3.2 shows how the timing constraints detailed in that work can be used to classify and detect these events specified by our anatomy and transition methodology, just as with RFC guidance. Due to space limitations, we only include Emergency Transitions 2 and 3.

Additionally, aspects of that work use features of our anatomy and our model of key life cycles that the RFC processes did not (specifically, the TotalDuration and relative ages). For example, the specifi-

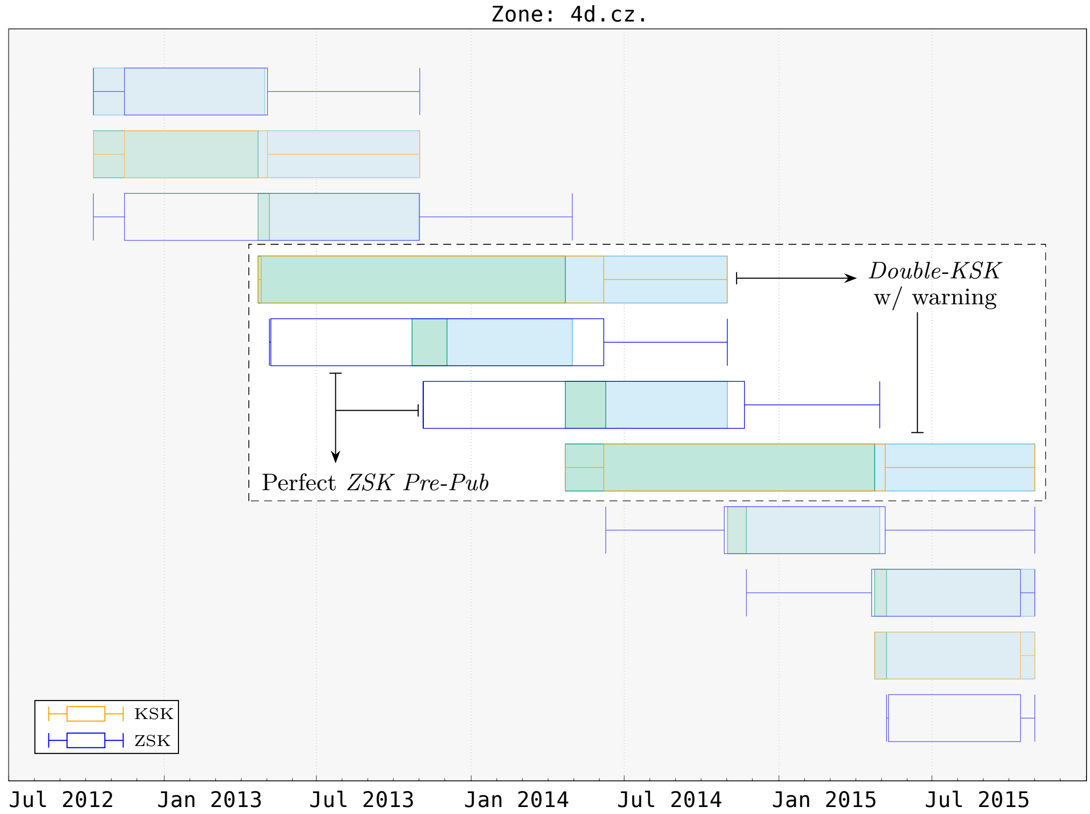
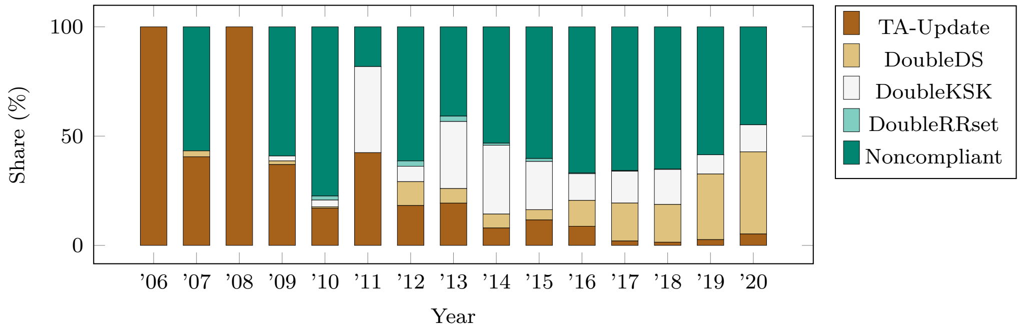
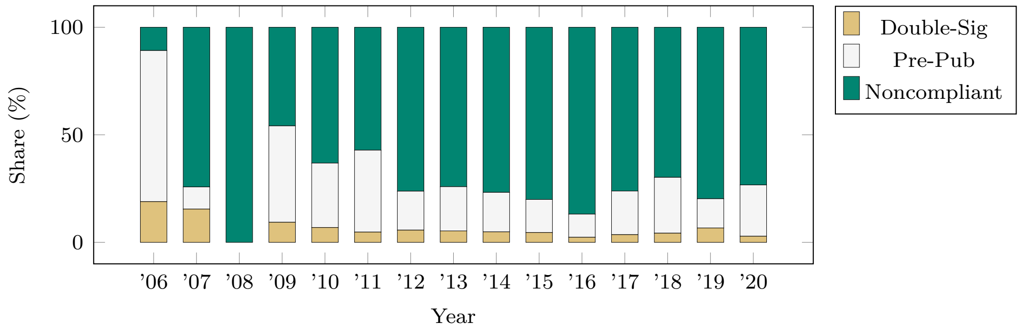


Abbildung 3.12: An example from our measurements showing how `4d.cz.` zone performs a perfect *ZSK Pre-Pub* transition and a *Double-KSK* transition with warnings (*Double-Sig* $\not> 0$ and *Retire* $\neq 0$).



(a) Key Signing Keys (KSK)



(b) Zone Signing Keys (ZSK)

Abbildung 3.13: RFC-based classification of key transitions for in-use KSK and ZSK.

cation requires remaining keys to be *newer* (see *relative ages* in Unterabschnitt 3.3.1), *ZSK Emergency 2* mandates the feature-set of the behavior-based classification *Multi-Signatures* (described below, in Section 3.7.4), and *ZSK Emergency 4* requires *Cutover*s.

In summary, using the prescriptions of prior work, we detected 49,894 emergency ZSK transitions (20,919 that were transitioning *back* or *aborting* transitions to older or same age keys), and 1,780,984 emergency KSK transitions (149,406 transitioning *back* or *aborting* to older or same age keys).

3.7.4 Behavior-Based Classification

In our behavior-based classification, we classify key transitions as being "Multi-Signatures," "Co-Present," "Cutovers" (with degrees of certainty), or "Unknown". This classification approach uses more holistic considerations of all keys in a transition (not just pair-wise) while ignoring the relationships to DS records. Here, key transitions were classified based on (*i*) the type and the total count of overlaps they had with *all* other *active* keys, and (*ii*) whether they were used to verify zone data during their transitions.

When more than one key was seen to be actively in-use (verifying signatures) at any given time, we classified the transition as a "*Multi-Signature*" transition. This indicates that redundant verification existed for data in a zone during the transition. If, on the other hand, keys were observed to have overlapped, but we did not observe *any* of them in use, we classify the transition keys as being "*Co-*

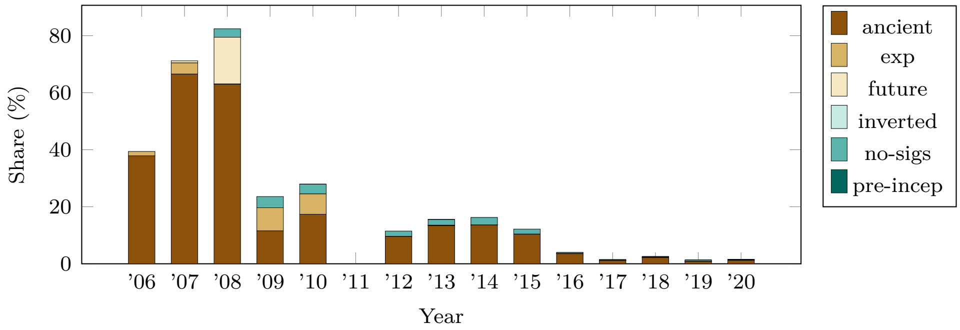


Abbildung 3.14: Classified key management errors.

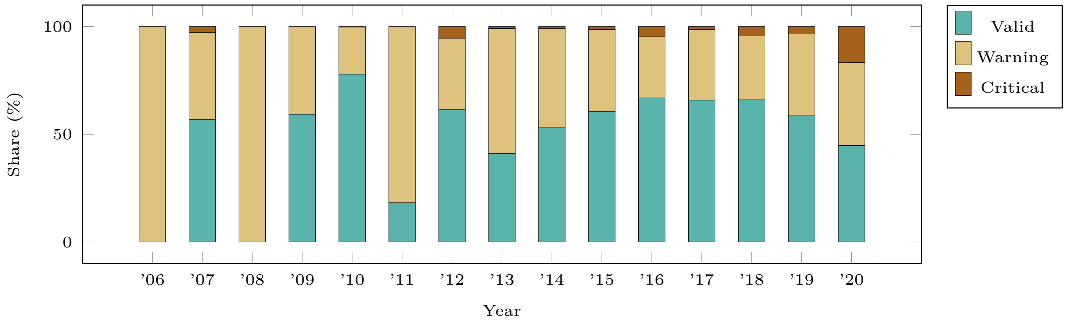
Present." This classification does not represent evidence that keys went unused, but indicates that our observations did not detect usage. Conversely, transitions are classified as "Cutovers" when a single key was observed to transition (or cut-over) to another single key, and they were seen to have been used to verify signatures. This type of transition depends heavily on measurement frequency (whereby the longer the gaps in observations, the more information is estimated from ghost records). Because of this, we further sub-classify Cutovers as "Cutover," "Likely-Cutover," or "Candidate-Cutover." The differences between these are based solely on how much usage (*i.e.,* active signing) was directly observed. If active signing (the inception of `RRSIGs`) was observed for the departing key right up until the remaining key began to be used to verify `RRSIGs` (and not after), then we classify this as a "*Cutover*." If, on the other hand, we did not observe new signatures, but the signature run-out (see Abbildung 3.10) of the departing key overlapped with the signatures of the remaining key, we classify the transition as a "*Likely-Cutover*". Finally, if the departing key was used at any point, and later the remaining key began being used with no other signatures seen in the time-gap, we classify this as a "*Candidate-Cutover*." The goal of these distinctions is to make our sub-classification results useful, but to also preserve their transparency.

3.8 Longitudinal Analysis of 15 Years of DNSSEC Key Transitions

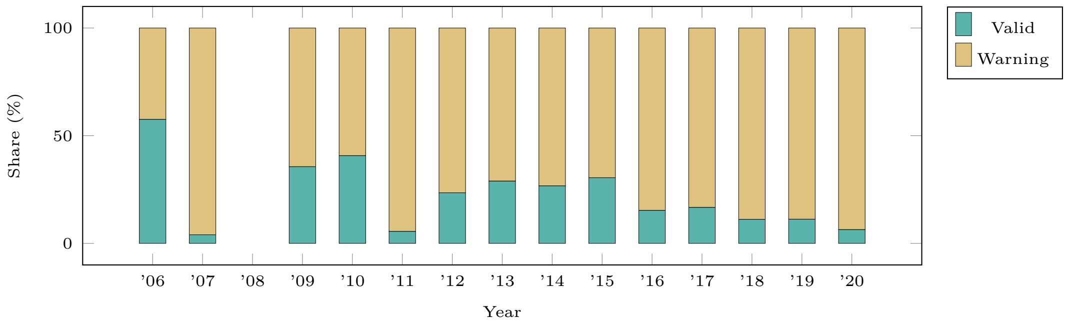
In this section, we present the results from answering the following three questions: (*i*) Do keys involved in transitions follow proper life cycle management policies? (*ii*) How does the anatomy of key transitions in practice compare with RFC guidelines specifying rollovers? (*iii*) Which characteristics are popular in key transitions beyond key rollovers in terms of our behavior-based classification?

3.8.1 Key Life cycle Management

The atomic management of the keys can end in errors, independent of transitions of other keys. Using the key life cycle management classification described in Abschnitt 3.6, we broadly examine the rates at which individual `DNSKEYs` follow proper key management life cycles. Our results are summarized in Abbildung 3.14. Key life cycle management errors were clearly highest during the early years of the



(a) Key Signing Keys (KSK)



(b) Zone Signing Keys (ZSK, excluding type noncompliant)

Abbildung 3.15: Share of valid, warning, and error rates of in-use KSK and ZSK.

DNSSEC global rollout. At that time, the tools that supported DNSSEC were in their nascent stages of maturity, which is very likely the core reason for larger rates of key life cycle management errors. This figure also illustrates that the error rates for keys were highest in 2008, just as the discovery rate of new DNSSEC zones surged and more than doubled the size of the global deployment in just a few months. This all correlated in time with publicity to remediate the newly announced Kaminsky vulnerability [478] by deploying DNSSEC, which could have also correlated with inexperienced operators making operational errors in a rush to secure their deployments.

3.8.2 Conformance to RFC Guidelines

KSK and ZSK Transitions in the Wild. Abbildung 3.13 summarizes RFC-based classifications of both measured KSKs and ZSKs per year. Any transition that could not be classified as RFC-compliant is marked as *noncompliant*. This figure confirms a common expectation but also reveals new insights. Unsurprisingly, in the early DNSSEC deployment KSKs were transitioned according to RFC 5011 (denoted here as “TA-Update”). This was necessarily the case because at that time there were very few registrars and also few parent zones that were able to offer secure delegations. Most Top-Level Domains did *not* deploy DNSSEC and were, therefore, unable to securely delegate.

In 2008, operational advice was given to deploy DNSSEC to counter the Kaminsky vulnerability [478]. Our data (see Tabelle 3.A.1 and 3.B.2) indicate that the number of DNSSEC-enabled zones more

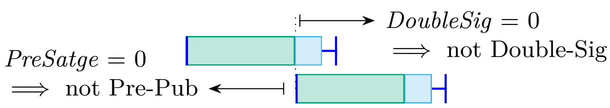


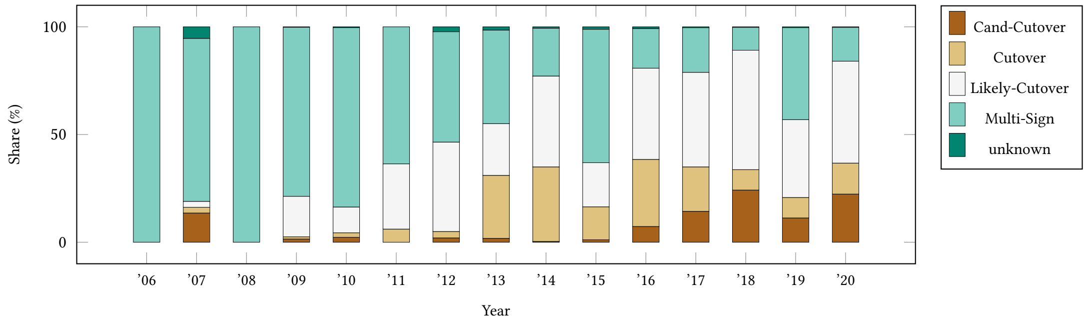
Abbildung 3.16: An example of RFC noncompliant transition performed among others by Root and `.com` zones

than doubled during three months. Based on our classification, it is clearly visible that subsequently more variety in key transition techniques appeared (see Abbildung 3.13a). In 2009, the Double-DS transition technique was the most popular. This technique, however, requires additional coordination between an authoritative zone and the operator of its parent zone. Despite an increasing deployment over time, our results illustrate that the popularity of managing transitions based on Double-DS and Double-KSK fluctuates. The majority always conformed to either RFC 5011 or was "noncompliant." This observation indicates that the RFC-specified key transition processes may not properly represent operational behavior.

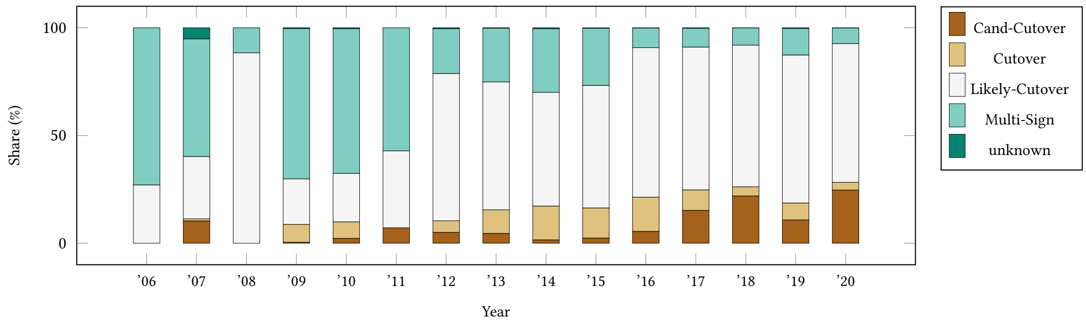
Our results also exhibit an interesting discrepancy compared to related work. In the longitudinal study of Chung *et al.* [45], no Double-DS transitions were reported. However, under `.com` alone, we observed 17,126 unique zones performing Double-DS transitions during the time period of their daily scans (from 2015-03-01 to 2016-12-31), and 7,256 transitions during the same period as their hourly scans (from 2016-09-29 to 2016-12-31). A concrete example of Double-DS transitions is given in Abbildung 3.C.2e (`willemvanveldhuizen.com`). Further understanding the discrepancies of these findings will be part of our future work.

In comparison to KSK transitions, an even larger portion of ZSK transitions have constantly been nonconforming to RFC specifications (see Abbildung 3.13b). While the "Double-Signature" alternative never accounted for more than 10% of transitions, it is most noteworthy that between 2007 and 2020 (except for 2009) the majority of the observed ZSK transitions did *not* conform to either prescribed mode of key transition. Ignoring RFC guidelines, however, does not necessarily mean that the zone verification would fail during the transition. Abbildung 3.16 depicts an example of such transition performed by prominent zones such as the Root and `.com` zones. This transition is neither Pre-Pub ($PreStage \not> 0$), nor Double-Sig ($\text{DoubleSig} \not> 0$), yet at any time during the transition the RRs can be verified by either or both keys. Here, we also see discrepancies with prior work [45], which might be traced back to different approaches used in classifying transitions.

Implications on Robustness. The warning and error rates for RFC-compliant transitions are notably different for KSKs and ZSKs, as depicted in Abbildung 3.15. KSK transitions show more valid cases than ZSK transitions overall. This corresponds also with a higher share of transitions following RFC guidelines (see Abbildung 3.13a). Later, between 2011 and 2020, critical error rates became even more prominent. This could be the result of inherent management complexity in terms of timing and data for (longer) chain-of-trust-based transitions between authoritative zones (`DNSKEY`s) and parent zones (`DS` records).



(a) Key Signing Keys (KSK)



(b) Zone Signing Keys (ZSK)

Abbildung 3.17: Behavior-based classification of key transitions for in-use KSK and ZSK.

For ZSKs, Figure 3.15b shows that the rate of warnings exceeded successes in every year, except in 2006, when the DNSSEC deployment was just beginning, and zones were deploying DNSSEC for the first time. In 2007, warning rates greatly exceeded success. In 2008, we observed only noncompliant transitions (see Tabelle 3.B.2), and from that year on the trend was increasing for success rates. In 2011, an outage in our monitoring system resulted in fewer observed transitions, and the success rates of that smaller set may have been skewed. Success rates dropped precipitously, mainly due to From 2012, after a brief improvement, we also observe decreasing success rates. This, however, again correlated with a large increase in the discovery rate of new DNSSEC-enabled zones. During this time, the DNS Root was deploying DNSSEC and transitioning its KSK for the first time. This correlation between a large increase in newly deployed DNSSEC zones and increase in error rates suggest that these rates may also be due to operators who were learning how to manage the security postures of DNSSEC in their DNS zones.

3.8.3 Holistic Characteristics of Transitions

Using the RFCs to classify key rollovers gives us a helpful start in understanding how to analyze key transitions. However, to overcome the amount of noncompliant and high error rates, we use the alternate behavior-based classification scheme introduced in Abschnitt 3.6.

When applying our behavior-based classification to ZSKs that were seen to be in use, Abbildung 3.17b shows that almost every key transition observed for 15 years could be classified. Only in 2007, 4.32% transitions were classified as unknown; in all other years, >99% of transitions were classified successfully. The trend over the first 15 years was a shift from Multi-Signature, to cleanly cutting over from a departing key to a remaining key.

Abbildung 3.17a illustrates that all of these trends were similar in the KSK transitions. A greater portion of the KSK transitions were Multi-Signature than in ZSKs, but the trend towards moving to Cutover (and the small incidence of unknown transitions) mimics the ZSK transitions. This could be due to the same operational approach being used for both ZSKs and KSKs by zone operators, but the DNSSEC tools available have traditionally offered more comprehensive automation for ZSK transitions, since they are managed within a single authoritative zone. By contrast, KSK transitions involve coordination with the parent zone.

3.9 Related Work

The topic of DNSSEC key rollovers has recently appeared in the literature. In 2017, the DNS Root zone began publicity around its, then, intentions to transition its KSK for the second time (since DURZ, as discussed in Abschnitt 3.5). At that time, van Rijswijk-Deij *et al.* [203] began the Root Canary project to track this specific transition. Later, Müller *et al.* [173] created a related monitoring tool to aid operators in successfully planning and conducting `DNSKEY` rollovers. More recently and maybe most closely related to our work, Müller *et al.* [174] conducted an in-depth study of key life cycle management in the DNS Root zone. Both of these results synergize with the observations of our work: key life cycle management of DNSSEC zones is critical to their operational health. Most importantly, and in contrast to prior work, we do not only consider key rollovers but provide a framework to analyze key transitions in general.

As DNSSEC deployment has grown, numerous studies have incrementally tracked its deployment. Among the first measurement studies, Osterweil *et al.* [188] observed that longitudinal measurement of DNSSEC deployment were critical to understand its health, and proposed a set of three metrics to summarize the status of the global deployment. Subsequent large-scale measurements [146] reported to confirm earlier findings, but also began to raise concerns about validating resolvers and the limited number of users being protected by DNSSEC.

More recent work [45] has conducted large-scale longitudinal analysis of the global DNSSEC deployment. Therein, the authors examined two years of DNSSEC deployment data and addressed the identification of key management errors as an area for concern. The authors also discussed the subject of recycled keys as key “sharing” and flagged the behavior as an error. Recycled keys are those that were shared in separate zones or used, removed, and then re-used. DNSSEC zones are often presumed to create distinct keys for themselves without sharing usage with other zones, and that once a key expires and completes its operational lifetime, it will not be used again. In our paper, we treated each appearance of a recycled key as a new key, because when being re-used its operational life cycle is different. From our corpus we identify 35,882,395 distinct DNSKEYs, 42,908,290 distinct DNSKEY/zone pairs, and 54,337,697 total operational lifetimes of keys. Our measurements confirm the earlier result, and further illustrate that a non-trivial number of DNSKEY records were shared between zones, and others may have been returned to service after completing an initial operational lifetime.

While recent related work signals a renewed interest in key transitions, previous literature exists that suggests augmenting the DNSSEC protocol to add explicit semantics that indicate ongoing key transitions. Guette *et al.* [98] suggested an extension to the DNSKEY format itself to indicate when key transitions are underway. In subsequent work [97], this approach was evolved by proposing the new Resource Record KRI. Interestingly, the semantics that those works suggest as being necessary are already observable in the current DNSSEC, when using the methodology we introduced in this paper. Explicitly exposing key transitions in the DNSSEC may have security implications, though. Nawrocki *et al.* [177] show that the presence of multiple keys in the DNS gets systematically exploited in DDoS attacks.

The DNSSEC algorithm rollovers, an aspect that is outside the scope of this work, has been studied in the past [45], [173]. In a recent publication, Müller *et al.* [175] study the lifetime of cryptographic algorithms for DNSSEC to conclude that *algorithm agility* has already been reached for DNSSEC.

Finally, the role of machine learning in assessing security aspects and detecting various attacks, which has been discussed in the related work, can also be useful for our future work specifically with respect to noncompliant transitions. Jin, Tomoishi, and Matsuura [116], for example, make use of machine learning to detect cache poisoning in DNS, specifically those caused by compromised name servers. In the context of Web PKI, Dong, Kane, and Camp [63] define a set of features to describe X.509 and apply deep neural networks to detect rogue certificates. And finally, in general context of relational data, Lou *et al.* [151] proposes a method to cluster related data.

3.10 Discussion

The global rollout of DNSSEC is flourishing, and is giving operators experience at scale in managing cryptography in a core Internet protocol. We believe that now is the right time to investigate what has been (and can be) learned from these experiences.

Capturing the right security model. Caching plays an important role in DNS(SEC) because it enables scalability and availability. This service is controlled by TTL values of records. From a security perspective [283], when changing DNSKEYs, care must be taken to provision consistent DNSKEY and RRSIG material so that what can appear in caches remains verifiable at all times (see Abschnitt 3.4).

The DNSSEC *availability* protections are important but distinct from its data *integrity* assurances. Availability is entirely governed by TTL, whereas the integrity protection is entirely governed by DNSKEYs, RRSIGs, and DSes and their timings and cryptographic life-times. TTL-based availability protections (designed for caching) are not involved in integrity assurances. Conversely, integrity protections actually mollify cache poisoning attacks, which was a central design goal of DNSSEC [377]. In such attacks, availability is not a factor because adversaries influence the presence of data in caches. Though DNSSEC operations involve the confluence of these two aspects of DNS(SEC). In this work, we focused on the DNSSEC integrity protections and stress the importance of evaluating their exclusive role. Any changes to RRs can succeed within the temporal constraints imposed by DNSSEC (see Abschnitt 3.4), and in turn, how caches regard `TTL` values has no impact on security guarantees provided by DNSSEC. We proposed that the necessary conditions to preserve integrity protections during a DNSSEC key transition are:

1. Zone KSKs must be covered by valid and verifiable `DS` records to establish a chain of trust.
2. `RRSIG`s covering current data (`DSNKEY` and other RRSets) must be verifiable by at least one authorized `DNSKEY` at all times.

These conditions together are sufficient to provide data integrity assurance in the DNSSEC during key transitions. We used our proposed anatomy to describe how to evaluate these in operational deployments. We also discussed (Abschnitt 3.4) the security implications of key transitions both in terms of the relationship of a key to its own signatures, to other keys, and to delegation signers in case of KSK. Our proposed temporal constraints do not only allow for a formal security analysis of transitions but can also be used to design and validate software used by authoritative name servers to manage key transitions.

Sampling frequency versus TTL-level polling. There is debate in the literature on the general topic of whether to monitor `DNSKEY` transitions at the granularity of DNS `TTL`s or signature lifetimes. Specifically, work by Chung *et al.* [45] considers `TTL` to be the primary discriminator of change frequency. We argue that when conducting a key transition, changing zone contents (`DNSKEY`s or otherwise) before their definitive `RRSIG`-based lifetimes have elapsed (*e.g.,* at the granularity of `TTL`s) exposes zones to vulnerabilities (*e.g.,* replay attacks), as discussed in Abschnitt 3.4. This holds because the security guarantees of DNSSEC are orthogonal to `TTL`s and caching because the TTL values cannot ensure the absence of replayed (possibly compromised) keys into validating resolver caches. More specifically, an adversary can replay compromised keys whose `RRSIG`s are still valid, regardless of their `TTL`. For example, if a zone private key was compromised and the operator of the zone was to replace it with a new key (*i.e.,* remove the old key immediately or when `TTL`s have expired) a zone would still be vulnerable to replay attacks. While this does not stop operators from performing key transitions like this, one of the objectives of our analyses was to demonstrate the relationships (and gaps) between protection and practices.

While our analyses illustrate weaknesses, our measurement corpus actually includes observations taken at half the `TTL` values of the DNS Root zone and all TLDs whose TTLs are all two days. We posit that one-day polling of these zones is sufficient based on the Nyquist-Shannon sampling theorem [182], [221]. Measuring continuous phenomena with discrete sampling can approximate those phenomena and accurately characterize them, if polling occurs at a frequency that is at least twice the rate of change. This substantiates our three conclusions. (*i*) Key transitions need to be measured at frequencies relative to `RRSIG` lifetimes. (*ii*) Properly operated infrastructures (*e.g.,* the Root and TLDs) perform their operations in accordance with these analyses. (*iii*) This work is able to compare these different timing hypotheses (*i.e.,* RRSIG vs. TTL) using longitudinal measurements from the wild.

Finally, it should be noted that discrete measurements, regardless of the measurement frequency, are going to inevitably miss alterations that happen between subsequent measurements. With regard to our measurements, this means that violations to temporal constraints (Abschnitt 3.4) might be overseen as we construct a continuous model using the binding, bridging, and busting method (Abschnitt 3.6).

Key transitions are complex. The anatomy of key transitions in the wild shows a large diversity in the different ways that zone administrators are deploying them. This diversity has often not conformed to those prescriptions set down in RFCs, but is that a problem of the zone owners (and their software) or with the standards? Based on the ≈18 million transitions that we observed in this work, a more foundational question arises: Are operator practices exposing correctable security problems or are their implementations displaying insights that should be ingested—similar to the Continuous Improvement paradigm [318], which has been applied recently in other areas [23]? This question serves as an additional motivation for the behavior-based classification approach we described in Section 3.6. We believe that defining an anatomy and measuring related features can serve as a rigorous methodology, which may give rise to a sound feedback loop between standardization and operational practice.

3.11 Conclusion and Outlook

In this work, we defined the *anatomy of key transitions* as ten separate, measurable timing features based on a temporal model for DNS keys. In addition, we identified related measurable aspects of key life cycle management (*e.g.,* relative age and key management errors), and defined a holistic behavior-based classification method for transitions. We then introduced a novel methodology for converting discrete observations into continuous DNSKEY lifetimes, named *Bridging, Busting, and Binding*. Using this substrate, we created anatomies of key transitions from the wild and empirically evaluated this complex phenomenon for the first 15 years of DNSSEC operation.

Our work allows us to measure how well operators have followed guidance and, as an example of the general utility of our methodology, where related work [256] has applied. More generally, we were able to classify the kinds of key transitions operators have been using. We found that the majority of KSK key transitions and the vast majority of ZSK key transitions do not conform to RFC guidance. Additionally, using our behavior-based classification, we also observed that in some years significant rates of key transitions were rolling backward. We conclude that the anatomy and methodology of this work serve as a useful platform for investigating DNSSEC key transitions in the wild, and results are well suited to inform evolving key transition practices.

As operational guidance, we advise operators should (*i*) aim for transition methods that reduce dependencies on parent zones and (*ii*) use the analyses of this paper to help verify that RRsets covered by keys remain verifiable when seen from external vantage points. Public recursive resolvers may serve as such vantage points to verify the state of cached entries.

Going forward, we intend to examine the specific implications of popular key transitions, to understand where deployed innovations can advise security standards. To better evaluate and understand noncompliant transitions, it is also possible to make use of statistical approaches common in artificial intelligence and machine learning to cluster similar transitions [151] or even to engineer and select new features beside the ten introduced in this work. Further, we intend to investigate the applicability of using our anatomy for other large-scale object security systems, such as the Resource Public Key Infrastructure (RPKI) [391] and the Web PKI [370]. Finally, we intend to make our feature set and the longitudinal dataset from [428] public, to encourage additional investigations from the community.

Acknowledgments. This work was supported by the Commonwealth Cyber Initiative, an investment in the advancement of cyber R&D, innovation and workforce development. For more information about CCI, visit `cyberinitiative.org`. Additional support in parts was provided by the German Federal Ministry of Education and Research (BMBF) within the projects *Deutsches Internet-Institut* (grant no. *16DII111*) and *PRIMEnet*.

Appendices

3.A Raw Data of Monitored Zones

The number of DNSSEC-enabled zones that we have monitored has constantly been increasing in the past 15 years. Abbildung 3.A.1 depicts the growth of secure zones for various sources in our data corpus. Tabelle 3.A.1 tabulates the absolute values per year and source.

Tabelle 3.A.1: Total number of monitored DNSSEC-enabled zones per source.

	Year															
	'06	'07	'08	'09	'10	'11	'12	'13	'14	'15	'16	'17	'18	'19	'20	
AXFR Scrape	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	10071	5087	39106	23001
CZDS Scrape	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1486740	586367
DLV Crawl	NA	NA	NA	29882	316879	437260	NA	99910	418429	409863	354055	85947	39029	11005	61829	10211
NSEC Walk	NA	NA	8818	17432	39416	125015	NA	45052	187869	189780	165194	44299	19470	5690	32058	5252
Other	2	113	10847	20440	37176	443876	NA	522203	7699153	8078773	6984714	1748770	796517	252877	1685094	297176
P-DNS	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	9669908	9950352
Registry Scrape	NA	1	853	1374	1402	969	NA	176	566	552	1150145	9710245	10571470	3794568	16348389	2477047
User Submission	12	390	39589	77625	137853	139604	NA	354217	7108250	7709995	7344963	1811286	1138116	335147	1826450	286705

3.B Raw Data of Key Transition Classifications

Throughout the paper we normalize our data to better distinguish and highlight trends. Tables 3.B.1–3.B.4 tabulate the raw numbers used to generate Figures 3.14–3.17.

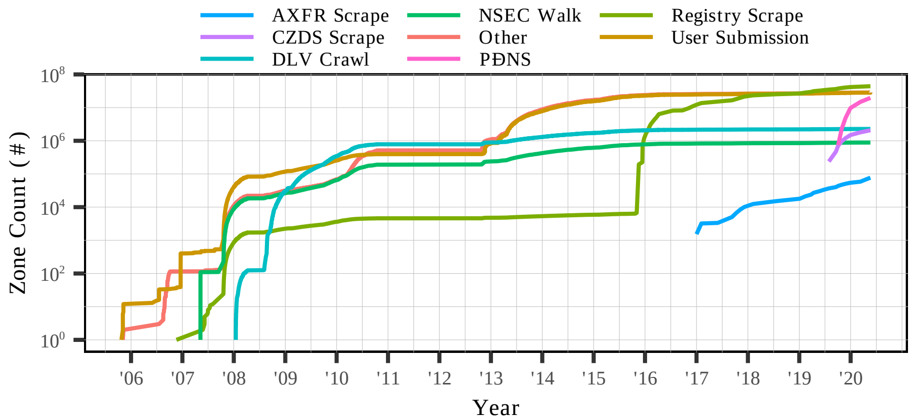


Abbildung 3.A.1: Total number of monitored DNSSEC-enabled zones per source and year

Tabelle 3.B.1: Total number of key management errors.

	Year														
	'06	'07	'08	'09	'10	'11	'12	'13	'14	'15	'16	'17	'18	'19	'20
ancient	175	4765	36176	1998	NA	41217	25409	167665	186119	158774	203820	69694	196446	59613	287679
exp	7	282	40	1399	NA	16929	12	1898	548	82	222	5355	6930	4937	20361
future	NA	52	9426	NA	NA	53	NA	495	232	103	124	52	628	143	325
inverted	NA	NA	NA	NA	NA	NA	NA	NA	28	22	10	4	NA	3	6
no-sigs	NA	NA	1703	679	NA	8018	5039	24889	35518	27482	27931	14958	25906	33372	55879
pre-incep	NA	NA	NA	NA	NA	122	NA	254	NA	NA	NA	NA	8	25	105
valid	280	2064	10098	13211	NA	170944	235059	1061506	1145816	1346260	5537326	5875182	8889718	7368282	23240768

Tabelle 3.B.2: RFC-based classification: Number of different key transitions for KSK (top) and ZSK (bottom).

	Year														
	'06	'07	'08	'09	'10	'11	'12	'13	'14	'15	'16	'17	'18	'19	'20
KSK Transitions															
RFC 5011	11	15	4	179	2202	14	759	5395	5101	7084	18061	5445	3905	16100	33845
DoubleDS	0	1	0	8	78	0	455	1862	4087	2847	24411	47035	46284	183326	243542
DoubleKSK	0	0	0	11	410	13	293	8565	20180	13424	25206	39552	42935	53756	80228
DoubleRRset	0	0	0	0	240	0	102	691	613	825	793	718	388	162	396
Noncompliant	0	21	0	286	10036	6	2554	11397	34103	36636	137853	178225	174423	357362	290231
ZSK Transitions															
DoubSig	7	15	0	250	1913	2	1274	17330	16555	20042	39605	79402	95631	232452	116298
PrePub	26	10	0	1202	8326	16	4068	67146	62260	67601	181305	451082	583795	474944	980721
Noncompliant	4	72	43	1229	17555	24	17110	241960	260294	352031	1462711	1693579	1568447	2781118	3012187

Tabelle 3.B.3: Number of valids, warnings, and errors for KSK (top) and ZSK (bottom; excluding type noncompliant).

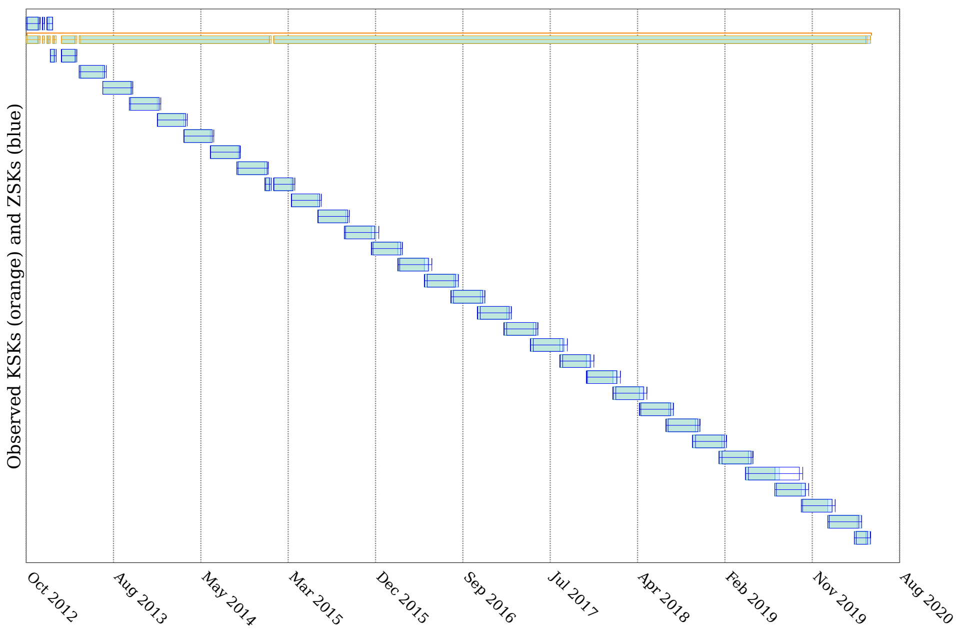
	Year														
	’06	’07	’08	’09	’10	’11	’12	’13	’14	’15	’16	’17	’18	’19	’20
KSK Transitions															
Critical	0	1	0	0	30	0	225	245	620	820	9892	3781	11700	18749	108796
Valid	0	21	0	287	10103	6	2555	11442	34160	36777	137956	178466	176729	357478	290280
Warning	11	15	4	197	2833	27	1383	16223	29304	23219	58476	88728	79506	234479	249166
ZSK Transitions															
Valid	19	1	0	517	4170	1	1255	24428	21048	26721	33833	88625	75826	79421	69998
Warning	14	24	0	935	6069	17	4087	60048	57767	60922	187077	441859	603600	627975	1027021

Tabelle 3.B.4: Behavior-based classification: Number of different key transitions for KSK (top) and ZSK (bottom).

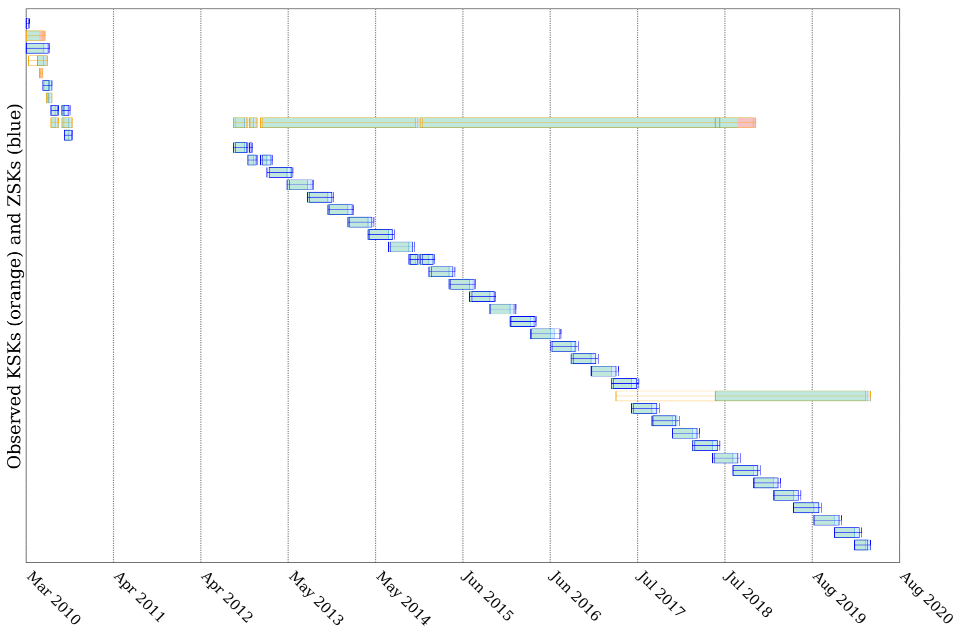
	Year														
	’06	’07	’08	’09	’10	’11	’12	’13	’14	’15	’16	’17	’18	’19	’20
KSK Transitions															
Cand-Cutover	0	5	0	7	296	0	81	486	195	683	14957	38777	64779	68512	144642
Cutover	0	1	0	5	267	2	124	8162	22182	9288	64264	55846	25438	57983	93264
Likely-Cutover	0	1	0	91	1551	10	1729	6712	27041	12475	87465	119004	148585	220946	306866
Multi-Sign	11	28	4	380	10804	21	2133	12123	14230	37662	37929	56173	28458	260701	101843
unknown	0	2	0	1	48	0	96	427	436	708	1709	1175	675	2564	1627
ZSK Transitions															
Cand-Cutover	0	10	0	12	612	3	1127	14709	5063	10416	91476	337814	492256	375772	1012625
Cutover	0	1	0	222	2131	0	1209	35893	53307	61372	268118	211801	95878	274009	149507
Likely-Cutover	10	28	38	566	6256	15	15341	193788	179113	250088	1168321	1474685	1478350	2396792	2643776
Multi-Sign	27	53	5	1872	18677	24	4679	81445	100092	116860	154093	194519	179959	431296	299926
unknown	0	5	0	9	118	0	96	601	1534	938	1613	5244	1430	10645	3372

3.C Transitions in the Wild

To illustrate the complete key life cycles of zones, Figures 3.C.1 and 3.C.2 include both ZSKs (in blue) and KSKs (in yellow). Red sections indicate periods in which keys have the revoke bit set [464]. Notable in these (and other) zones is the difference of life cycles of ZSKs and KSKs.

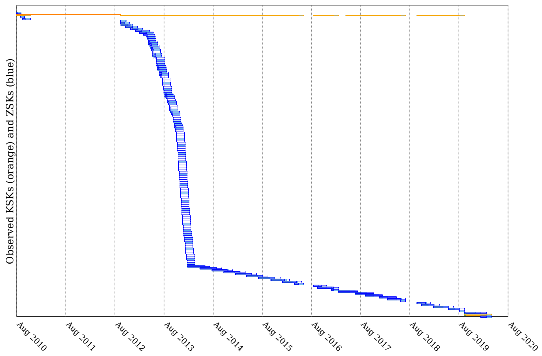


(a) The key management of the `.com` zone shows exemplary regularity and structure in its key transitions.

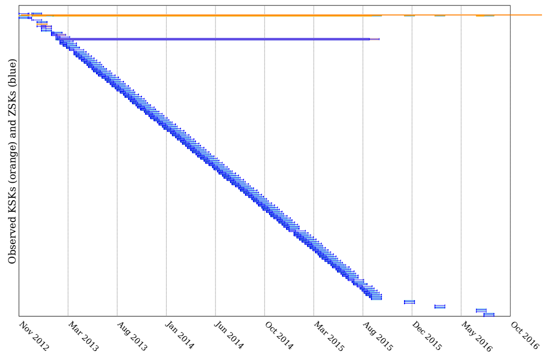


(b) Shown here is the Root zone's KSK transition away from the DURZ key (followed by revocation), and regular periodicity in its ZSK transitions.

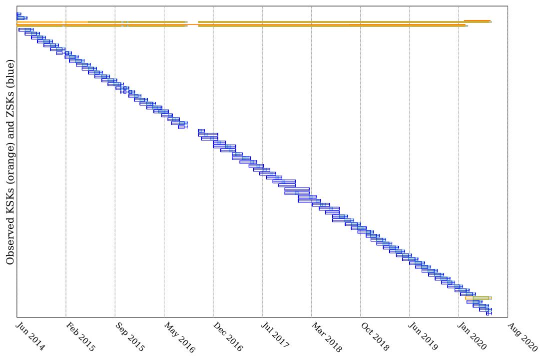
Abbildung 3.C.1: The key transition behaviors of two prominent DNSSEC zones.



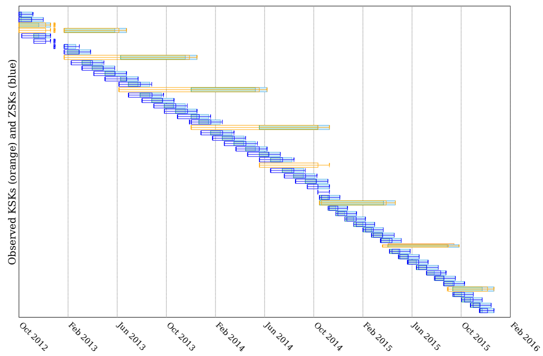
(a) `softwarehotels.se` shows distinctive key transition behavior, with up to 55 valid keys (though not all in use at the same time).



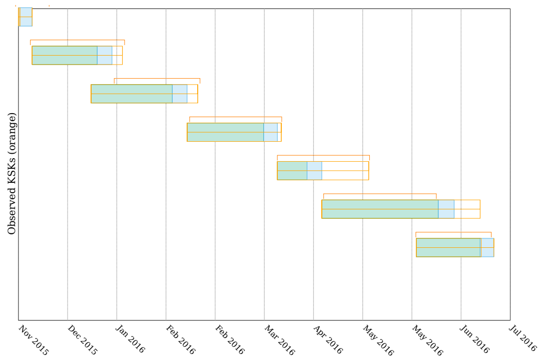
(b) The zone `cyrrax.com` shows high-frequency transitions of its ZSKs, with regular periodicity and extended *Retire* times.



(c) The key management of ARIN (a Regional Internet Registry, RIR), uses extended *PreStage* periods for its keys (ZSKs and KSKs).



(d) The key management of `advantage.gov` visibly changes from having protracted *PreStage* periods to a different model at the end of 2014.



(e) The key management of `willemvanveldhuizen.com` exhibits regular DoubleDS transitions without having any ZSKs in the zone.

Abbildung 3.C.2: Key transition patterns from other operational zones in the wild.

Kapitel 4

How to Measure TLS, X.509 Certificates, and Web PKI: A Tutorial and Brief Survey

Abstract

Transport Layer Security (TLS) is the base for many Internet applications and services to achieve end-to-end security. In this paper, we provide guidance on how to measure TLS deployments, including X.509 certificates and Web PKI. We introduce common data sources and tools, and systematically describe necessary steps to conduct sound measurements and data analysis. By surveying prior TLS meaurement studies we find that diverging results are rooted in different setups instead of different deployments. To improve the situation, we identify common pitfalls and introduce a framework to describe TLS and Web PKI measurements. Where necessary, our insights are bolstered by a data-driven approach, in which we complement arguments by additional measurements.

4.1 Introduction

The Transport Layer Security (TLS) is a widely used Internet protocol that brings end-to-end authentication, confidentiality, and integrity on top of otherwise insecure TCP and UDP transport layers. Authentication over TLS is commonly realized using X.509 certificates, and Web PKI is the ecosystem to issue, distribute, and manage X.509 certificates for the Web and other Internet services. TLS integration spans a wide range of protocols from IoT (*e.g.,* CoAP [453]), to directory services (*e.g.,* LDAP [451]), email (*e.g.,* SMTP [361]), the secure Hypertext Transfer Protocol (HTTPS) [336], and even new generation of HTTP (*i.e.,* HTTP/3 over QUIC [295]).

Due to their integral role, TLS and Web PKI have been popular measurement subjects. Measurement studies have investigated the *prevalence* of specific traits such as penetration rate of TLS 1.3 [365], the *conformance* to standards and specifications such as maintaining correct certificate structure [67], *performance* in the face of specific deployments such as newer TLS versions [141], [227], the *security* impact of concrete settings such usage of TLS sessions [228], and finally how various facets of TLS and Web PKI ecosystem can impact user *privacy* (such as to track users [233]), among others.

Measurement studies can act as experimental apparatuses to produce feedback and practical guidance to those involved in specifying , and deploying TLS and Web PKI. A common use case is to

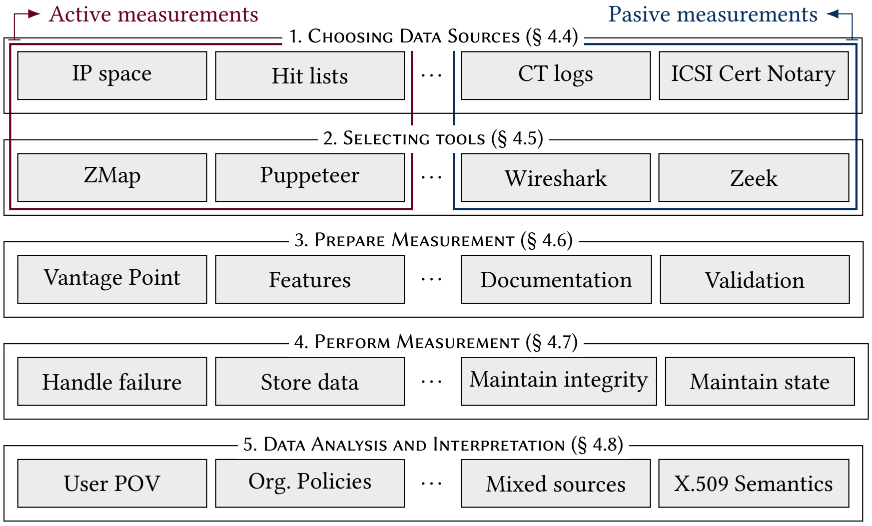


Abbildung 4.1.1: Stages of TLS and Web PKI measurement alongside selected aspects as discussed in this paper

measure if a newly introduced approach fulfills its goals. For example, if Certificate Transparency Logs [388] have been used effectively to detect certificate misissuances [102], [213]. Measurement apparatuses can also illuminate how a specific technology is perceived and utilized by its target audience, *e.g.*, how TLS servers and clients respectively support and validate stapled Online Certificate Status Protocol (OCSP) [43]. Another aspect pertains to proper implementation and deployment of TLS and Web PKI specifications. For example, if CAs properly validate CAA records to effectively reject illegitimate requests, and if not, which steps need to be taken by CAs to avoid faulty behavior [213]. Measurement results have also been effective in determining if errors go beyond implementation and are caused by errors in specifications, as it is the case for CAA specification which disregards some fundamental DNS assumptions [79]. Such specification deficiency might also have its root in the discrepancy between theoretical and practical perspectives. For example, one study [178] shows that the arbitrary amplification limit of QUIC contradicts current deployment trends and practically defeats reduced round trip times. Even sound specifications, and proper software implementations cannot eliminate all risks. Various measurement studies [33], [54], [141] on TLS confusion attacks have shown how bad configurations and practices can enable redirecting of TLS connections without endpoints noticing. And finally, users of TLS and Web PKI technologies have also been among the beneficiaries of measurement studies. For example, various researchers [139], [491] have discovered common challenges faced by users when deploying DNS-based Authentication of Named Entities (DANE), proposed solutions, and even provided tools to address these pitfalls.

Measuring and analyzing TLS, X.509 certificates, and the Web PKI is challenging due to (*i*) application-specific implementations, (*ii*) inherent flexibility of TLS, and (*iii*) different measurement setups. Applications may adapt TLS to their needs by defining further requirements and constraints. This leads to TLS implementations with different mandatory features. HTTP/2, for example, requires TLS to support Server Name Indication (SNI) and prohibits use of specific cipher suites [293, §9.], while such constraints are not given for SMTPS [361]. In terms of flexibility, TLS allows endpoints to negotiate parameters such as cryptographic algorithms during handshake. Finally, differences between measurement setups inherently lead to measurements that answer the same questions in incomparable ways, and can cause their findings to be inconsistent in some cases.

In this paper, we address a set of basic challenges and pitfalls that can occur when measuring the TLS ecosystem. To support our arguments and illustrate pitfalls, we take a data-driven approach. In addition to surveying prior work, we conduct our own measurements and provide three independent datasets. Our main contributions are:

1. A survey of prior TLS measurement research to illustrate common data collection techniques and measurement features.

2. A framework for consistently exploring TLS and Web PKI deployments.

3. An evaluation of common tools and data sources.

4. Systematic guidance on how to prepare and perform *reproducible* measurements.

5. A summary of common pitfalls when interpreting measurement data.

To the best of our knowledge, this is the first systematic tutorial on how to *reproducibly* measure TLS and Web PKI. The focus of our tutorial and survey lies in measurement methods and not resultant approaches or features that are evaluated using Internet measurements [35], [120]. We hope that these insights help to guide and improve future measurements.

The remainder of this paper is structured as follows. We start by providing a comprehensive picture of TLS and the Web PKI (Abschnitt 4.2) and then introduce a simple framework to describe and compare measurements (Abschnitt 4.3). In Abschnitt 4.4, we review common data sources and discuss their merits. Abschnitt 4.5 comprises a list of measurement tools and their capabilities. Subsequently, we discuss various aspects of preparing (Abschnitt 4.6) and performing (Abschnitt 4.7) measurements followed by pitfalls when interpreting measurement data (Abschnitt 4.8). We reflect on lessons learned in Abschnitt 4.9, and conclude this paper in Abschnitt 4.10. Information about our own measurements, *i.e.,* method, tool chain, and datasets, are provided in Abschnitt 4.10.

4.2 Background

In this section we provide a brief overview of TLS, the Web PKI, and DNS(SEC) and discuss related aspects. Abbildung 4.2.1 depicts a simple overview of entities involved in issuing and validating a certificate over TLS.

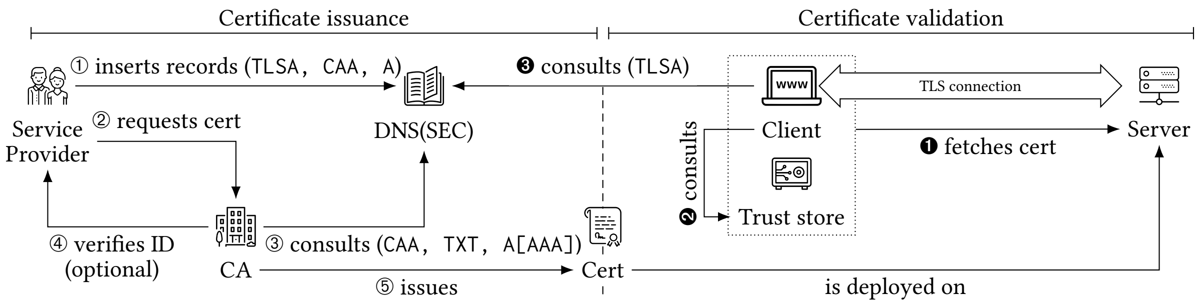


Abbildung 4.2.1: Simplified overview of certificate issuance (left) and certificate validation (right) procedures.

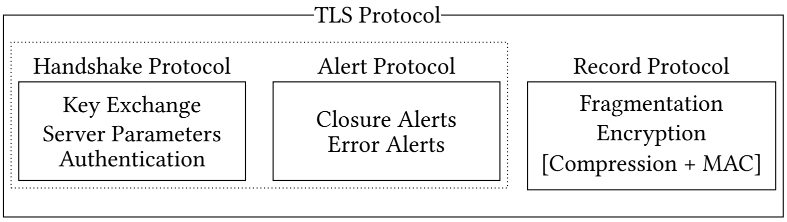


Abbildung 4.2.2: TLS Components: Handshake, Alert, and Record Protocols

4.2.1 Transport Layer Security

TLS, the successor to Secure Sockets Layer (SSL), provides authenticated integrity and confidentiality over widely used transport layers of TCP, UDP, and QUIC [439], [441], [472]. In this paper, we limit our discussion to TLS versions 1.2 [440] and 1.3 [439], as TLS versions 1.0 and 1.1 have already been deprecated [412].

Components. The TLS protocol is split into two layers: the higher layer includes TLS-specific messages (*e.g.*, alerts and handshake messages), as well as application data (*e.g.*, HTTP), and the lower layer, the *record layer*, is responsible for fragmentation, encryption, and compression of higher-level messages before handing them to the transport layer. An overview of TLS' components and their relation to application and transport layers is given in Abbildung 4.2.2.

Handshake. To establish a secure channel, peers perform a handshake to determine security parameters, establish a shared key, and authenticate each other. Server authentication in TLS 1.3 is mandatory, while authentication is optional for both peers in previous versions. Authentication is commonly realized using X.509 certificates [297] or through pre-shared keys (PSK) [290], [439].

Extensions. The functionality of TLS protocol can be extended using TLS extensions [329] Server Name Indication (SNI), and Stapled Online Certificate Status Protocol (OCSP) are among the prominent extensions.

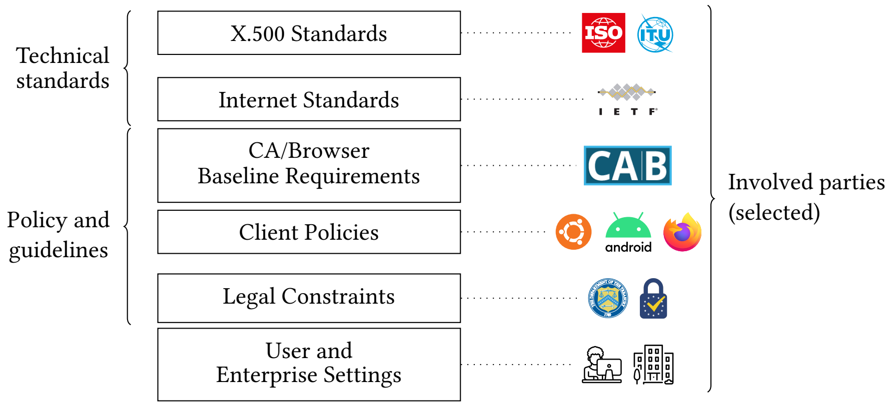


Abbildung 4.2.3: Technical and political aspects shaping X.509 Certificates in Web PKI

4.2.2 Web PKI and X.509

The most widespread approach to authentication on the Internet is based on the Web PKI: an ecosystem of certification authorities (CA), applicants and subscribers (*e.g.,* website operators), and relying parties (*e.g.,* users). A CA is a trusted third party (TTP) which validates the identity of an applicant and issues a corresponding X.509 certificate, *i.e.,* binds a public key to a subject. The applicant then becomes a subscriber of the CA. A relying party can subsequently authenticate the subscribers of its trusted CAs based on their presented certificates.

Technical basis and governing entities. The technical basis of X.509 certificates is standardized by the International Telecommunication Union (ITU) and International Organization for Standardization (ISO) as part of the X.500 standard series [373]. This is further specified and adapted by IETF for the Web PKI in RFC5280 [297]. A set of common guidelines and policies (Baseline Requirements) are also defined by the CA/Browser Forum [307]. Additionally, each CA defines its own policies and procedures in Certificate Policy (CP) and Certification Practice Statement (CPS) documents. And finally, individual organizations such as browser and operating system vendors impose their own requirements on X.509 certificates (see for example Mozilla PKI Policy [414]). Abbildung 4.2.3 depicts this hierarchy.

Usage and semantics. A certificate is used for authentication and identification through its binding of a public key to a subject. Originally, a subject was meant to identify a natural or a legal person within a globally unique directory (X.500 [372]), which never came into existence. Alternative identities, or more precisely *identifiers*, of the same subject from other namespaces (*i.e.,* domain name space) are included as Subject Alternative Names (SAN). Within the Web PKI ecosystem, the most basic form of an identifier is a domain name. Less commonly used identifiers are IP and E-Mail addresses.

Before issuing a certificate, a CA validates that an applicant has functional control over included domain name(s) (Domain Validation, DV). Additional validation procedures with higher identification assurances are Organization Validation (OV), Extended Validation (EV), and Individual Validation (IV) [306], [307]. For DV validation, an applicant can prove its control over respective domain name(s) through automated methods (*e.g.,* using ACME protocol [292]), or manually through e-mail or phone calls. OV and IV certificates include additional identification information regarding respective organizations or individuals as defined in CA/B Baseline Requirements [307]. EV certificates, originally introduced to make e-commerce more trustworthy, are subject to stricter requirements, audits, and life cycle management [306]. Further validations types also exist for other purposes (*e.g.,* S/MIME authentication) [308].

4.2.3 DNS and DNSSEC

DNS is a hierarchical and distributed database with an eventual consistency model that maps domain names to Resource Records (RR) of various types, for example IPv4 and IPv6 addresses (RR type `A` and `AAAA`). DNSSEC [362] is a set of security extensions that address a number of DNS security shortcomings (see RFC4033 [444, §3]).

In the context of TLS and the Web PKI, DNS(SEC) is utilized to counter various challenges. As such, one common view is the holistic measurement of TLS and Web PKI deployment must also include its relation to DNS. In the following, we discuss three use cases of DNS(SEC) in the context of TLS and the Web PKI.

Binding domain names to certificates. CAs are generally not restricted in their certificate issuance [297, §4.2.1.10.], *i.e.,* a CA can issue a certificate for any arbitrary subject name (*e.g.,* domain name). As such, a relying party cannot definitely determine if a given certificate was authorized by the entity denoted in the subject field or if it has been misissued.

The *DNS-Based Authentication of Named Entities* (DANE) for TLS [363] addresses this ambiguity between public keys and CAs by allowing the owner of an identity (*e.g.,* a domain name or an e-mail address) to specify or characterize authorized certificates. DANE relies on DNSSEC [362] and `TLSA` RRs. An experimental TLS extension also enables embedding complete DNSSEC chains and `TLSA` records into the TLS handshake [327].

DANE can be an effective remedy against rogue or compromised CAs as well as CAs used by intelligence services and governments [226] such as the root certificates generated and forced by the Kazakhstan Government [378] or the TrusCore CA, which recently was disclosed to have ties to 'contractors for U.S. intelligence agencies and law enforcement' [405].

Domain names that authorize certificate issuance. Similar to relying party, CAs also need additional information to know if an applicant for an identity is actually authorized by identity owner (*e.g.,* a domain name owner) to avoid misissuance. Domain name owners can use the CA authorization (`CAA`) resource record [357] to indicate which CAs are allowed to issue certificates for their domain names and also to report policy violation.

Domain names signaling opportunistic TLS. While browsers commonly lack DANE support for HTTPS [491], DANE has found popularity among SMTP servers [140] to resist downgrade vulnerabilities using `STARTTLS` [8]. Mail server operators that are reluctant or unable to deploy DNSSEC, *e.g.,* Google, alternatively use *SMTP MTA Strict Transport Security* (MTA-STS) [399]. In this approach a

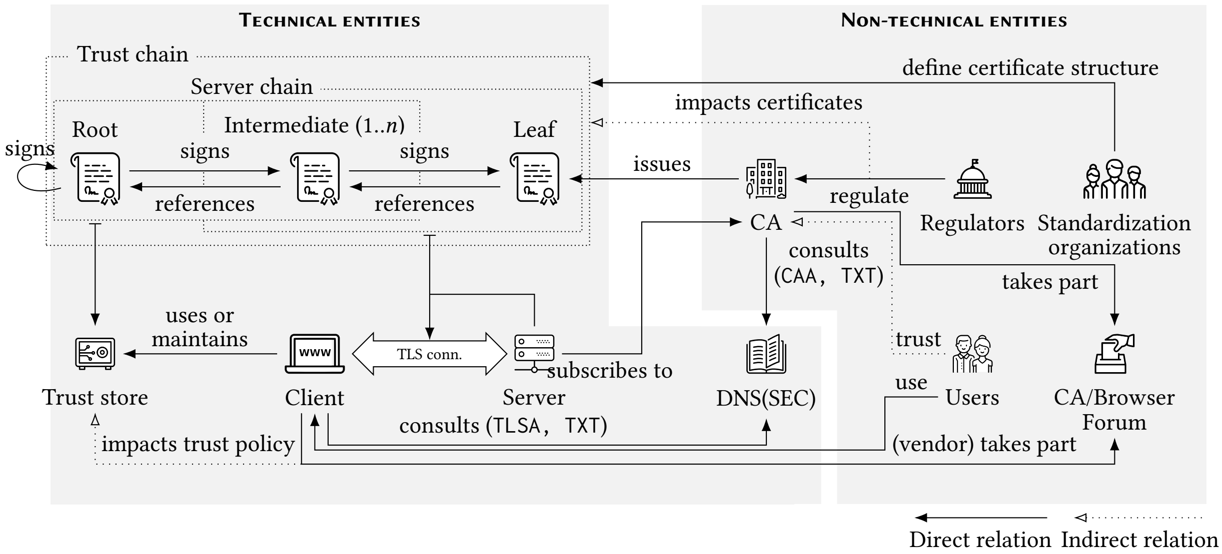


Abbildung 4.2.4: An overview of involved parties and components in TLS and Web PKI ecosystems

TXT resource record under a well-known subdomain is used to signal TLS support by respective mail servers. Related policies can then be fetched over HTTPS from a well-known URI (see RFC8615 [427]).

Abbildung 4.2.4 depicts all technical and non-technical entities discussed in this section.

4.3 Principles for a TLS Measurement Framework

Due to the behavioral variability of TLS and Web PKI, there is no one-size-fits-all measurement approach. At the same time, all measurement studies share commonalities to varying degrees. To describe measurement studies and make them comparable, we call for a systematic classification as the basis for our measurement framework. For this, we survey various measurement studies published between 2011 and 2014.

Our proposed framework classifies measurement studies on four dimensions: (*i*) measurement objective, (*ii*) measured features, (*iii*) measurement frequency, and (*iv*) data collection method as depicted in Abbildung 4.3.1. We avoid fine-grained classes and categories in favor of generality. Abschnitt 4.C summarizes the classification of surveyed work. In the following we take a closer look at each aspect.

4.3.1 Measurement Objective

Each measurement study follows a specific goal and aims to answer a concrete research question. We introduce five coarse measurement goals and classify studies in our survey accordingly. We hope to (*i*) gain a better understanding of well-studied as well as less considered aspects, and (*ii*) provide an overview of topic for researchers getting familiar with TLS and Web PKI measurements. We propose following groups:

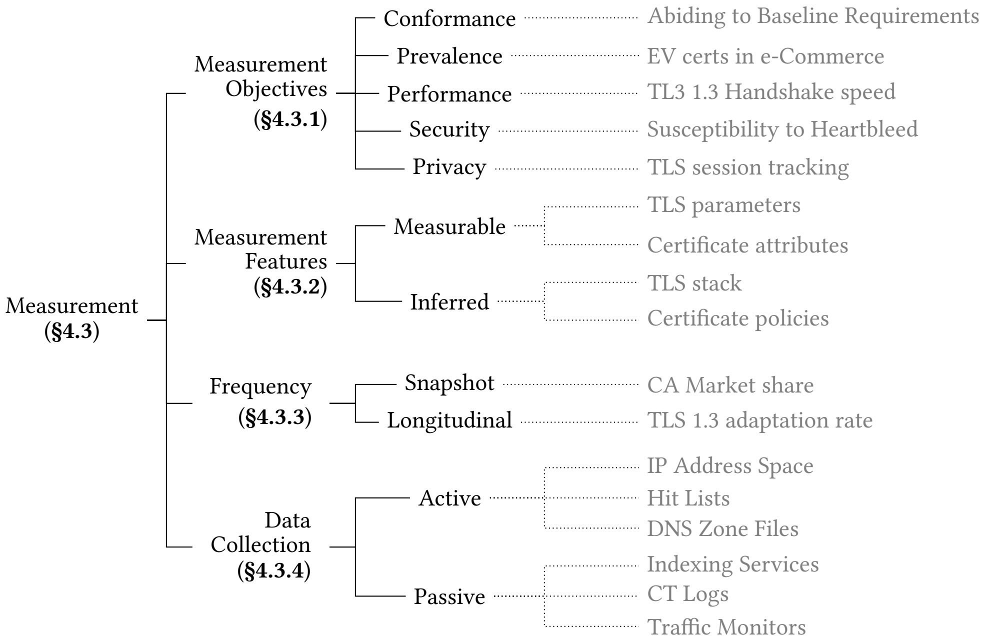


Abbildung 4.3.1: A taxonomy for TLS and Web PKI measurement with selected examples (in gray)

Prevalence measurements regard occurrence of a set of features or traits such as TLS version or supported cipher suites, EV certificates, or misconfiguration of CAA or DANE.

Conformance measurements consider if TLS implementation, X.509 certs, or any relevant procedure conform to corresponding specifications, standards, or regulations such as RFCs (*e.g.,* supporting mandatory cipher suites), CA/B Baseline Requirements (*e.g.,* maintaining revocation deadlines), or browser requirements (*e.g.,* provision of Signed Certificate Timestamps).

Performance measurements quantify or qualify the impact of a feature, variation, or version of TLS protocol or X.509 on, among others, number of roundtrips (*e.g.,* when reusing TLS), fragmentation of packets in underlying layers (*e.g.,* by large CRL packets), execution speed (*e.g.,* of TLS handshake), or certificate size (*e.g.,* in face of sharing).

Security measurements investigate how specific deployment or presence of a set of features impact security aspects such as confidentiality (*e.g.,* when using weak TLS symmetric keys), integrity (*e.g.,* by using weak hashing algorithms), or availability (*e.g.,* impact of large certs on QUIC amplification).

Privacy measurements study how specific features can impact privacy, for example, through information leakage over unencrypted TLS handshake, tracking users using TLS session resumption, or mapping devices to IP addresses using invalid certs.

Measurement studies have their focus on one or more objectives. Our summary in Abschnitt 4.C reveals security and prevalence topics are the most popular among measurement studies while privacy is less considered.

4.3.2 Measurement Features

We differentiate between two types of measurement features *measurable* and *inferred*. Measurable features can directly be measured and are quantifiable: X.509 public Key length, TLS session validity period, *etc.* are examples of this type. Inferred ones can be inferred from primary features or raw data and might not be quantifiable: certificate quality, TLS stack in use by a server, CA market share, and certification policies (CP and CPS document), to name a few. Inferred features might be useful in interpreting data. In our own measurements, for example, we observe all Iranian banks subscribing to an obscure Polish CA in contrast to their European counterparts that commonly use services from popular CAs (*e.g.,* DigiCert). To explain this irregularity, we took look at CA CPS documents (linked in respective certificates) to find out that none of the heavy-weight CAs provide services to Iranian companies due to US/EU sanctions. Abschnitt 4.8 provides a detailed discussion.

Here, we rely on specifications (*e.g.,* RFCs) and insights from our survey to define a set of primary features for TLS and Web PKI that are relevant for measurement studies. Our focus lies on study of TLS usage by Web servers, and X.509 certificates used for client or server authentication. Note that some TLS and Web PKI aspects are associated with DNS(SEC) records and require collecting respective records. Due to decentralized, hierarchical nature of DNS, its measurement usually involves querying multiple name servers. For example, CAA records for a given domain name can be positioned at any hierarchy level up to the root zone, so evaluating CAA is bound to a recursive DNS query. Similarly, to construct DNSSEC trust chain, respective records are to be fetched starting from the root zone (or any other trust anchor) up to the domain name in question. When using recursive resolvers to perform DNS queries, cached and potentially stale data, can influence measurement results. For an exhaustive discussion DNS and DNSSEC measurement methods, we refer to related work [186], [188].

Transport Layer Security. TLS can be measured regarding each of its components (see Abbildung 4.2.2) and the integration of the protocol as a whole in context of an application:

T1 Handshake layer features (key exchange, server parameters, and authentication method)

T2 Record layer features (fragmentation, encryption, compression, and hashing algorithms)

T3 Integration with the higher level application layer

Web PKI. Web PKI, as a complex ecosystem, has been studied regarding various aspects including its global structure [67], [111], [128], [189], [210], established trust stores [71], [127], [152], [407], [429], [195], and CA certification policy and procedures [86], [102], [214], [217]. X.509 certificates, at the core of Web PKI, can further be analyzed regarding the following properties:

X1 Attributes (*e.g.,* subject, issuer, and validity fields)

X2 Extensions (*e.g.,* expected usage, SAN, validation method, and key usage flags)

X3 Revocation status (OCSP and CRL)

X4 Relation to relevant DNS records (*e.g.,* CAA and TLSA)

Note that not all certificate types, *i.e.,* root, intermediate, and leaf, carry the above properties. Root certificates, for example, are commonly excluded from revocation checks over OCSP and CRL endpoints. Additionally, the validation method, *i.e.,* domain, individual, organization, and extended validation methods, have an impact on certificate properties. Domain validation (DV) certificates, for example usually carry a single name as common name in the subject field, whereas extended validation (EV) certs include real-world identity information about the owner. How TLS and X.509 features have been contextualized in related work to answer specific research questions is summarized in Abschnitt 4.C.

4.3.3 Frequency

Throughout our survey, we found that frequency, *i.e.,* how often a measurement is performed within a given time period, is an important feature. Two groups are considered: snapshot and longitudinal. Studies that are interested in observing temporal developments and discovering causal relations, *e.g.,* in response to introduction of a new technology such as QUIC, perform the measurement multiple times over a period of time. Such longitudinal measurements can be considered as regular repetition of snapshot measurements.

Snapshot. This type of measurement is based on probing a sample set on a set of features at a given point in time. Such measurements help to understand state of affairs at a single point in time. Shobiri *et al.* [224], for example, investigates if CDN provides observe security hygiene when communicating to backend servers by monitoring incoming connections from CDN edge servers.

Longitudinal. Repeating a snapshot measurement over a period of time is referred to as longitudinal measurement. In contrast to snapshots, longitudinal data are suitable for understanding causal relations. Probing same targets over time enables capturing temporal changes, *e.g.,* in response to an incident. Durumeric *et al.* [68], for example, analyze the patch rate of HTTPS servers vulnerable to Heartbleed by probing Alexa Top 1M sites and a sample subset of IPv4 address space every 8 hours for a period of ca 3 months. Similarly, Zhang *et al.* [265] study revocation and reissuance of certificates for vulnerable hosts at the wake of the Heartbleed.

The frequency of longitudinal measurements is in practice be proportional to the frequency of changes that are to be captured. Chung *et al.* [45], for example, align their measurement frequency with DNS TTL values in a longitudinal DNSSEC study. Osterweil *et al.* [186], refers to the *Nyquist–Shannon sampling theorem* stating that measurement frequency must be at most half of the target frequency if all changes are to be captured. Even so, outages, failures, and other disturbances can leave temporal gaps in collected data which has to be accounted for to establish data coherency. For example, Osterweil *et al.* [186] introduces a method that can detect, fill in, or discard gaps in DNSSEC data collected over 15 years.

4.3.4 Data Collection

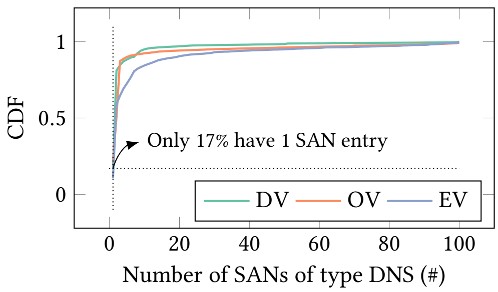
Final aspect of our classification regards how measurement studies collect their data. Two approaches are common, active and passive measurements [117]. Active measurements entail crafting packages and injecting it into the network. In contrast, passive measurements do not intervene with network traffic and collect data from existing traffic. In the following we briefly introduce each method and in the next section we introduce data sources for active and passive measurements that are used in practice, and discuss their merits.

Passive. Measurement studies that collect data passively rely on existing data flow, *i.e.,* as it transits over the network. Among surveyed measurements, the most common approach is to intercept data, *e.g.,* as it goes through routers or caught by honey pots. Many research institutions have their own infrastructure to intercept data from own networks (see Abschnitt 4.C). Intercepting TLS packages, however, poses a non-trivial obstacle as data might be encrypted and cannot be dissected, *e.g.,* during the handshake or upon TLS session resumptions. To address this, alternatives have been proposed to collect the data directly at the endpoint. Oakes *et al.* [184], for example, collect data from a web monitoring tool installed on user devices. This can also be the on the server side, *e.g.,* a web server that logs all necessary information about incoming packages.

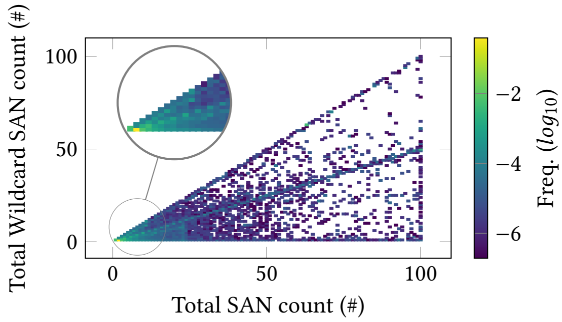
Passive measurements are useful for studying traits that regard users-related aspects, e.g., actual cipher suites selected by users. The main handicap is the relatively complex requirements, *e.g.,* having access to special network nodes or user browsers, that limits the scope of passive measurements. For example, passively measuring censorship in countries with restricted Internet access (*e.g.,* Iran or China) from the outside and without having access to users endpoints or the network infrastructure is infeasible. In face of such challenges, active measurements might be more effective as Raman *et al.* [197], [198], for example, show.

Active. In active measurements purposefully generated or manipulated data packets are exchanged with hosts. Both request and response data are then recorded for further analysis. For this, a set of target hosts, denoted by an IP address or respective domain name, is required. Abschnitt 4.4 gives an overview of most common approaches to compose or obtain such a target list. Beside own infrastructure, active measurement studies in our survey also make use of crowdsourcing platform such as RIPE Atlas [204] and OONI [76].

Active measurement is suitable to investigate the behavior of a counterpart in presence of a concrete condition or data packet. Hebrok *et al.* [107], for example, analyze security of TLS sessions by generating special handshake messages and testing these against TLS servers. On the downside, active measurements put a strain on the network in contrast to passive ones. Specially for large scale studies the amount of data that is generated can have a negative impact on the network and even be interpreted as predatory. Furthermore, active TLS measurements are especially complex due to the negotiations performed during the handshake. In Abschnitt 4.6, we discuss best practices to address and avoid this issue.



(a) Distribution of number of DNS SANs (capped at 100) by validation type (Domain, Organization, or Extended Validation)



(b) Frequency of valid certificates based on the number of wildcard and total SAN of type DNS in each certificate (capped at 100)

Abbildung 4.4.1: Subject Alternative Names (SAN) statistics of X.509 certificates in our TRANCO dataset

4.4 Data sources

Another dimension that TLS measurement studies exist across is the choice of data source. For active measurements, this is a set of hosts that support TLS. Such set is commonly composed through IP address space scanning (§4.4.1), sourced from Hit Lists (§4.4.2), extracted from logged certificate (§4.4.3), or obtained directly from TLD registries (§4.4.4). For passive measurements, various methods are used by the surveyed studies. Most common are crowdsourcing (§4.4.5), using data collected and provided by indexing services (§4.4.6), or monitoring network traffic (§4.4.7).

In the following, we introduce various sources for data collection and discuss their merits, and pitfalls. In Abschnitt 4.6, we dig deeper into the ramifications of choosing specific data sources for measurements, and in Abschnitt 4.8 we discuss pitfalls when interpreting data from specific sources.

4.4.1 IP Address Space Scanning

A straightforward and commonly used approach to TLS and Web PKI measurement is first to scan the IP address space (or a selected subnet) for ports reserved for TLS-enabled protocols (*e.g.*, 443 for HTTPS) and then to establish a TLS connection with matching hosts. It should, however, be noted that since presence of an open port on its own is not a guarantee that respective protocol is also available at that endpoint [69], [110], false positives need then be discarded during the actual measurement. In Abschnitt 4.5, we discuss tools that address both IP scanning and service detection.

A disadvantage of IP scanning is the lack of a relative priority or importance among IP addresses, *e.g.*, in terms of popularity or usage, so that at first glance the IP address of a home router (possibly accessed by a single individual) has the same weight as an IP address of a critical or high-traffic server, *e.g.*, `www.google.com`. Additionally, detecting cases of opportunistic TLS, *e.g.*, `STARTTLS` over SMTP, require scanners to perform protocol handshakes to determine if a security upgrade is supported.

Most critically, research [69], [111], [247] shows how IP scanning can skew measurement results and result in an incomplete picture of TLS and Web PKI landscape, partly due to virtual hosting and required usage of SNI. Although SNI is an optional TLS extension [329, §3], its use is mandated by some higher-level applications protocols. HTTP/2 [293, §9.2.] and HTTP/3 [295, §9.3.], for example, require clients to use SNI for the TLS connection if server is identified by a domain name. Respectively, web servers behavior vary depending on existence of SNI. Cloudflare, for example, aborts the TLS handshake (alert 40) if no SNI is provided, whereas Firebase (Google) and Wordpress return wildcard certificates (`*.firebaseapp.com` and `*.wordpress.com`) that would match the free subdomains allocated for their customers. The possibility of encountering such cases can be elaborated by Abbildung 4.4.1a which depicts the distribution of the number of DNS SAN in certificates from our `TRANCO` dataset (see Abschnitt 4.10). We observe that only about 17% of all certificates enlist a single SAN entry and among the rest about 16% contain at least two unique eSLDs signalling that the certificate might be shared among two different subjects but served under the same IP.

4.4.2 Hit Lists

We refer to a set of curated or purposefully collected domain names that fulfills specific criteria as a *hit list*. Related work has already evaluated different aspects, such as reliability, reproducibility, and fluctuation of hit lists alongside their usage implications in research [137], [211], [214].

Based on our survey, the most prominent hit lists are Alexa and Tranco. The latter practically replaced the former since it became defunct. In the following, we also discuss other actively maintained and publicly available lists. Tabelle 4.4.1 provides an overview.

Tranco List. An aggregation of various popularity-based top lists with the aim of being consistent, manipulation-resistant, reliable, and reproducible [137]. By default, domain names in the Tranco List are trimmed at the effective second level domain name (eSLD) and contain only the public suffix (effective TLD; eTLD [402]) plus the next label (referred to as *pay-level domains* by list maintainers). For example `city.ofunato.iwate.jp` (an actual entry) consists of the public suffix `ofunato.iwate.jp` and `city` as its effective SLD. This default structure can be modified through Tranco website[1] where registered users can apply additional filters such as setting the combination algorithm or removing domains marked as unsafe by Google Safe Browsing database.

It should be noted that not all entries in this list resolve to IP addresses: at the time of this writing, for example, 116 listed domain names out of the top 500 and 92,166 (9.2%) of the top 1M do not resolve to an IP address (*i.e.,* have no DNS `A` records). The main reason for this is the trimming behavior mentioned above that causes the list to include domain names belonging to CDNs, cloud service providers, or infrastructure operators, *e.g.,* `akamaiedge.net`, which do not resolve to IP addresses at that level. Creating a custom list with subdomains beyond the eSLD partly solves this issue but results in inclusion of public suffixes such as `gob.es` or `ca.us`. Beside a few exceptions (*e.g.,* `gov.uk`), public suffixes do

[1] `https://tranco-list.eu`

Tabelle 4.4.1: Comparison of various hit lists

Name	Source	Scoped	Ordered	Entries
Tranco	List Aggregation	✘	✓	$\approx 4.2M$
CISCO Umbrella	Passive DNS measurements	✘	✓	$1M$
Cloudflare Radar	Passive DNS measurements	✘	✓	$\geq 1M$
Majestic Million	Active Web crawls	Web	✓	$1M$
Chrome UX Report	Browser telemetry	Web	✓	$1M$
HSTS Preload	Zone owner submissions	Web	✘	$126k$

not resolve to IP addresses and some browsers (*e.g.*, Chrome) even consider them as a search query when entered in the address bar.

Tranco list has become popular among researchers due to its transparent ranking algorithm and reproducibility. The combination of various hit lists in one list simplifies the decision for a hit list. And, customization features, such as excluding a specific list, allows tailoring the final list for measurement objective. Finally, the relatively low fluctuation in the Tranco list make it suitable for specific measurements (see for example Tatang *et al.* [469]).

CISCO Umbrella. This is a popularity ranked list based on passive measurements of DNS queries over CISCO Umbrella network[2] (formerly OpenDNS services). Similar to Tranco, reproducibility is provided (by date), but list entries are not trimmed to their functional SLD. As a DNS-based hit list, the position of a domain name in this list is proportional to the number of its DNS queries as observed by CISCO recursive resolvers.

This list suffers from three handicaps. First, about every fifth entry in the top 500 list (total of 118) does not resolve to an IP address. Second, many entries share the same eSLD or belong to the same owners. For example, more than half of the top 25 entries (14 in total) are related to `netflix.com` or `netflix.org`. Finally, the ranking and normalization algorithms are neither public nor verifiable.

Although none of the surveyed work make use of this list, it can be advantageous for measurement studies that are specifically interested in popular hosts in terms of DNS query volume, *e.g.*, mobile push notification endpoints.

Cloudflare Radar. Similar to Cisco Umbrella, Radar statistics by Cloudflare are based on observations at own DNS resolvers (`1.1.1.1`). The data can be fetched over Cloudflare website or through its API which provides more detailed information. Beside a global ranking (over $1M$ domains), top 100 domains per country are also available. Domain names in this set are trimmed at effective second level domain. Much like Tranco, trimming leaves a non-negligble portion of entries without a respective IP address.

[2] `https://umbrella.cisco.com`

Similar to CISCO Umbrella, this list is useful when DNS query volume is relevant. In addition, as Cloudflare DNS enjoys a relatively higher market share in comparison to other open resolvers [196], this hit list better reflects actual relations on the Internet.

Majestic. The Majestic Million [398] top list is a 'link-level backlink index' (popularity ranked) based on data gathered from web crawls. The domain rank here (trimmed at eSLD) relates to (*i*) the number of its 'External Inbound Links', and (*ii*) counts of referring domain, IP, and subnets which are calculated during the crawl [337].

Among the top 500 domains in this list only 16 entries do not resolve to an IP address. Invalid entries are either public suffixes (*e.g.,* `go.id`), private suffixes (*e.g.,* `azurewebsites.net`), or simply miss a label (*e.g.,* `miit.gov.cn` without `www` label).

Majestic provides a suitable basis for measurements that focus on the Web. Specifically if popularity in terms of reference is relevant.

Chrome UX Report. As part of its user experience report, Google provides a popularity ranking of domain names based on data collected from Chrome browser users. The ranking is further divided by country, platform, and popularity metric [210]. In contrast to Tranco, host names are not trimmed at eSLD and Ranking is only provided 'on a log10 scale with half steps'[3].

The main advantage of this list is having browser data as its main source which leads to a better accuracy compared to other top lists [211]. For example, only 10 out of the top 1000 names in this list do not resolve to an IP address at the time of this writing.

HSTS Preload List. Curated by Google as part of the Chromium Project, the 'HTTP Strict Transport Security (HSTS [360]) Preload List' is a collection of domain names that supporting browsers contact only through HTTPS[4]. At the time of writing more than 120k names are listed including public and private suffixes (*e.g.,* `zip`, and `now.sh`), eSLD domains, and individual domain names including all subdomains (*e.g.,* `www.aclu.org`). In contrast to the aforementioned hit lists, the entries here are not ordered and do not fulfill any specific criteria except supporting TLS. Moreover, not every domain name in this list is delegated (`NXDOMAIN`) or carry an `A` record.

4.4.3 Certificate Transparency Logs

To address the problem of 'misissued certificates', Certificate Transparency (CT) Logs [388], [213] were introduced. These are publicly available and auditable data structures that are modified in an append-only manner. Major browsers such as Chrome and Safari (with a combined global market share of over 90% [462]) validate server certificates only if they are logged in multiple CT logs [284], [353], [401]. For this reason many CAs (*e.g.,* Let's Encrypt) log the certificates they issue in CT logs by default.

CT logs are a direct source for X.509 certificate. However, it should be noted that not every logged certificate is also deployed on a TLS server. Cloudflare, for example, vicariously applies for at least two certificates for its customers: a production and a backup certificate to instantly replace it in emergency

[3]See `https://developer.chrome.com/docs/crux/methodology/metrics`

[4]`https://www.chromium.org/hsts/`

cases (*e.g.,* key compromises) [384]. Furthermore, it is not possible to infer which service actively uses a logged certificate (*e.g.,* HTTPS or SMTP). So in cases where actual deployment of certs is relevant for the measurement study, further steps need to be taken to address aforementioned issues.

Logged certificates can also be used to harvest domain names which are in turn used as a point of departure for active TLS measurements. Note that for a given logged certificate (even if not expired or revoked) it is not guaranteed that the domain names that it covers is still delegated or resolves to an IP address. Furthermore, due to wildcard certificates, it is not possible to discover every viable domain name from CT logs, and in turn every TLS-secured host relevant for the measurement study. For example, if a wildcard certificate is shared among different domain names that resolve to different TLS servers, this method will miss some or all of those servers. In our `TRANCO` dataset (see Abschnitt 4.10), for example, about 40% of all valid certificates have at least one wildcard SAN while less than 1% solely include wildcard names. Abbildung 4.4.1b depicts the frequency of certificates in our `TRANCO` dataset based on number of wildcard SAN entries and the total number of SANs in each certificate.

A unique advantage of CT logs lies in their public APIs that allow to monitor CA certifications. Since certificates are usually published in logs within 24 hours of issuance (*i.e.,* maximum merge delay), CT monitors can closely follow certificate issuance. For example, to check if a CA correctly validates CAA records, DNS records at the time of issuance are required. Scanning CT logs allows querying DNS as soon as a certificate is published and increase the chance of having a consistent view of CAA records as observed by the CA. Any later time would increase the probability of DNS records being changed and CAA validation unreliable [86], [214].

4.4.4 DNS Zone Files

The Internet Corporation for Assigned Names and Numbers (ICANN) allows interested parties to directly request generic TLD (gTLD) registries for their zone files through the Centralized Zone Data Service (CZDS). Zone files for TLDs are commonly used in DNS and DNSSEC measurements [139], [186], [491] but are also suitable in context of TLS and Web PKI (see, for example, Holz *et al.* [365]).

The downside of using DNS zone files is twofold: (*i*) not every TLD is part of CZDS and not every request is granted, and (*ii*) the massive amount of names —`.com` alone is estimated to have 160M delegations [324]—in no specific order poses a challenge for both the measurement procedure and interpretation, *e.g.,* to weigh domain names by popularity. Merits of using DNS zone file as point of departure have been discussed in related work [491].

4.4.5 Crowdsourcing Services

Open measurement platforms such as RIPE Atlas [204] and OONI [76], rely on volunteers to perform actual measurement, while data is collected and stored centrally. On the one hand, crowdsourcing caters for better scalability and physical distribution of measurements. And on the other hand, the lack of control over measurement probes, and unawareness of actual network condition or configuration where measurements were performed poses a challenge for data interpretation.

4.4.6 Indexing Services

Numerous commercial companies in the field of Internet measurement and scanning provide access to their data. Two prominent examples are Censys Search[5] and Rapid7 Open Data[6]: Censys performs daily scans of the IPv4 address space and provides an API to query and search collected data. Data by Rapid7 is limited to its Open Data repository containing artifacts of regular scans by Project Sonar [437] and lacks search and query features.

Downside of using such services as data sources is lack of control and transparency on underlying scanning procedures. Wan *et al.* [252], for example, discuss how regular scans by Cencys (using ZMap; see **??**) lead to its source IP addresses being blocked and consequently missing up to 4.6% of all HTTPs hosts. A similar phenomenon is also observed by Chung *et al.* [42] for datasets from Rapid7 and University of Michigan, suggesting that data from regular scanners are susceptible to being incomplete due to blacklisting.

4.4.7 Internet Traffic Monitors

Instead of collecting data, we could rely on data collected by traffic monitors. Passive measurements present an authentic snapshot of Internet usage but are usually limited in their scope, *e.g.,* traffic on a university network, and require access to special network nodes.

Note that although using passive measurements for TLS and Web PKI analysis is widespread [10], [102], [110], [111], [128], [189], such data might not be suitable for specific use cases [269], [491].

4.5 Tooling

In this section we provide an overview of some widely established tools for measuring TLS and Web PKI and discuss their merits regarding aspects introduced in Abschnitt 4.3 and their actual deployment in surveyed measurement studies. The majority of tools below are meant for active measurements. The only exception, Zeek, can be used for passive measurements through traffic monitoring. Tabelle 4.5.1 provides an overview.

OpenSSL. An open source implementation of a wide variety of cryptographic algorithms and protocols, OpenSSL is used in client (*e.g.,* curl) and server software (*e.g.,* NGINX) alike and lends itself as an appropriate tool to collect and analyze TLS and Web PKI data.

It can be used programmatically through its C API or any of its CLI tools, such as `s_client` (a TLS client), or `x509` (an X.509 certificate parser). However, manipulating low level details, *e.g.,* of handshake (**T1**) and record layers (**T2**) or X.509 extensions (**X2**), is limited to its API. OpenSSL is capable of validating signed certificate timestamps (SCT) and DANE records as well as revocation status over CRL, OCSP, and stapled OCSP (**X3**) if required resources, *e.g.,* DNS resource records, are provided accordingly.

[5] `https://censys.io/data-and-search/`
[6] `https://opendata.rapid7.com`

Tabelle 4.5.1: Overview of selected TLS measurement tools and their capabilities

			Command Line Interface		TLS†			X.509‡			
For type	Tool	API	Batch Process	Machine Readable	**T1**	**T2**	**T3**	**X1**	**X2**	**X3**	**X4**
Active	OpenSSL	`C`	✘	✘	●	●	○	◐	●	●	◐
	curl	`C`	✘	(JSON)	◐	◐	●	◐	◐	◐	○
	ZMap Suite	`GO`	✓	JSON	●	◐	●	●	●	◐	○
	TLS-Attacker Suite	`Java`	✓	DB	●	●	○	○	●	○	○
	testssl.sh	✘	✓	CSV / JSON	◐	◐	◐	◐	○	●	◐
	goscanner	✘	✓	CSV	●	○	●	●	○	○	○
	Puppeteer	`NodeJS`	— N/A —		○	○	●	●	◐	○	○
Passive	Zeek	`Zeek Script`	✓	TSV / JSON	◐	◐	○	◐	◐	◐	○

† **T1**: parse or manipulate handshake layer / **T2**: parse or manipulate record layer / **T3**: support application layer protocols

‡ **X1**: parse X.509 attributes and validate certificate / **X2**: parse and interpret X.509 extensions / **X3**: evaluate revocation status (OCSP and CRL) / **X4**: relate the certificate to DNS records

● Full support / ◐ Partial support / ○ No support

OpenSSL suffers from four shortcomings: (*i*) low level API (*e.g.,* manual memory management), (*ii*) non-machine-readable CLI output, (*iii*) partial nonconformity with RFCs, and (*iv*) missing integration with higher level application protocols (**T3**). Furthermore, standard distributions of OpenSSL do not contain weak cipher suites which are relevant for security measurements (**T1** and **T2**).

curl. A command-line tool and library (`libcurl`) that implement various Internet protocols. curl is extensive in terms of configuration and protocol support (**T3**) but has its focus mainly on transferring data. As such, the CLI only provides a rudimentary option to print data in machine-readable format: either by selecting from a set of predefined keys or by printing all the keys in JSON format.

The drawbacks are as follows: (*i*) specific data, *e.g.,* raw certificates, as well as granular access to handshake and record layers (**T1** and **T2**) are not given over the CLI but only through `libcurl`. This API, however, is not as extensive as the underlying TLS engine, *i.e.,* provides limited callbacks to OpenSSL functions or available cipher suites. And, (*ii*) curl is limited in parsing X.509 certificates and provides unstructured data over the CLI and uses its own structure over the API (**X2**), (*iii*) it does not support validation of SCT and OCSP (**X3**), and finally (*iv*) it cannot be configured to validate CAA or DANE (**X4**). By default, curl distributions are based on OpenSSL and inherit all respective shortcomings as discussed above. It is, however, possible to manually build curl with other TLS engines.

ZMap Project. ZMap is a network scanner similar to Nmap but with a focus on modularity and speed [69]. It can be used both programmatically and over the CLI. Since its introduction, ZMap has been grown into a software suite including an application-layer scanner (ZGrab), a DNS lookup tool (ZDNS), and an X.509 certificate parser (ZCertificate) and linter (ZLint). ZMap is used in production settings, *e.g.,* by CAs and monitoring services such as Censys, and is prominent among measurement studies and scanners [11], [66].

ZMap CLI tools produce machine-readable output in JSON format and can be combined using standard Linux pipes. For example combining `zmap`, `zls` [114], and `zgrab2` allows scanning hosts for various ports, detect services, and finally perform an application layer handshake. It accepts custom client hello handshakes and allows modifying record layer through cipher suites selection (**T1** and **T2**). It also supports various application layer protocols using (opportunistic) TLS (**T3**). X.509 certificates can be checked against standards, such as RFCs and ETSI, as well as other relevant policies, such as CA/B baseline requirements and browser PKI/CT policies. It also implements certificate validation similar to common browsers but without integrating proprietary CRL lists.

The shortcoming of ZMap Suite is twofold: (*i*) it only supports stapled OCSP and not OCSP or CRL (**X3**), and (*ii*) it lacks built-in means to correlate DNS and X.509 to relevant DNS records (**X4**).

TLS-Attacker Suite. A collection of tools and libraries in Java with a focus on security analysis of TLS servers and clients. Similar to ZMap it is capable of performing Internet-wide TLS scanning and vulnerability detection. It also allows for fine-grained manipulation of TLS handshakes and exchanged messages (**T1** and **T2**), and provides tools to parse and generate X.509 certificates (**X2**). Support for Web PKI aspects, however, are rather limited (see Tabelle 4.5.1).

testssl.sh. A portable script to analyze security of SSL/TLS servers, testssl.sh checks for known TLS vulnerabilities (**T1** and **T2**), and simulates TLS handshakes for a variety of known clients (*e.g.,* Firefox 100 on Win 10). It also supports automatic certificate validity checking over CRL and OCSP endpoints (**X3**), and looks for DNS CAA records (without matching however). This tool is rather suitable for debugging single noticeable observations from measurement datasets.

Disadvantages of testssl.sh can be summarized as follows: (*i*) its lack of an API, and its monolith design that does not support modification without changing the core codebase, (*ii*) insufficient certificate information (**X2**) and lack of an option to store the full certificate chain, and finally (*iii*) file storage for results causes slow-downs in high concurrency measurements due to file system operation overhead (see Abschnitt 4.7).

Goscanner. A command line tool written in Go language. It has the ability of fingerprinting TLS handshake using various methods [281], [227] (**T1**), validating and storing X.509 certificate chains (**X1**), and using custom handshakes when contacting servers. The output is stored in multiple files in CSV format.

Puppeteer. A Node.js library (JavaScript) that enables programmatic control of browsers which support Chrome DevTools Protocol (CDP)[7]. Although limited to the Web, measurements using Puppeteer can precisely simulate user experience (see Abschnitt 4.8). It can, for example, properly follow redirects of various types (*e.g.,* HTML and JavaScript) and extract certificates as presented by the browser to users.

Puppeteer, however, is limited in scope by four factors: (*i*) the CDP protocol does not provide access to lower level API and many features are still experimental (*e.g.,* retrieving certificates), (*ii*) Chromium configuration flags that define its behavior upon initiation are not well-documented, (*iii*) page naviga-

[7] `https://chromedevtools.github.io/devtools-protocol/`

tion (equivalent to opening a tab in browser) spawns multiple processes and consumes a relatively high amount of resources (computation and memory). As the number of concurrent Puppeteer jobs increases, Puppeteer fails to properly kill Chromium process and free up memory. Finally, (*iv*) underlying browsers used by Puppeteer might exhibit non-standard behavior. Chrome, for example, artificially generates an HTTP redirect (status code `307`) when encountering an HSTS header even if the server does not explicitly indicate a redirect.

Zeek. Formerly known as Bro, Zeek is an online monitoring and analysis tool that is commonly used for passive measurements. It has its own scripting language that allows for customizing its behavior. Zeek lends itself to selectively collect network packages on transit and provides analysis to some degree. Note that Zeek can only decrypt TLS data when encryption keys are provided. Furthermore, the capabilities regarding X.509 certificates are rather limited.

4.6 Preparing Measurements

Measurement studies take various steps in advance to ensure correct design and verifiability, as well as to avoid potentially negative impact of the measurement. Drawing from experiences of existing studies, here we introduce established ethical considerations, provide a brief discussion on proper documentation for reproducibility, and show the importance of software/hardware validation in advance.

4.6.1 Ethical considerations

Performing measurements can be a source of disturbance or may jeopardize safety and privacy of involved persons. To reduce negative impact, both on systems and people, measurement studies address potential ethical issues in advance.

In case of active TLS measurements, specifically large-scale or Internet-wide, the scanning puts a burden on network resources [11] and can be regarded as a malicious attack on the network. As a countermeasure, infrastructure operators, *e.g.,* CDNs, commonly implement detection and mitigation measures that terminate TLS connections or drop TCP packages altogether if the source is deemed as a scanner [252]. Not only targets, but also intermediate networks, are affected by measurements. Durumeric, Wustrow, and Halderman [69, §5] recommend 7 practices that address ethical aspects of Internet-wide measurements ranging from coordinating with local network admins to information provision at source DNS and IP addresses, and catering for simple opt-outs. These practices are, for example, implemented by Cencys [65, §2], an Internet-wide scanner (see also Abschnitt 4.4).

For passive measurements, legal challenges and the privacy aspect is more prominent [117]. In contrast to active measurements, an a priori knowledge of affected parties is not given. As such, prior consent cannot be fetched from those, whose data is captured during the measurement. So, measurement studies rely on sampling, anonymization, desensitization, or similar post-processing methods to remove private information [117].

Applicable to both passive and active measurements are non-technical aspects. Partridge and Allman [190] go beyond technical means and discuss ‘tangible harm to people’ that can be caused by active measurements. In a recent paper, Pauley and McDaniel [191] provide an overview of previous work of ethical measurement, summarize existing community guidelines, and discuss recommendations to establish an ‘ethical framework’ for Internet measurements.

4.6.2 Documentation for Reproducibility

It is common for measurement studies [77] to cater for reproducibility, *i.e.,* it should be possible to reach to the same results using the same procedure or artifacts [278]. We refer to related work [16], [17], [34], [55], [215] for an extensive discussion of best practices and challenges of reproducibility. Here, we confine ourselves to a brief overview regarding software and data.

Software. In Abschnitt 4.5, we discussed a number of popular open source tools among measurement researchers. When using such off-the-shelf software, noting the exact version is essential for reproducibility. For open source software a reference to specific state, *e.g.,* a git branch or commit, helps others to maintain the same software setup. If modification are applied to the software, a brief documentation, *e.g.,* using inline comments, accompanied by a short reason, *e.g.,* bug in original software, is helpful for reproductions. Nawrocki *et al.* [178], for example, published their modified version of Quiche (QUIC client) and quicreach (QUIC ping tool) alongside measurement data, while the changes by the latter were merged with the original codebase.

Complimentary information with a direct impact of the measurement, such as parameters, inputs, or a description of vantage points, should also be provided. This also includes implicit data that utilized software might implicitly (*e.g.,* due to convention over configuration) rely on, for example, the set of root certificates used by OpenSSL which differ from one execution environment to the other. Tehrani *et al.* [84], for example, publish separate files consisting of root certificates used for validation, CT logs used to retrieve certificates, certificate policies, *etc.* to enable reconstructing measurement setup just as the time execution.

Furthermore, the list of dependencies (software and system alike) should be documented and provided in a readily installable format, *e.g.,* `pip freeze` and `pip install -r` for Python software. Optimally, a configure script (*e.g.,* generated by GNU Autoconf) can help other researchers to check their systems for missing libraries and other dependencies. Darwich *et al.* [52] (the best artifact at IMC ’23), for example, combines Poetry (Python dependency manager), and Docker (containerization solution) in a shell scrip to fetch and setup all dependencies at once.

Measurement artifacts are commonly accompanied by a ‘readme’ file containing instruction on how to start, configure, and optimize each program. This also includes structure and content of generated output and error codes alongside a step-by-step manual to execute a complete measurement. To replicate a specific measurement relevant information, such as configuration and input parameters (*e.g.,* list of measured hosts) alongside deployment settings (*e.g.,* vantage point), are also provided.

Source code, configurations, and even dependencies can be bundled together, *e.g.,* in a container image, to cater for easier and faster bootstrapping. Some tools also allow defining virtual environments that can be exported and imported on any arbitrary system.

Data. Publishing measurement data, specifically in its original format (*i.e.,* unprocessed), is advantageous in two regards, (*i*) it enables validation of own analysis by others, *e.g.,* in replication studies, and (*ii*) it can be used by related measurements, *e.g.,* as complimentary data. For X.509 certificates, the standard format is ASN.1 binary, while TLS has its own record message that is commonly serialized in packet captures (pcap). Among surveyed studies with published data, we find both raw data (as described above), and processed data. For the latter, we find data format to be related to the utilized tools Abschnitt 4.5). For example, ZMap processes network packets and produces results in JSON format (tree structure), goscanner produces CSV files (tabular) with certificates in base64 PEM format, and Zeek stores raw packet captures (pcap). Regardless of structure, enhancing data with metadata describing its content, structure, timestamps, *etc.* is essential [16]. Providing both raw and processed data together, *e.g.,* as Holz *et al.* [365] do, allows validating processed data.

A measurement might consist of multiple phases where output from one phase is fed as input to another phase. Providing output for each step, instead of only the end results, is helpful for others to comprehend the measurement procedure and also to detect any potential errors that might be masked when data is merged. For example, in place of stating if a host supports stapled OCSP, TLS handshake containing OCSP response alongside the CA certificates (from local trust store) used to validate it should be provided.

Beyond collected data, best practices also recommend accompanying code to generate end results (*cf.,* Scheitle *et al.* [215]). For example, for their measurement study of DANE, Lee *et al.* [140] provide extensive instructions on a dedicated website[8] with scripts to analyze data and create plots. A tool that has become popular for this mean is Jupyter notebook [430], a method to combine code, documentation, and visual elements (*e.g.,* plots).

4.6.3 Validation of Software and Hardware

The final step in preparing measurements is to validate the software and hardware before executing the complete measurement. Wan *et al.* [252], for example, runs their measurement in small scale (1% or IPv4 address space) from all selected vantage points to confirm that both software and hardware can reach desired scanning speed while avoiding extraordinary packet drops.

4.7 Performing Measurements

In this section, we discuss the importance of maintaining temporal integrity while collecting data from different sources, different types of stateful measurements, detecting and handling errors during measurement, and finally how to best store data.

[8] See `https://dane-study.github.io/security2020/server-side/`

Tabelle 4.7.1: Selected measurement features which are temporally bound and are collected at various sources

Type	Source	Collected at	Temporal indicator
DNS RR	Authoritative name server	DNS resolution	TTL field†
DNSSEC `RRSIG`	Authoritative name server	DNS resolution	Inception / expiration fields‡
X.509 cert	Web server / CT logs	TLS handshake	Not before / not after fields‡
TLS session	Web server	TLS handshake	Lifetime hint†
Stapled OCSP	Web server	TLS handshake	This update / next update‡
OCSP	CA OCSP endpoint	Certificate validation	This update / next update‡

† Relative / ‡ Absolute

4.7.1 Vantage Points

The physical location where measurements are performed can have a twofold impact on the results in terms of coverage, and consistency. As such, same measurement (method and target) from different vantage points can lead to different results.

Coverage. Wan *et al.* [252] discuss how host reachability can vary depending on the vantage point. They show how the origin location in single-probe IPv4 address space scans can lead to missing around 5% of all HTTPS hosts in the worst case (See Censys in Abschnitt 4.4). In our own measurements, we also observe similar phenomena, *e.g.,* none of the Iranian banks (in `IR-BANKS` dataset; §4.10) are accessible from vantage points outside Iran, so we use certificates from CT logs for our analysis.

Coverage itself, however, can be subject of measurement instead of being a mere measurement artifact. Raman *et al.* [198], for example, purposefully selects vantage points in various countries to measure respective Web censorship measures. In such cases it is desired to measure data from specific vantage points to study coverage, even by crowdsourcing the measurements through platforms (see **??**) to avoid detection or blocking while maintaining ethical aspects.

Consistency. Depending on the vantage point, the same domain name can be resolved and routed to different IP addresses, *e.g.,* to the nearest point of presence of a cloud provider. Lee *et al.* [142] show how different edge servers hosting content for the same domain name can be configured differently with respect to TLS security.

4.7.2 Temporal Integrity

There are numerous pieces of temporally bound data that are collected from various sources: Tabelle 4.7.1 provides an overview. To capture a correct snapshot, measurement studies make sure that these features are collected with short temporal discrepancies. That is to reduce the time distance between probing related features, for example between querying DNS `A` records and establishing the TLS connection, or fetching certificates and validating OCSP response. Nawrocki *et al.* [178], for example, notes how a short time difference between two measuring X.509 certificates can result in collecting different certificates for the same host due to key roll-overs. Zirngibl *et al.* [274] observe in their analysis of QUIC how Google rotates certificates for `googlevideo.com` *even* before these are expired. And finally, in their study of DANE use by SMTP servers, Lee *et al.* [139] note the necessity of maintaining temporal integrity and collects DANE records and X.509 in an atomic operation.

Abbildung 4.7.1 depicts temporal values for different records that we collected for our TRANCO dataset.

4.7.3 Stateful Measurement

Some aspects of TLS cannot be analyzed based on one-off and stateless measurements. In the following we discuss cases that require measuring same servers multiple times or measuring different servers in the same context and with respect to each other.

Same host and settings. Here, for each host multiple measurements are performed. For example, Liu *et al.* [149] observe that some servers deliver stapled OCSP only if it has already been cached locally [149, §4.3], so to determine if stapled OCSP is supported, multiple successive measurements are performed. In a later study [43], the authors expand their scope to investigation of OCSP responders using hourly measurements in a longitudinal study (4 months period) from multiple vantage points, practically forcing servers to always query and cache OCSP responses. This procedure is useful when intermittent caches have an impact on measurements.

Same host with different settings. In other cases, measurement settings and parameters are altered. For example, discovering how TLS sessions are implemented and deployed requires performing multiple handshakes with and without session IDs, tickets, or pre-shared keys. In a study from 2018, Sy *et al.* [233] regularly establish TLS connections to Alexa Top 1M sites and show how TLS resumption can be used to track visitors. Hebrok *et al.* [107] examine cryptographic issues of TLS resumptions by executing 10 TLS handshakes for every (virtual) host while storing session keys alongside session tickets. Furthermore, collected session tickets were modified to check for server-side authentication vulnerabilities. This procedure is useful in validating if servers can properly handle stateful operations.

Different hosts with same settings. In some cases, we need to perform subsequent measurements based on information retrieved during an initial measurement of a sample. For example to study TLS

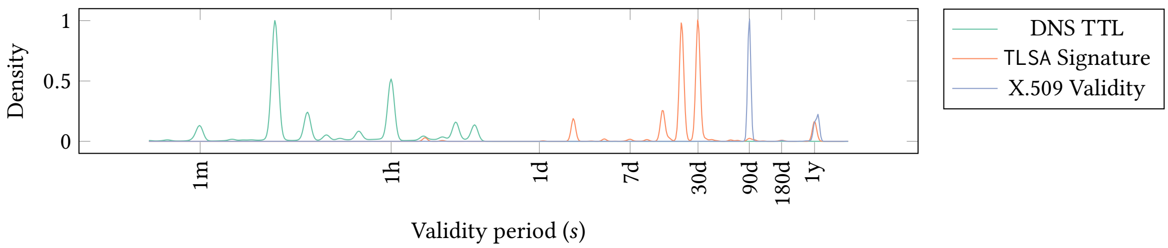


Abbildung 4.7.1: Validity period of X.509 certificates, DNSSEC signature validity of TLSA records, and actual remaining TTL of A records at the time measurement from our TRANCO dataset

sessions that are shared among virtual hosts at the same or among different servers. Sy *et al.* [234] show performance advantages of sharing session among domain names given that they are all controlled by the same entity. Similarly, Springall, Durumeric, and Halderman [228] study shared TLS session among different domain names but set the focus on security ramifications instead of performance. For this, session tickets were collected regularly over a period of 9 weeks and finally compared with each other to detect, among others, security risks of servers reusing session cache, session ticket, and Diffie-Hellmann parameters. This procedure is useful in analyzing the impact of shared resources, *e.g.,* caches, by different hosts.

Different hosts with different settings. Finally, there are cases that an initial measurement of a single sample are used to define and perform measurements on related samples but under different settings. An example is the family of TLS confusion attacks. This is an umbrella term referring to a type of MITM attacks where an attacker manages to divert TLS traffic from the intended server to another without the client noticing.

Delignat-Lavaud and Bhargavan [54] show how a TLS terminating proxy server can be tricked into redirecting HTTPS traffic meant for a virtual server to another. The anecdotal examples of this study are extended by the systematic analysis of Zhang *et al.* [266] where for a set of host names sharing an X.509 certificate TLS connections are made to different IP addresses and ports. Brinkmann *et al.* [31] introduces a cross protocol variation of TLS confusion attacks where traffic is diverted from one protocol to another, *e.g.,* HTTPS to SMTPS. This procedure is useful in analyzing security and privacy aspects among logically related hosts.

4.7.4 Failure handling

It is integral to detect errors early on and be capable of resuming measurements without having to start atop. In the following, we draw on our own experience performing TLS and Web PKI measurements, and discuss various aspects of failure handling.

Detecting causes of errors and failures. Errors and failures can be rooted in bad configurations, *e.g.,* missing SNI in TLS handshake, faulty implementations, *e.g.,* wrong implementation of CAA relevant set [357, §3], network failures, *e.g.,* selected DNS resolver fails to handle DNSSEC, *etc.* Some errors are intermittent, *e.g.,* due to server outage, and are solved upon repetition, while others persist.

Validating software and hardware (see Abschnitt 4.6) with a small sample set before starting the actual measurements is a quick method that can reveal bad configuration or software/hardware bugs. It also gives a first impression about measurement complexity in terms of execution time, storage requirement, *etc.* which can be used to detect anomalies and potential failures during the actual measurement, *e.g.,* if a single probe taking ‘too long’.

Additionally, we should monitor the measurement to detect dormant causes of errors and failures. For this, we can consult logs. Nearly all tools discussed in Abschnitt 4.4 support debug mode which can be used to log extra operational information. Debug and error messages, however, are not always in machine-readable format or include necessary information (*e.g.,* passed arguments) to detect causes for failure. Another (complimentary) approach is to dump data packets in transit during measurements, *e.g.,* using Zeek, and to dissect them in case of failures that cannot be debugged otherwise.

Detecting measurement artifacts. In addition to technical failures, such as software bugs, that manifest themselves in error logs, software crashes, *etc.*, deficiencies in measurement techniques can lead to measurement artifacts. This kind of error is harder to detect and commonly materialize itself after inspecting measurement results. For example consider the measurement agent being marked as bot by cloud providers or having a TLS connection terminated due to lack of SNI. In both cases the logs will not indicate any failures, but the software has practically failed to capture a truthful snapshot. Although not all measurement artifacts can be avoided, it is important to monitor measurement for any cases that might serve as an indication.

Resuming and repeating failed measurements. Specially in case of large-scale active measurements or limited resources, it is desirable to be able to resume failed measurements, *i.e.,* skip successful measurements, rerun failed ones, and continue with the rest. If intermediate data is stored separately (see Abschnitt 4.6), measurement can be started at the point of failure instead of from atop. See Abbildung 4.B.1, for example, where for our own measurements we separate reachability test, application detection, and TLS communication from each other, and generate separate files for each step.

None of the tools with built-in batch process (see Tabelle 4.5.1) supports measurement resumption. To address this shortcoming, tools can be wrapped inside process managers, *e.g.,* GNU Parallel [238], that are able to parallelize and monitor processes and cater for resumption in cases of failures. An additional advantage of using process monitors is that failure of a single probe does not impact the rest. Furthermore, such process managers allow adapting number of parallel jobs to available resource, or to stop the whole process if given threshold on failed jobs is reached, *e.g.,* due to network failures.

4.7.5 Data Storage

Measurement data are commonly stored in traditional databases (*e.g.,* TLS-Attacker), flat-file databases (*e.g.,* Z-Map suite), or simple files (*e.g.,* per certificate, per TLS handshake, *etc.*). Storing each feature in a single file simplifies searching and filtering but the added overhead due to file system and disk I/O operations can lead to throttling measurement speed and parallelization.

In practice, a mixture of all strategies might be chosen. Holz *et al.* [110], for example uses Z-Map to discover viable hosts, collects data using OpenSSL and stores the result in a database.

4.7.6 Reporting

If security vulnerabilities are detected during measurements, researchers are encouraged to inform affected entities through *Coordinated Vulnerability Disclosure* (aka Responsible Disclosure). Governmental organization (*e.g.,* US CISA or Germany's BSI), standardization institutes (*e.g.,* NIST or ETSI), and other relevant entities have their own guidelines and procedures to document and submit vulnerabilities. The *CERT Guide to Coordinated Vulnerability Disclosure* [366] is, for example, a comprehensive guide by Carnegie Mellon University.

4.8 Interpreting Data

Insights from our survey and own experience reveal common pitfalls when interpreting TLS and Web PKI data.

4.8.1 Interpreting Web PKI Identities and Identifiers

Every X.509 certificate denotes the identity of its owner (subject) and its issuer. Each CA has its own policies and procedures to denote subjects and issuers in a certificate. In the following we discuss some aspects that should be noted when interpreting subject and issuer fields.

Certificate subjects. It might be useful to detect the subject of a certificate, for example, to detect if a certificate is being shared among distinct service providers [33], [147]. Depending on validation type (DV, OV, IV, and EV; see Abschnitt 4.2) corroborating data might be needed to uniquely identify a certificate owner.

For EV certificate, the subject name explicitly denotes the entity running a website, *i.e.,* the organization responsible for its content [306, §2.1.] and is sufficient to extract the certificate owner. In contrast, OV and IV certificates identify the certificate applicant [307, §3.2.2.]. In case of *delegated services* [147], *e.g.,* when a CDN operator vicariously applies for a certificate, the subject might actually identify the delegatee and not the delegator and cause a misinterpretation of the certificate. Cloudflare, for example, uses OV certificates with its own identity information for its free-tier customers —a marketing stunt that can be avoided by upgrading to paid plans [384].

DV certificates pose the biggest challenge, as the only subject information is provided in form of SANs (domain name, IP addresses, etc.) and cannot be used to identify the certificate owner. Although SAN entries are supposed to be *alternative names* of the same subject, there is no guarantee that all entries actually denote the same owner. Multitenant infrastructure operators, *e.g.,* Google and Imperva, list domain names of different customers as SANs in the same certificate. To address such issues, Cangialosi *et al.* [33, §4.1], for example, devise a method based on e-mail addresses included in DNS WHOIS data to infer if two domain names belong to the same organization. This method, however, might be less fruitful for future measurements as information in WHOIS databases are more restricted nowadays due to privacy concerns and regulations such as EU GDPR.

Certificate issuers. Each certificate denotes its issuing and signing CA in the issuer field. Understanding the role of CAs in the Web PKI has been part of research. Durumeric *et al.* [67], for example, give a detailed overview of CAs (*e.g.,* country of origin and market share) alongside an analysis of which type of organizations (*e.g.,* libraries, museums) are awarded with unconstrained intermediate CA certificates (see Abschnitt 4.2). Fadai *et al.* [71] go further and investigate CAs in terms of corruption, human rights, *etc.* at their respective country of origin.

Identifying the operator of a CA using only its (subject) name is, however, a non-trivial task as Ma *et al.* [154] notes. The authors show, for example, how a given root CA does not refer to its operator, DigiCert, in its subject name (`CN=Hotspot 2.0 Trust Root CA - 03; O=WFA Hotspot 2.0;C=US`).

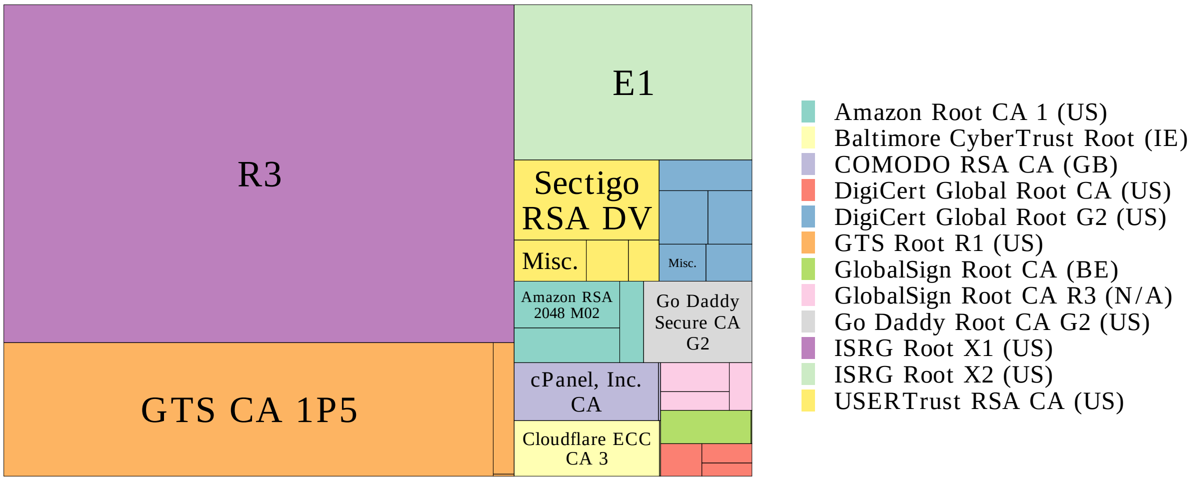


Abbildung 4.8.1: CA market share from our `TRANCO` dataset. Each colored box denotes a trust anchor (root certificate) and nested boxes are intermediates. *Misc.* denotes an aggregated share of multiple intermediates. The size of each box is directly proportional to the number of leaf certificates issued by respective intermediate CA. Some labels are removed due to space constraints.

The DigiCert PKI repository[9] also reveals a number of other root CAs operated by DigiCert but under different subject names such as Baltimore CyberTrust Root, GeoTrust Primary Certification Authority, *etc.* The respective impact on research data interpretation is twofold: (*i*) analysis that is based on identity of CAs requires additional care. For example, 5 out of the 14 listed CAs in the market share statistics by Aas *et al.* [1, §7] all belong to DigiCert. And, (*ii*) knowing the actual operator behind CAs helps understanding specific trends. For example, as discussed in the next section, why all banks in our `IR-BANKS` set are subscribers to an obscure CA (Unizeto).

Abbildung 4.8.1 depicts the market share of CAs from our `TRANCO` dataset divided by trust anchors (colored boxed) and intermediate CAs (nested boxes). The size of each box is directly proportional to the number of leaf certificates issued by respective intermediate CA. More than half of certificates are issued by Let's Encrypt intermediates R3 and E1, respectively under trust anchors ISRG Root X1 and ISRG Root X2. There are also trust anchors that are exclusively used by a single company, *i.e.,* GTS Root (Google), Amazon Root CA, and GoDaddy Root CA. We also observe PKI operators such as USERTrust and DigiCert that provide PKI as-a-service and manage intermediate CAs for third parties. It is also notable that nearly all root certificates in this set are under US jurisdiction, even Baltimore CyberTrust Root which denotes Ireland as country in its subject field is part of DigiCert since 2015.

[9] `https://www.digicert.com/kb/digicert-root-certificates.htm`

4.8.2 Corroborating Data using CA Operational Policies

All relevant information from applicability of a certificate to its issuance, revocation, and definition of custom OIDs and CAA identifiers are defined in CP and CPS [338], [470] of its issuing CA.

These documents might be helpful in better understanding abnormalities in measurements. For example, all Iranian banks (in `IRB` set) use certificates issued by *Unizeto Technologies*, a Polish CA, while none of the banks in our `ECB` set (except the *PKO Bank Polski* in Poland) subscribe to that CA. Consulting CP and CPS document for the QuoVadis and DigiCert (the former being a subsidiary of the latter), which cover more than half of all European bank and credit institutes, reveals that DigiCert is prohibited by the US law to provide any services to countries or companies on a *government denied list* [321, §4.1.1], *e.g.,* Iran.

When considering CP and CPS documents, three aspects should be regarded. First, although RFC-3647 [338] streamlines the structure of CP and CPS, the terminology and concepts applied might vary among CAs. Second, in practice CAs do not necessarily behave in conformance with their own or other relevant guidelines. For example, the *ECC CA-3* intermediate CA certificate from Cloudflare[10] carries OIDs for DV, OV, and IV validation types, and violates both its issuer CP (IV certificates are issued to individuals) [322, §3.2.2] and CA/B Baseline Requirements (IV certificates must carry `surname` and `givenName` in their subject field) [307, §7.1.2.7.3]. And third, there is no guarantee of logical coherence. Let's Encrypt, for example, asserts no relationship between subscribers and registrants of domains names in a certificate [374, §3.1.4] in its combined CP/CPS and in the same document requires subscribers to be the legitimate registrant of respective domain names [374, §9.6.3].

4.8.3 Correctly Evaluating Certificate Validity

Authentication presupposes presentation of a valid certificate. Beside formal requirements (see RFC-5280 [297]), a relying party dictates the validation methods and validity criteria. The most basic form of validating a certificate is *path validation* described in RFC5280 [297]: validation succeeds if certificate subject matches the desired subject and if there is trusted path to a trust anchor (root). Depending on the relying party (user) or the client it uses (*e.g.,* a web browser), the set of trust anchors and validation policies might differ. Respectively, to verify if a collected certificate is valid three features needs to be defined: (*i*) trust store, (*ii*) revocation status, and finally (*iii*) additional policies (see Abbildung 4.2.3).

Trust stores. Major browsers, operating system vendors, governments, and even individual organizations maintain sets of *trustworthy* certificates (intermediate and root CAs; see Abbildung 4.2.4). Although users are generally oblivious to their role, trust stores are integral in securing or jeopardizing the security [405], [226] of users daily communications.

Each trust store maintainer has its own set of requirements for incorporating a CA, for example, through consensus as it was the case for Debian OS (`ca-certificates` package [456]), well-defined policies (including undergoing audits and conforming to requirements) as in Mozilla Root Store Poli-

[10] `https://crt.sh/?id=2392142533`

cy [414], or by establishing a legal framework as in the EU eIDAS Regulation [332, Article 24]. Related work has extensively studied root stores and their relation to each other [71], [127], [152], [431], [195].

Beyond established trust stores, users may also decide to trust CAs by adding them manually or through software installations. It is, for example, common practice for online banking software in South Korea to install additional CAs in local trust stores [429]. The impact of such settings can also be observed in TLS measurements. A study from 2016 [42], for example, notices Korea Telecom as one of top Autonomous Systems that host invalid certs presumabely because the researchers rely on generic trust store for validations. Zhang *et al.* [269] study the prevalance of such 'hidden root CAs' within the Web PKI ecosystem.

Revocation status. A certificate that fulfills path validation might still be invalid if it has been revoked. Depending on the issuing CA, relying parties can check the revocation status using Certificate Revocation Lists (CRL) [297] or the Online Certificate Status Protocol (OCSP) [448]. Let's encrypt, for example, only provides revocation checks through OCSP. Web servers can also staple the OCSP response in the TLS handshake if a client explicitly requests it.

In addition, there are proprietary alternatives which aim to improve performance or reliability of CRL and OCSP. Google, for example, maintains a list of revoked certificates (CRLSets [294]) partly by crawling CRL lists and uses this in Chromium-based browsers. Microsoft [407], Mozilla [352], and Apple [286] also have similar revocation lists used in their products. It should, however, be noted that such proprietary solutions have been show to be non-exhaustive. Liu *et al.* [149], for instance, claims that CRLSets only cover 0.35% of all revocations as observed by researchers.

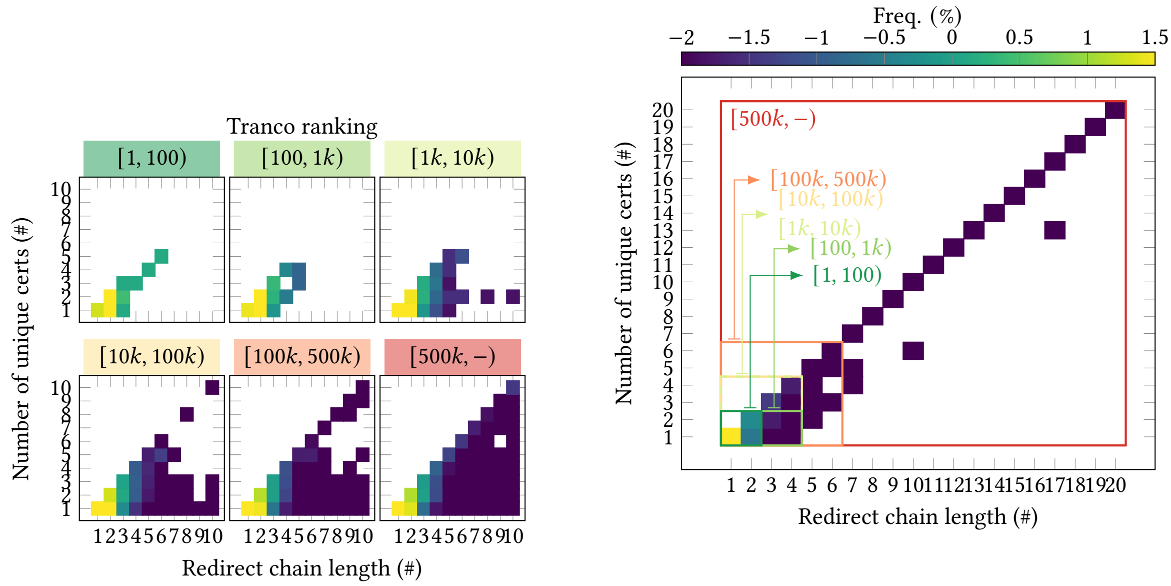
Relying party policies. Relying parties, *e.g.*, browsers, may also define additional validation policies. For example, Apple products such as the Safari browser require at least 2 SCTs for certificates with a validity of less than 15 months [284].

4.8.4 Properly Understanding Users Perspective

Users interact with TLS-secured server through a client: a web browser, mail client, smartphone app, or alike. Depending on measurement approach, a discrepancy might arise between what collected data reveals and what users actually experience in their daily interactions over the Internet. Here we show how users experience using a web browser might differ what measurements might reveal and discuss 4 aspects: (*i*) point of access, (*ii*) redirects and resources, (*iii*) certificate validity, and (*iv*) bot and intrusion detection.

Point of access. Domain names were introduced to take the burden of remembering IP addresses from users. However, instead of remembering domain names and navigating to them, users navigate the Internet through the lens of search engines, social media, and streaming services [325]. Research shows that users in part take browser address bar for a *search bar* and have difficulties in differentiating components of domain names and URLs in general [206].

When choosing a data sources (see Abschnitt 4.4), it should be taken into account that users might never (directly) encounter or interact with items in that source. For example, `mtalk.google.com` —at



(a) Redirect (HTTP and HTML) chains constructed by custom script based on `libcurl` (see **P.3** in Abbildung 4.B.1).

(b) Redirect chains constructed by Puppeteer headless browser (see **M.1** in Abbildung 4.B.1). Only valid certificates are considered.

Abbildung 4.8.2: Frequency of unique certificates in redirect chain of a given domain name in relation to redirect chain length for hosts in our TRANCO dataset.

position 110 of Cisco Umbrella Top 1M (for list `2023-06-21`)—is an endpoint used by Google for push notifications and is never visited directly by users. The same applies for IP addresses: when scanning the IP address space, endpoints such as home routers might be discovered that are publicly accessible by potential misconfigurations.

Redirects and resources. Redirects are common on the Web: HTTP (3xx status codes), JavaScript (`window.location` property or History API), or HTML (`http-equiv` meta tag) all provide means to redirect a URL to another. During redirects, browsers establish new TLS connections that are not signaled to the user. For example, navigation to `google.com` triggers a redirect to `wwww.google.com` whereby only the connection and certificate information for the latter connection are presented to the user. The connection attributes or certificates provided at each redirect are not necessarily the same as the previous or next step; `google.com`, for example, provides a different certificate compared to `wwww.google.com`. Furthermore, when users navigate to a webpage, additional TLS connections to possibly different hosts are established to fetch resources such as scripts and style sheets.

The role of redirects can be observed from the measurements of our TRANCO dataset as depicted in Figures 4.8.2a and 4.8.2b. The former figure depicts the frequency of total HTTP and HTML redirects in relation to the total number of unique certificates as observed by our custom `libcurl` script (corresponding to step **P.3** in Abbildung 4.B.1). Although the majority of domain names are redirected less than 3 times with 1 or 2 unique certificates in the redirect chain, we observe that as the ranking in-

Tabelle 4.8.1: Difference between the length of valid certificate chains as delivered by servers in our TRANCO dataset and the length of the shortest valid chain built against Mozilla trust store (frequency in %).

	Length difference with the shortest verifiable chain					
Tranco ranking	-3	-2	-1	0	1	2
[1, 100)	0	0	0	69.23	25.64	5.13
[100, 1k)	0	0.14	0	73.13	16.76	9.97
[1k, 10k)	$1.45 \cdot 10^{-2}$	0.31	0	68.99	20.33	10.36
[10k, 100k)	$2.85 \cdot 10^{-3}$	0.8	$1.42 \cdot 10^{-3}$	65.55	25.28	8.36
[100k, 500k)	$8.04 \cdot 10^{-3}$	1.01	$1.93 \cdot 10^{-3}$	67.16	26.72	5.1
[500k, −)	$6.81 \cdot 10^{-3}$	0.78	$8.29 \cdot 10^{-4}$	69.79	26.78	2.64

creases (*i.e.,* popularity decreases) both the length of redirect chain and number of unique certificates increase. Note that certificates are not validated here. The host names of TLS secured servers at the end of each redirect chain is then fed into a browser (corresponding to step **M.1** in Abbildung 4.B.1). Abbildung 4.8.2b shows the results. Here, JavaScript redirects are also followed while only valid certificates are considered. Similarly, we observe that over 98% of all hosts having no extra redirects.

Certificate validity. Understanding how often users face invalid certificates, requires understanding the validation context as described above. The impact of having the proper context is reflected in the discrepancy between studies that use generic contexts and those using the actual context of relying parties, *e.g.,* browsers. Oakes *et al.* [184], for example, collect complete certificates chain directly from users' machines and observe that 99.7% of all collected chains, *i.e.,* those used by clients, are valid. In contrast, other studies based on IPv4 scans on generic root stores classify the majority of certificates (ranging between 66% and 88%) as invalid [42], [73]. Similarly, a study by Holz *et al.* [110] observes that passively collected data exhibit salient percentages of valid certificates in comparison to certificates collected by active scanning of IPv4 address space.

Another source of discrepancy is the proprietary revocation lists discussed previously. Liu *et al.* [149], for example, show how CRLSets (Google) only cover a fraction of all certificate revocations, so there is a chance that a Chromium-based browser would validate a certificate even though it has been revoked. Note that none of the tools introduced in Abschnitt 4.5 (excpet Puppeteer) have integrated nor apply these lists.

Finally, user clients are more robust in correcting malformed certificates chains (wrong order, duplicate certificates, missing intermediates, *etc.*) which might, for example, cause OpenSSL to mark a certificate chain as invalid when using default settings. In our measurement of TRANCO dataset, we observe that about 70% of all hosts provide the shortest verifiable trust chain (from leaf to root) as depicted in Tabelle 4.8.1 (column 0). The rest is either providing more certificates (columns 1 and 2), *e.g.,* cross-signed variation of root certificates, or fewer certificates (columns -3, -2, and -1), *e.g.,* missing intermediate certificates. Whereas providing extra certificates can be considered as an attempt to increase compatibility, *e.g.,* to cover cases where a root certificate is not yet included in a trust store, leaving intermediate certificates out might lead to certificate invalidation, *e.g.,* when intermediate certificates are not explicitly configured by the relying party.

Bot and intrusion detection. Some infrastructure operators, *e.g.,* Akamai [280], offer *bot detection* solutions to block unwanted or malicious access. Such systems might block measurements if they flag the utilized tool as non-human and skew the results respectively.

4.8.5 Practically Analyze Security Impact

TLS attacks make use of protocol flaws, implementation deficiencies, or cipher suite and extension weaknesses. For an attack to be realistic, both endpoints must be susceptible to the respective attack vector. For example, while security shortcomings of TLS 1.0 and 1.1 are known (see RFC8996 [412]), none of mainstream browsers support these versions anymore, so that a web server supporting TLS versions 1.0 up to 1.3 does not in practice pose a higher security risk to users (with an up-to-date browser) than a server only supporting TLS 1.3.

Passive measurements provide a better basis for assessing the real impact of a security vulnerability. Kotzias *et al.* [128], for example, use passively collected TLS handshakes to detect if clients actually make use of insecure cipher suites or the vulnerability remains only on the server side.

4.8.6 Consistently Use Data from Mixed Sources

In Abschnitt 4.6, we discussed the impact of vantage point and importance of maintaining temporal integrity. These aspects must also be regarded when interpreting data from mixed sources. Cangialosi *et al.* [33], for example, use WHOIS data from third party sources to distinguish between domain name owners and subsequently detect private key sharing among distinct service providers. The temporal discrepancy between data from the original and the third party sources can lead to false identification of a domain name owner. A study by Liu *et al.* [149] is another example that relies on data from both passive and active measurements from different sources and times.

4.8.7 Using CT Logs to Fill Informational Gaps

A measurement might not capture all the information required to answer a research question. For example, active TLS scans are going to only cover deployed and not backup certificates [384], thus not suitable to calculate global CA market share. In cases, where certificates are missing in the original dataset, CT logs can be used to fill those gaps.

In our `TRANCO` dataset, for instance, we encounter TLSA records, that do not authenticate the deployed certificate. From CT logs, however, we were able to collect the majority of certificates referenced by DANE. As depicted in Abbildung 4.8.3, while less than half of mismatching TLSA records point to valid, yet non-deployed certificates, the rest reference certificates that are long expired or removed from trust stores.

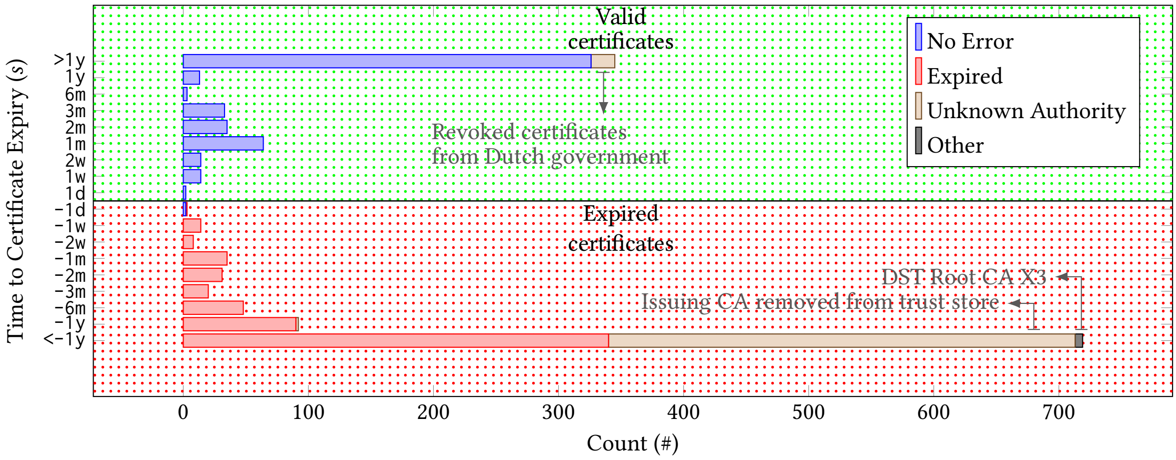


Abbildung 4.8.3: Certificates from CT logs that match TLSA records but are not deployed on the web server, grouped by their relative expiration time and annotated with their validation state.

4.8.8 Correctly Incorporating DNS Data

DNS, TLS and Web PKI are intertwined. Relying parties and CAs use data from DNS for various purposes (see Unterabschnitt 4.2.3). Analyzing records that are meant for users, *e.g.*, DANE or MTA-STS, are less challenging than those meant for use by CAs. User-facing records are expected to be up-to-date at any time, while CA-facing records, *e.g.*, CAA or authorization tokens, are in practice checked only once at the time of issuance. To examine this type of records, *e.g.*, to validate CA procedures, we need to have the same DNS records as seen by a CA. From a third party perspective, the exact time of issuance remains unknown and can only be inferred from the certificate when deployed or CT logged. So, there is always a time discrepancy between CA query of DNS, and the measurement time.

The effect of such time disparity can be highlighted on example of CAA. Abbildung 4.8.4 depicts the matching state of deployed CAA records in our TRANCO dataset. The share of mismatching certificates grows with increasing certificate age, indicating that CAA records have changed since the certificate

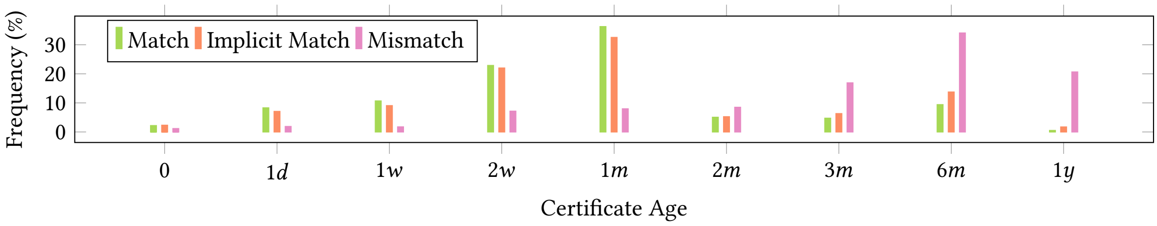


Abbildung 4.8.4: CAA matching states by age of certificates with matching subject name at the time of measurement. Implicit matches denote existence of CAA records that do not constrain issuance. 75% of certificates are younger than 3 months. CAA mismatches appear relatively more often in certificates older than 3 months.

issuance. This effect has also been shown by previous research [79], [213]. Frequent measurements (to observe key rollovers), regular scanning of CT logs (to observe cert issuance), or having access to historic DNS data, can help reduce this discrepancy.

4.9 Lessons Learned

Before we conclude, we reflect on lessons learned while composing this works, surveying the related work, and performing our own measurements. This section provides a set of insights useful for any general Internet measurement followed by remarks specifically for TLS and Web PKI measurements.

4.9.1 General remarks

Measurements must be ethical. Measurements directly impact others such as network administrators, infrastructure operators, and host owners. These should be informed about running measurements and be provided with means to opt-out from measurements, or to request information removal from collected data. In case of security measurements, any detected vulnerability should be reported to involved parties. Ethical codes and established best practices can guide researchers in this matter.

Reproducibility should be an integral objective. Other researchers must be given a chance to reproduce measurements and to validate underlying methods. Beside providing collected data, measurement setup, and execution must also be publicly documented. This includes documenting software version, underlying environment, exact configuration, and applied parameters. Ideally, all these are provided in a self-contained format such as container images with quick start scripts and documents.

Lurking variables skew results. Occasionally, the causalities inferred from measured data might be influenced from lurking variables. For instance, in two similar studies [42], [73] seven years apart, same researchers come to the conclusion that a majority of certificates are invalid. However, this can be traced back to the measurement method and lack of SNI this fails to cover virtual hosts and servers that require SNI. When analyzing, careful thought must be given to remove such confounders.

4.9.2 TLS and Web PKI specific remarks

Data sources vary greatly in quality and capabilities. Every data source has its own limitations. These shortcomings are, however, more than often not factored in during data analysis. Beside existing evaluation of various sources [137], [211], the quality of a data source can be verified using a sample set (as we did in Abschnitt 4.4) or through comparison with other sources on a small scale. Specifically for TLS and Web PKI measurements, hit lists provide a good starting point due to their manageable scope and order. Tranco top list is well-established among researchers, while Google CRUX has a better quality, specifically when focusing on the Web. CT logs provide easy access to X.509 certificates, yet relevant metadata (*e.g.,* deployment status) are missing. Finally, the IP address space is less suitable due to its low coverage of actual existing hosts (*e.g.,* virtual host).

Vantage point impacts dataset. The access point from which measurements are performed has an impact on the final dataset. Among the reasons are inconsistent TLS configurations or certificates on CDN edge servers, geo-specific policies (*e.g.,* geo-fences), and non-conformant middleboxes manipulating data on the fly (*e.g.,* dropping DNSSEC records when querying for DANE).

In some cases, this impact is subject of the study, *e.g.,* when researching censorship or topological network effects. In other cases, however, vantage point might turn out to be a confounder. Specifically, when data is sourced from a third party, *e.g.,* an indexing service. To eliminate this effect, or at least to become aware of its extent, a small sample must be probed from other vantage points as comparison. If possible, and specially for global measurements, various vantage points can be used to perform multiple measurements of the same sample set.

Tools in production are most reliable for measurements. Nearly all general-purpose programming languages offer implementations of common Internet protocols, *e.g.,* TLS and HTTP. However, not all are appropriate for measurement purposes. For example, the standard crypto library of Node.js, a JavaScript engine, is limited to parsing selected fields from an X.509 certificate, and implementing client-side session tickets is a tedious task over its TLS API. At the same time, own implementations are prune to errors that might remain undetected when not tested extensively. The Heartbleed bug, for example, was introduced by the author of TLS Heartbeat extensions RFC [483], and later detected by security team at Google. Thus, we encourage researchers to use established tools (see Abschnitt 4.5) for their measurements.

Among these tools, some are also used in production environments. For example, ZMap is used by Censys, goscanner is used for regular scans at TU Munich, OpenSSL is integrated in TLS software, and Puppeteer uses the same engine as the Chrome browser. As such, these are generally well-maintained and up-to-date. ZMap, and goscanner are furthermore optimized for large-scale scans, for instance, by automatically blacklisting local and reserved IP addresses.

In some cases, it is necessary to combine different tools to reach a measurement goal. This can be done if these tools support same output and input formats, *e.g.,* within ZMap suite, or if they share the same underlying library or programming language and can be integrated programmatically. The Go programming language, for example, have found wide usage among both networking and measurement community with many production-ready tools.

Consistent measurements yield consistent data. At times, data from different sources is required, *e.g.,* DANE record from DNS to validate a server certificate. These data need to be consistent both spatially, *e.g.,* queried from the same vantage point (see above), and temporally, *i.e.,* queried within the shortest time discrepancy.

The temporal integrity among cryptographically secured data, *e.g.,* an X.509 certificate and corresponding DANE TLSA records, can be validated if they also carry a validity period. If records are revocable, respective revocation data must be regarded, for example, OCSP not only indicates if a certificate is revoked, it also contains information about revocation time and reason. If validity information are missing, we need to consider an error margin when interpreting data. Such cases have been discussed in related work, for example, in case of validating CAA records [79], [213]. Historic data, *e.g.,* from indexing services, can be uses to corroborate temporal integrity.

Non-technical aspects impact technical aspects. Web PKI is a complex sociotechnical ecosystem (see Abbildung 4.2.4). Its inspection only through the technical lens can lead to misinterpretations. Specifically, investigating user experience with respect to TLS and Web PKI is challenging. For one, while major CAs and infrastructure operators (*e.g.,* CDN and DNS) serve customers globally, each is tied to laws and regulations of the country where they are registered.

To better understand notable cases and outliers in a measurement, such non-technical aspects can be helpful. An example is the relative high share of invalid certificates in South Korean finance and banking services as deemed by researches. However, knowing that it is common practice for banking apps to sideload their own certificates on users devices (see Abbildung 4.2.3), can explain this phenomenon.

4.10 Conclusion

The TLS protocol, as a mean to establish secure communication channels over the Internet, and the Web PKI, and a common basis for TLS authentication, have been a popular target of Internet measurements. In this tutorial we first introduced a systematic approach to TLS, Web PKI, and their interplay, and used it to work out relevant aspect comprising potential data sources, measurement features and attributes, and temporal aspects of measurements. We provided an overview of tools most commonly used for TLS and Web PKI measurements, discussed how to prepare, and perform measurements, and finally demonstrated various pitfalls in interpreting measurement data.

We discovered discrepancies and even contradictory findings in related works and discussed probable causes, and how such cases can be avoided when designing, performing, and interpreting own measurements. In addition, the summary of related work of the past decade provided in this work can serve as an overview of which aspects of TLS and Web PKI measurements have previously been subject of study and how these measurements have been performed.

Appendices

We have conducted our own measurements which we use to exemplify and clarify the aspects that we discuss in Abschnitt 4.3. The following comprises a brief overview of our dataset (see Tabelle 4.A.1), and our measurement and analysis method (depicted in Abbildung 4.B.1).

4.A Dataset

We use three sets, comprising domain or institution names, as point of departure for our measurements:

TRANCO Tranco full list

EU-FINANCE List of EU *Monetary Financial Institutions*

IR-BANKS List of Iranian Banks

Each dataset has been selected due to its unique properties. The first dataset, TRANCO, comprises most popular domain names (see Abschnitt 4.4). The second dataset, EU-FINANCE, was selected under the assumption that financial institutions apply security precautions that might be reflected in their choice of certificates or TLS setup. Finally, the list of Iranian banks, IR-BANKS, represents a group of institutions under US and EU sanctions but reliant on services from those country for their online operation. Tabelle 4.A.1 summarizes these datasets.

4.B Method

We perform our measurements from a single vantage point in Germany. Before performing actual measurements, we run our initial data through two preparation pipelines. For the TRANCO dataset, we perform a reachability test and discard any domain name that doesn't resolve to an IP address or is not reachable through ports 80 or 443. In case of EU-FINANCE dataset, preparation phase maps real-world institution names to domain names. Abbildung 4.B.1 depicts our method. Note that none of the entries in IR-BANKS are reachable from outside Iran so that our toolchain is not applicable. For this set we limit ourselves to X.509 certificates that we manually collected from CT Logs.

Reachability test. Domain names in the TRANCO dataset are not necessarily delegated or resolve to an IP address (see Abschnitt 4.4). To filter out unreachable hosts, we first try to resolve the given name to an IPv4 address (using Google `8.8.8.8` recursive resolvers). $\approx$ 13% of entries do not resolve to an IP address and a timeout occurs in 1150 cases.

For all reachable hosts, we scan to see if ports 80 and 443 (HTTP and HTTPS respectively) are open. We follow HTTP (`3xx`) and HTML (`http-equiv`) redirects. For all hosts and open ports we use a custom

Tabelle 4.A.1: Datasets used in this paper

Name	Entries	Source	Access Date	Measurement Date
TRANCO	3.77 M	`tranco-list.eu`	2024-04-04	2024-04-{07,12}
EU-FINANCE	4 k	EU Central Bank	2023-06-12	2023-06-{12,14}
IR-BANKS	28	Online search	2023-06-15	2023-06-15

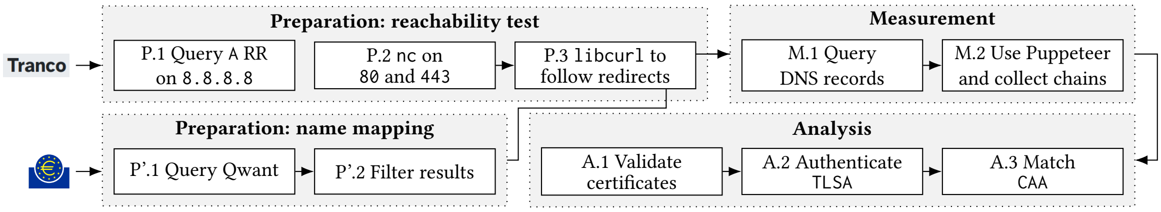


Abbildung 4.B.1: Simplified toolchain and procedure used in collection, measurement, and analysis of TRANCO and EU-FINANCE datasets

tool based on `libcurl` to contact the host over HTTP or HTTPS. In this process, we also follow HTTP and HTML redirects and record all intermediate domain names. 145*k* domains allow neither HTTP nor HTTPS connections. The remaining names resolve to IP address and host a web server.

Name mapping. For our EU-FINANCE, we first need to map real world names to domain names. For this, we use Qwant, a French search engine which provides a free and publicly accessible API. For each entry, we query the institution name and store matching results. We then generate a list of all domain names over all queries and sort it by occurrence frequency. Based on this list, we generate a blacklist of social media, encyclopedia, yellow pages, and alike. For each result page per institution we remove entries matching the blacklist and take the first remaining entry as the matching domain. Finally, the results are manually verified to remove any possible false positives. It should be noted that in contrast to entries in TRANCO dataset, we already assume reachability for the results of this phase (search bot visits).

Measurement. For the actual measurement, we feed processed domain names from TRANCO and EU-FINANCE datasets to Puppeteer (a headless browser; see Abschnitt 4.5). Puppeteer enables us to have a similar experience to users when navigating to these domains. For each domain name, we again follow redirects (including JavaScript redirects) and store X.509 certificate chains alongside HTTP headers. We also collect A, SOA, NS, TLSA, and CAA DNS records for each domain name.

Data analysis. We use the crypto library from ZMap project to parse and validate certificate chains (step **A.1**) against Mozilla Root Store from 2024-03-11[11]. To imitate browser behavior, we also manually include all non-expired and non-revoked intermediate certificates (from 2023-04-24[12]) chaining up to the root certificates described above. In cases where multiple validation paths were available, for example, due to cross certification, we used the shortest for our analysis, *e.g.,* CA market share. For DANE authentication we used the library from Shumon Huque[13]. To match CAA records with leaf certificate issuers we took list of CAA identifiers from Mozilla Common CA Database, compared them with PKI repositories of respective CAs and enhanced the list with our own data.

[11] https://curl.se/docs/caextract.html
[12] https://wiki.mozilla.org/CA/Intermediate_Certificates
[13] https://github.com/shuque/dane

4.C Survey

Tabelle 4.C.1: Brief survey of related work that has been cited in this work

Year	Authors	Goal	Data collection†‡	Measurement Features: Measurable	Measurement Features: Inferred	Frequency
2011	Holz *et al.* [111]	· Prevalence · Conformance · Security	Ⓐ TLS from DE over Alexa 1M Ⓐ TLS outside DE over Alexa 1M Ⓟ TLS research net traffic from DE Ⓐ TLS from EFF over IPv4	**T1 T2 X1 X2**	· Certificate quality · Certificate sharing · CA market share	2009-11 ϟ 2011-04 2011-04 2010-09 and 2011-04 2010-03 – 2011-06
2013	Durumeric *et al.* [69]	· Prevalence · Conformance · Security	Ⓐ X.509 over IPv4	**X1 X2 X3**	· CA ownership · Ca market share	2012-06-06 – 2013-08-04
2014	Liang *et al.* [147]	· Security	Ⓐ DNS, TLS, and HTTP over Alexa 1M	**X1 X3**	· Certificate sharing · Abandoned unrevoked certs	*undefined*
2014	Durumeric *et al.* [68]	· Security	Ⓐ TLS over Alexa 1M Ⓐ TLS over 1% of IPv4	**T1 T2 X3**	· Vulnerable devices/software · Patching behavior	2014-04 and 2014-05
			Ⓐ TLS over IPv4	**T1**	· False negative rate	2014-04-24, 2014-05-01
			Ⓟ TLS from ICSI over research net Ⓐ TLS from Uni Michigan over IPv4	**T1**	· Pre-disclosure patching · Certificate replacement[1]	2014-04 2014-04 – 2014-05
			Ⓟ TLS from LBNL net Ⓟ TLS from NERSC net Ⓟ TLS from private honeypot	**T1**	· Attack activity[1]	2012-02 – 2012-03, 2013-02 – 2013-03, 2014-01 – 2014-04 2014-02 – 2014-04 2013-11 – 2014-04
2014	Zhang *et al.* [265]	· Security	Ⓟ X.509 from Rapid7 over IPv4 Ⓐ CRL endpoints of collected X.509 Ⓐ TLS over Alexa 1M	**T1 X1 X2 X3**	· Certificate replacement · Time diff revocation and replacement	2013-10-30 – 2014-04-28 2014-05-06 *undefined*
			Ⓐ TLS over Alexa 1M (from [68]) Ⓐ TLS over Alexa 1M	**T1**	· False negative rate	2014-04-09 2014-04-28
2015	Delignat-Lavaud and Bhargavan [54]	· Security	Ⓐ X.509 from EFF and Alexa 1M (from [2], [53])	**X2**	· Certificate sharing · TLS cache sharing	2013-07-31
			Ⓐ HTTPS over Alexa 10k	**T3**	· Cross-protocol redirection	*undefined*
2015	Zhu *et al.* [491]	· Prevalence · Security	Ⓐ TLS over ca 1k domains w/ TLSA	**X4**	· `TLSA` RR size and IP fragmentation	2014-07-14 ϟ 2014-12-03
2015	Liu *et al.* [149]	· Prevalence · Security · Performance	Ⓟ X.509 from Rapid7 over IPv4 Ⓐ CRL endpoints of collected X.509 Ⓐ OCSP endpoints of collected X.509	**X1 X3**	· CA behavior on revocation · CRL sizes and IP fragmentation	2013-10-31 – 2015-03-30 2014-10-02 – 2015-03-31 2015-03-31
			Ⓐ TLS from Uni Michigan over IPv4	**T1 X3**	· Stapled OCSP support	2015-03-28
2016	Chung *et al.* [42]	· Security · Privacy	Ⓐ X.509 from Uni Michigan over IPv4 Ⓟ X.509 from Rapid7 over IPv4	**X1 X3**	· CA and Device diversity · Certificate to device map	2012-06-10 ϟ 2014-01-29 2013-10-30–2015-03-30
2016	Holz *et al.* [110]	· Security	Ⓐ TLS over IPv4 Ⓟ TLS from UC Berkley net	**T3 X1 X3**	· Certificate sharing	2015-06-09 ϟ 2015-08-04 2015-07-29–2015-08-06
2016	Springall, Durumeric, and Halderman [228]	· Prevalence · Security	Ⓐ TLS over Alexa 1M	**T1**	· TLS session vulnerability window	2014-07-14 ϟ 2014-12-03
2016	Cangialosi *et al.* [33]	· Prevalence · Security	Ⓟ X.509 from Rapid7 over IPv4	**X2**	· Domain ownership · Certificate sharing	2013-10-30–2015-03-30
			Ⓐ X.509 from Zhang [265] (see above)	**X2 X3**	· Revocation of self-hosted vs managed certs · Certificate quality	*undefined*
2016	VanderSloot *et al.* [247]	· Prevalence	Ⓐ X.509 from Censys Ⓐ TLS over IPv4 Ⓐ TLS over Alexa 1M Ⓐ X.509 over CT logs Ⓐ X.509 over domains in CT logs Ⓐ TLS over `.com`, `.net`, `.org` Ⓐ TLS over domains in Common Crawl Ⓟ X.509 ICSI over research net	**X1**	· Cert coverage of various sources	2016-08-29–2016-09-08
2017	Ouvrier *et al.* [189]	· Prevalence	Ⓟ TLS from Uni Calgary net	**T1 T2 T3 X2**	· CA market share	2015-10-11–2015-10-17

2017	Amann *et al.* [10]	· Prevalence · Conformance · Security	Ⓐ TLS over 193M domain names (from TUM and Uni Sydney)	**T1 T3 X2**	· Prevalence of SCT · Usage of HSTS and HPKP	*undefined*
			Ⓟ TLS from UC Berkley net	**T1 T3 X2**	· Prevalence of SCT	*undefined*
			Ⓟ TLS from TUM net Ⓟ TLS from Uni Sydney	**T1 X1 X2 X3**	· [Validating measurement above]	*undefined*
			Ⓟ TLS from ICSI over research net	**T1**	· TLS version evolution	2012-04-01–2017-09-01
2018	Scheitle *et al.* [213]	· Conformance · Prevalence	Ⓟ TLS from UC Berkley net Ⓐ TLS over 423M domain names	**T1 X1 X2 X3**	· Prevalence of SCT · Info leakage over CT logs	2017-04-26–2018-05-23 2018-05-18
2018	Sy *et al.* [233]	· Prevalence · Security	Ⓐ TLS over Alexa 1M	**T1 T2**	· Session sharing	2018-04–2018-05
2018	Kotzias *et al.* [128]	· Prevalence · Security	Ⓟ TLS from ICSI SSL Notary Ⓟ TLS over IPv4 and Alexa 1M (Censys)	**T1**	· TLS fingerprinting · Vulnerability analysis	2012-02–2018-03 2015-08–2018-05
2018	Chung *et al.* [43]	· Prevalence	Ⓟ X.509 from Censys Ⓟ TLS over Alexa 1M (Censys)	**T1 X1 X3**	· OCSP responder availability	*undefined*
2019	Oakes *et al.* [184]	· Prevalence	Ⓟ X.509 from 2M residential clients	**X1 X2**	· CA market share	2017-07-12–2018-01-12
2019	Aas *et al.* [1]	· Prevalence	Ⓐ X.509 from Censys Ⓟ X.509 from CT logs Ⓐ TLS over names from CT logs Ⓟ Firefox client telemetry	**T1 X1**	· Prevalence among top sites · CA market share	*undefined*
2019	Holz *et al.* [365]	· Prevalence	Ⓐ TLS over Alexa 1M + `com`/ `net`/`org`	**T1**	· Prevalence among TLDs and hosting services	2019-05-{1,3,4,5}
			Ⓟ TLS from ICSI SSL Notary Ⓟ TLS from an Australian uni net	**T1** **T1**	· TLS version prevalence	2012-03-01–2019-09-01 2019-05-09–2019-05-13
			Ⓟ TLS from Lumen Privacy Monitor	**T1**	· Client/Server support	2016-12-01–2019-04-01
2020	Lee *et al.* [140]	· Prevalence · Conformance · Security	Ⓐ TLS over DANE-enabled servers	**T1 T3 X1 X3**	· DANE management trends	2019-07-11–2019-10-31
2020	Zhang *et al.* [266]	· Prevalence · Security	Ⓐ TLS over domains from CT logs	**T3 X2**	· Context confusion vulnerability	*undefined*
2020	Wan *et al.* [252]	· Prevalence	Ⓐ TLS over IPv4	**T1**	–	2019-10 ϟ 2019-12
2020	Raman *et al.* [197]	· Security	Ⓐ TLS to vantage points in Kazakhstan Ⓐ TLS to web servers in Kazakhstan	**T1 X1 X2**	· Interception triggers · Interception detection	2019-07-20 2019-07-{22,23}
2021	Lee *et al.* [142]	· Security	Ⓐ TLS over names from Rapid7	**T1 T2**	· Location-dependent security	*undefined*
2021	Brinkmann *et al.* [31]	· Security	Ⓐ TLS over IPv4	**T1**	· Vulnerability to Cross-Protocol Attacks	*undefined*
2021	Tatang, Flume, and Holz [469]	· Prevalence · Security	Ⓐ TLS over Tranco 1M	**T1 X1**	· MTA-STS prevalence	2019-06-26 ϟ 2019-10-13
2021	Zhang *et al.* [269]	· Prevalence · Security	Ⓟ TLS over *360 Secure Browser*	**X1**	· Evaluation of *hidden roots*	2020-02-01–2020-06-30
2021	Ma *et al.* [154]	· Prevalence	Ⓟ X.509 from CT logs	**X1 X2**	· CA operators landscape	2020-07-01
2022	Nawrocki *et al.* [178]	· Prevalence · Security	Ⓐ TLS over Tranco 1M (TCP) Ⓐ TLS over Tranco 1M (QUIC)	**T3 X1 X2**	· Impact of X.509 certs on QUIC	*undefined*
2022	Lee *et al.* [139]	· Prevalence · Conformance · Security	Ⓐ TLS over `com` / `net` / `org` / `se`	**X1 X2 X4**	· Deployment quality · Key rollover security	2019-07-13–2021-02-12
2023	Farhan, and Chung [73]	· Prevalence	Ⓟ X.509 over IPv4 from Rapid7 Ⓟ X.509 from CT logs	**X1**	· CA market share evolution	2013-09–2021-12 2013–2021-02
2023	Ma *et al.* [153]	· Security	Ⓟ X.509 from CT logs	**X1 X2 X3 X4**	· Third-party key access	N/A
2024	Tehrani *et al.* [86]	· Prevalence · Conformance · Security	Ⓐ TLS over Tranco full Ⓟ X.509 from CT logs	**X1 X2 X4**	· Interlink of CAA, DANE, and CT logs	2024-04-{07,12} 2024-04-24

† Ⓐ: active measurement, Ⓟ: passive measurement † includes only TLS and X.509 datasets ϟ Irregular measurement

1 In combination with active and ICSI scans of the same study

Kapitel 5

Do CAA, CT, and DANE Interlink in Certificate Deployments? A Web PKI Measurement Study

Abstract

Integrity and trust on the web build on X.509 certificates. Misuse or misissuance of these certificates threaten the Web PKI security model, which led to the development of several guarding techniques. In this paper, we study the DNS/DNSSEC records CAA and TLSA as well as CT logs from the perspective of the certificates in use. Our measurements comprise 4 million popular domains, for which we explore the existence and consistency of the different extensions. Our findings indicate that CAA is almost exclusively deployed in the absence of DNSSEC, while DNSSEC protected service names tend to not use the DNS for guarding certificates. Even though mainly deployed in a formally correct way, CAA CA-strings tend to not selectively separate CAs, and numerous domains hold certificates beyond the CAA semantic. TLSA records are repeatedly poorly maintained and occasionally occur without DNSSEC.

5.1 Introduction

Secure and authenticated transport is essential to the modern web. Trust in the Web PKI is built on X.509 certificates and derives from accepted root certification authorities (CA). Any CA, which is part of a valid trust chain, can issue a trustworthy certificate for any service. Millions of devices rely on this ecosystem for browsing, shopping, online banking, *etc.* Given the immense reach of each CA, much effort has gone into securing the Web PKI (Abbildung 5.1.1)—most prominently DNS-based Authentication of Named Entities (DANE) [363], Certification Authority Authorization (CAA) [356], and Certificate Transparency (CT) logs [388].

DANE and CAA build on the Domain Name System (DNS), which allows domain owners to store accessible information. *DNSSEC* adds an intrinsic chain of trust along the DNS hierarchy. Different from the certificate chain-of-trust, only the entity that controls a domain can sign its records. Using DANE TLSA records, the DNS can provide additional information to verify X.509 certificates and even establish trust without relying on a CA. In contrast, *CAA* records allow domain owners to restrict which CAs are allowed to issue certificates for their domains. Independent of the DNS, *CT logs* are

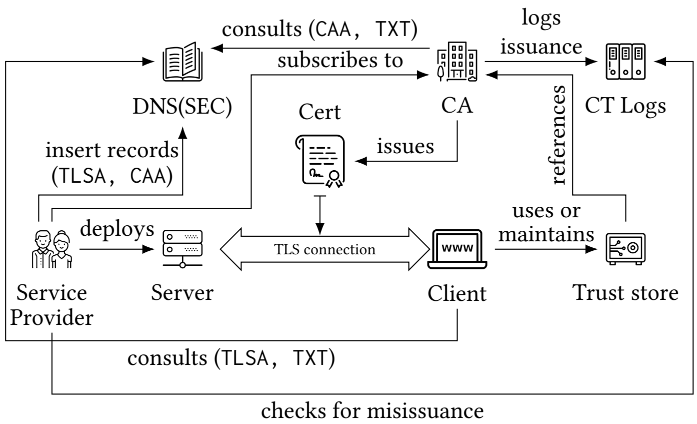


Abbildung 5.1.1: Overview of Web PKI entities and their relation to DNS(SEC).

append-only databases that collect published certificate and make misissued certificates visible to the public.

All three standards—DANE, CAA, and CT logs—concern X.509 service certificates but use different methods, different publication channels, and are targeted at different audiences. In this work, we want to learn about the use (and misuse) of these concurrent approaches to harden the Web PKI ecosystem. In a large measurement study, we collect data from 4M domains based on the Tranco top list, spanning DNS records, X.509 certificates, and CT log entries. We compare and analyze the different records in use with respect to their existence, content, and intended semantics, *i.e.,* correctness, consistency, and coherence. Our major findings read:

1. Nearly 9% (357k) of all domains deploy at least one of either DNSSEC, DANE, or CAA; overlapping deployment is much smaller.

2. CAA records are largely deployed correctly and consistent with certificate issuers (> 90%). Nevertheless, CA strings are not precisely defined; several strings match more than a single CA, one matches 21 CAs.

3. CAA records are mainly deployed without DNSSEC; even some TLSA records lack DNSSEC protection.

4. TLSA records are rare and often poorly maintained, some of which correspond to certificates that have been revoked or have expired since long.

The remainder of this paper is structured as follows. Abschnitt 5.2 introduces the Web PKI along with the relevant technologies for securing certificate deployments and related work. Our measurement method, data collection and its processing are explained in Abschnitt 5.3. Abschnitt 5.4 reports on the deployment of the DNS extension records and examine their correctness with respect to the actual certificates in use. Abschnitt 5.5 combines information from DNS records, certificates, and logs to

Tabelle 5.1.1: CT Logs, DANE, and CAA offer three different approaches to securing the Web PKI.

Technology	Core Idea	Infrastructure	Responsible	DNSSEC	Target Audience
CT Logs [388], [389]	Auditable certificate issuance	CT Logs	CA	n/a	Subscribers (Domain Owners)
DANE [363]	Bind public keys to names	DNS	Domain Owner	mandatory	Relying Party (Clients)
CAA [357]	Constraint issuer of domain names	DNS	Domain Owner	not mandatory	CA (Certificate Issuer)

systematically explore consistency and coherence of the security information set. We conclude in Abschnitt 5.7 with an outlook on future research.

5.2 Background and Related Work

The Web PKI ecosystem is the foundation for authentication on the Web using X.509 certificates at its core, *e.g.*, [84]. Since its inception, various extensions have been introduced to enhance its functionality or address its shortcomings. In this section, we briefly introduce Web PKI and discuss how DNS CAA, DANE, and CT Logs address one of the biggest challenges: unconstrained and global certification authority. Furthermore, we present prior work.

5.2.1 Background

A fundamental security challenge on the Internet is the trust in cryptographic keys used for authentication. In a controlled environment, keys can be manually attributed to specific entities, but such an approach is less applicable in large-scale distributed systems such as the Internet. We now discuss how Web PKI addresses this challenge, which shortcomings still exist, and which remedies have been proposed.

Web PKI. Web PKI introduces *Certification Authorities* (CA) to bind public keys to domain names (among other attributes) to form a certificate. The authenticity of a certificate can be verified using its cryptographic signature. In public key cryptography, a signature is generated by a private key (which is kept secret) and can be validated by the corresponding public key (which is published openly). A relying party (RP), *i.e.,* a piece of software that decides whether a certificate is valid or not, would then trust a certificate that is signed by or can be traced back to a trusted CA. On the Web, an RP is typically a browser, which maintains its own set of trusted CAs or Trust Anchors (TA) in a local *trust store.*

Issuing a certificate correctly is the most important task of a CA. Unfortunately, CAs are not restricted when issuing certificates. They can create certificates for any name. A compromised or malicious CA, then poses security risks for all entities that rely on security assurances provided by the Web PKI. Those incidents occur in practice, such as the DigiNotar incident [10]. To counter this threat, various solutions have been introduced, which we summarize in Tabelle 5.1.1 and discuss in the following.

Certification Authority Authorization (CAA). A CAA record [356] gives a domain name owner the ability to restrict issuance of certificates by defining which CAs are allowed to issue certificates for its name. Such a *constraint* is stored in the DNS using dedicated CAA resource records to describe restrictions for wildcard or fully qualified domain names (FQDN) for the namespace under control of the name owner.

A CAA record is composed of a flag, a tag, and a value. Issuance constraints are defined by `issue` and `issuewild` tags. The latter only constrains wildcard certificates while `issue` records concern both, wildcards and FQDNs, but are superseded by an `issuewild` record. These two tags have a well-defined syntax. When the syntax is violated, a certificate should not be issued. To forbid issuance explicitly, an empty value (`";"`) can be used.

CAA records allow learning details about the CA itself. DigiCert, for example, accepts `digicert.com` as well as `amazon.com` (among others) according to its Certification Practices Statement (CPS) [320].

CAA also enables CAs to report policy violations to the name owner based on information configured in the `iodef` tag. The value of this tag can be an email address or a URL. An example of a policy violation is when a certification request is submitted at a CA that does not satisfy the issuance constraints.

DNS-Based Authentication of Named Entities. DANE [363] allows the attestation that a public key is a valid key for a domain name. To enable this attestation, a domain name owner stores the public key in specific records of the name under attestation. For TLS-based services, DNS TLSA records signal RPs which (type) of certificate to expect from the server. This can be an end entity (EE) certificate (*e.g.,* a leaf certificate) or a TA certificate (*e.g.,* from a CA). A TLSA record can reference a certificate or its subject public key information (SPKI). The reference is either to the full raw data (*e.g.,* hex formatted certificate) or its digest (*e.g.,* SHA256 hash).

DANE makes use of DNS Security Extensions (DNSSEC) which bring authentication and integrity assurance to the DNS [444] and mitigate common DNS attacks such as cache poisoning. DNS records are signed in DNSSEC, have limited validity, and must be signed again after expiration. Without DNSSEC, DNS records are susceptible to spoofing and manipulation, thus defeating the purpose of DANE.

Certificate Transparency Logs. CT logs [388], [389] were introduced to enable public monitoring and auditing of issued certificates. Although logging is not mandatory [307], major browsers such as Chrome and Safari only accept certificates that are logged in at least two compliant CT logs. Due to their market shares, this forces CAs to comply or lose out on large shares of customers.

Target Audiences. Each technology follows a different core idea and is targeted at different parties, see Tabelle 5.1.1. Whereas CAA records are meant for CAs *before* a certificate is issued, CAs feed CT logs *after* issuance to make certificates visible to domain owners and thus allow verification. DANE is deployed by domain owners as well, but faced towards certificate consumers, *i.e.,* relying parties, signaling which (type of) certificate to expect or accept.

5.2.2 Related Work

Efforts to secure the Web PKI have been ongoing for more than 10 years. CAA records were standardized in 2013 [356] and updated in 2019 [357]. DANE became a standard in 2012 [363], and CT logs were standardized in 2013 [388] and updated to version 2 in 2021 [389]. Since 2017 the CA/B forum requests that CAs validate CAA records. To the best of our knowledge, this paper is the first study that comprehensively analyzes all these technologies together to better understand configurations in real deployments, including inconsistencies.

CAA. In 2018, Scheitle *et al.* [213] presented the first analysis of the CAA ecosystem, actively measuring the behavior of selected CAAs as well as auditing the ecosystem. They found that 3k of 95k domains in the Alex Top 1M list deployed CAA records. At that time, most domain owners (89%) configured a single CAA string, mainly letsencrypt.org (64%), and did not allow issuing certificates for arbitrary names of a domain or subdomain (59%). CA strings that occurred infrequent included many invalid strings due to misspellings or owners using their own domain as CA string—indicating a lack of automation and understanding of how to configure CAA. In a study focusing on nonfederal governments in the United States, Gebhard *et al.* [91] revealed that CAA is least deployed and grows slowly compared to the adoption of DNS records that strengthen identifying web servers (SPF, DMARC).

Certificate Transparency. In 2018, the same year Chrome made CT mandatory, Scheitle *et al.* [214] measured the adoption of CT and found an exponential increase of CT log entries. They further found two more use cases of CT logs. (*i*) CT logs can be used to identify phishing domains, (*ii*) malicious actors monitor CT logs to search for new targets. The latter was examined in a broader study in 2021 [194], which confirmed the continued use of CT logs for target discovery. In a longitudinal study of TLS certificates gathered from active scanning and CT logs, Farhan *et al.* [73] found an improvement in share of valid certificates and key strengths, but also observe a centralization in the Web PKI. 80% of valid certificates are now signed by only 10 keys.

DANE. Three years after DANE was published, Zhu *et al.* [491] found in 2015 that less than 1000 domains use DANE. In 2020, DANE still did not gain widespread adoption in browsers. While the email ecosystem saw a comparatively higher adoption rate, mismatches between TLSA records and certificates and incorrect DNSSEC were still frequent [140]. Lee *et al.* [139] observe that 94% of SMTP servers still rely on the certificates issued by CAs when they deploy DANE.

All the prior studies provide an in-depth understanding of CAA, CT, or DANE deployments. They focus, however, on each protection mechanism separately. In this paper, we first provide an update of recent deployments and then close a gap by taking a comprehensive view on the deployments of all three technologies together.

5.3 Method and Data Corpus

We collect a data corpus of 4M domains to examine the Web PKI ecosystem[1] using the processing pipeline in Abbildung 5.3.1.

[1]Source code, raw data, and our analysis are available under `https://doi.org/10.5281/zenodo.11081271`.

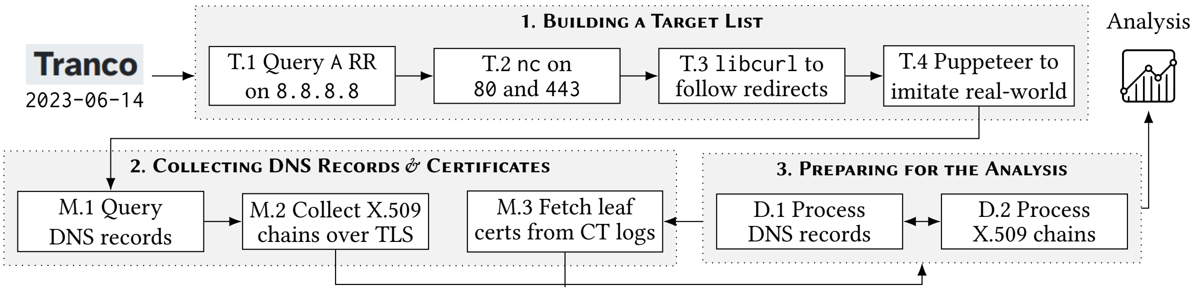


Abbildung 5.3.1: Simplified illustration of our toolchain for building a target list, collecting a dataset, and preparing data for analysis.

5.3.1 Building a Target List

Query DNS (T.1). As input, we take the Tranco top list [137] comprised of over 4.1M domain names ranked by popularity. We try to resolve each name to an A record and only keep names with valid records. $\approx 533k$ entries did not resolve to an IP address while 1150 timed out, likely due to trimmed domain names or dynamic DNS changes.

Check Ports (T.2) & Transport (T.3). Next, we check ports 80 and 443 (TCP using `netcat`) and establish an HTTP or HTTPS connection over open ports with our own tool based on `libcurl`. $145k$ domains allow neither HTTP nor HTTPS connections. The remaining names resolve to IP address and host a web server.

Browser (T.4). We then feed collected names into Puppeteer (T.4), a headless Chrome browser, and follow further redirects (including JavaScript). Our target list now contains $4M$ unique domain names (including intermediates).

5.3.2 Collecting DNS Records & Certificates

CAA, DANE, and DNSSEC from DNS (M.1). We collect the following DNS resource records (RR) for each domain: SOA, A, CAA, TLSA, as well as *contactemail* and *contactphone* TXT records, see Appendix 5.C. Additionally, we query CAA records for all parents of a given name by recursively removing the leftmost label up to the TLD, *e.g.*, for `www.example.co.uk` we query CAA records for the domain set {`www.example.co.uk, example.com, co.uk`}. Queries set the `DO` flag to request DNSSEC records (if any) and have DNSSEC validated by the resolver. For DANE, we only query TLSA records that are associated with TCP services on port 443 (HTTPS).

We use Google recursive resolvers for all DNS queries due to their availability, reliability, and provision of a JSON API.

X.509 Certificates via TLS (M.2). At this point, we do not validate certificates and we suppress TLS errors due to insecure cipher suites (OpenSSL security *Level 0*). By disregarding failures in TLS deployments, we can collect all available certificates to better understand shortcomings.

Leaf Certs from CT Logs (M.3). Domain owners might have applied for more certificates than the one deployed on their web server. For example, Cloudflare issues *backup certificates* for its customers to be able to swiftly replace keys in case of compromise [384]. We collect certificates appended to CT logs for the subset of domain names that have either CAA or TLSA records, using the open database provided by Sectigo under `crt.sh`.

5.3.3 Preparing for the Analysis

Parsing and Validating CAA (D.1). We parse CAA `issue` and `issuewild` values according to the Augmented Backus-Naur Form (ABNF) and consider malformed entries as empty (semantically equivalent to `";"`). Malformed entries forbid the issuance of a certificate [357]. For records with `iodef` tags, we verify that the value is a URL with the correct scheme (`mailto`, `http`, or `https`).

For records with the `issue` and `issuewild` tags, we devise an algorithm that matches a set of given CAA RRs to certification authorities, visualized in Appendix 5.D. It uses the following classifications:

1. **No CAA**: No applicable CAA RR was found.
2. **Implicit Match**: No CAA RR constraints issuance.
3. **Issuer Match**: At least one CAA RR matches cert issuer.
4. **Issuer Mismatch**: No CAA RR matches the issuer.
5. **Malf. Mismatch**: All CAA RR are malformed.
6. **Empty Mismatch**: Only empty (`";"`) CAA RR.

This algorithm relies on a mapping from CAA issuer domain names (*i.e.*, values in `issue` and `issuewild`) to CA certificate properties. Our mapping is based on the "List of CAA Identifiers" in the Common CA Database [471]. We enrich this list manually (*i*) based on in Certification Practice Statements (CPS) documented identifiers, and (*ii*) our own observations of undocumented string identifiers. The CPS is usually linked in the certificate. If this link is not valid, we manually identify the respective CA and browse its website to find the statement.

In contrast to X.509 certificates or DNSSEC signatures, CAA RRs do not carry a validity timestamp and are in practice only validated by the CA at the time of issuance. This can cause a discrepancy between our observations at the time of measurement and what CAs observed when they issued a certificate, and thus leads to misclassifications. To verify whether our measurement setup introduces

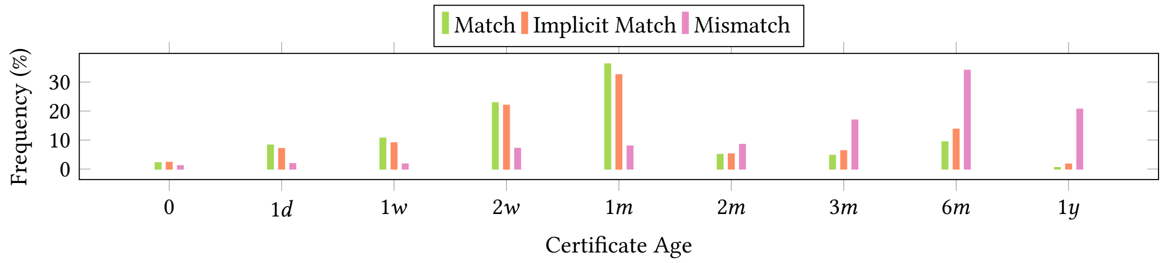


Abbildung 5.3.2: CAA matching states by age of certificates with matching subject name at the time of measurement. 75% of certificates are younger than 3 months. CAA mismatches appear relatively more often in certificates older than 3 months.

such misclassifications, we calculate the difference between our probe time and `not before` timestamp carried in certificates for domains with CAA records, see Abbildung 5.3.2. Regardless of CAA matching state, our measurements occurred in 75% within three months of the issuance. Mismatches, however, occur much more frequently for certificates that are older. This indicates that our setup is not prone to mismatches that are actually valid.

Parsing and Validating DANE (D.1). We validate that TLSA records are RFC-conformant using an open-source library [367] but leave the verification of the DNSSEC integrity to the recursive resolver.

Parsing and Validating X.509 Certificates (D.2). We parse X.509 certificates with ZCrypto [487] and check three things. (*i*) The subject (alternative) names in the certificate should match the queried domain name, *i.e.,* the name is included as a SAN or covered by a wildcard SAN. (*ii*) The certificate chain should be valid. We select the Mozilla Common CA Database [471] as our trust store. And (*iii*) Attached SCTs (if any) should be valid, *i.e.,* the SCT corresponds to the respective certificate and is signed by a trustworthy log. We use Google's library [354] to verify signatures with keys extracted from the list of all complying [353] logs [355]. Here, we assume that log operators behave correctly and omit to check the logs directly.

5.4 Configuration of X.509 Certificates, DNS CAA, and DANE Records

In this section, we focus on the deployment of X.509 certificates alongside CAA and DANE records within the DNS to better understand to which extent name owners care about correct configuration of each security extension. In total, we observe 357k unique domains (8.85% of all names in the Tranco list) that deploy at least CAA, DANE, or DNSSEC. We visualize the overlap of support for different technologies per name as an UpSet plot [145] in Abbildung 5.4.1. The bar plot on the left shows how many domains fall into each category while the bar plot on top shows the size of exclusive intersections between the categories marked in the matrix below. It is clearly visible that domain name owners do not focus on comprehensive security support.

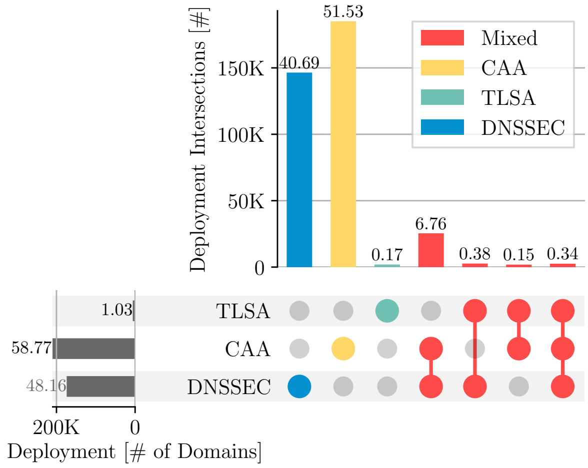


Abbildung 5.4.1: The UpSet plot contrasts the use of DNSSEC, CAA, and TLSA among domains that deploy at least one of them. CAA is most deployed (≈52%) followed by DNSSEC (≈41%) but they only co-occur in <7% of domains. DANE (TLSA) is rarely deployed (≈1%).

5.4.1 X.509 Certificates

Roughly 97% of collected certificates are valid (based on our trust store, see Unterabschnitt 5.3.3). Validation failures stem from age (47k), untrusted signees (≈ 15k), and malformed certificates (60k). All valid certificates were submitted to either 2 (72%), 3 (23%), or 4 (5%) CT logs. With 73%, most certs were submitted to logs operated by Google, two thirds were submitted to logs operated by Cloudflare, and about half to logs operated by DigiCert.

Two out of every three certificates were issued by one of four major CAs: Let's Encrypt with a total share of 52%, followed by Google Trust services (16%), DigiCert (4.5%), and Sectigo (4.5%). This does not include resellers, *e.g.*, ZeroSSL using Sectigo infrastructure.

In our observation, 7.2% of unique domains point to hosts that provide certificates with mismatching subjects, *i.e.*, not matching the original domain name. These are in part default (self-signed) server certificates or service provider certificates for parked domains.

5.4.2 CAA Deployment

5.2% of the 4M domains we scanned support CAA records. Most (4.55%) domains only specify constraints for CAs for fully qualified domain name or wildcard names (respectively `issue` or `issuewild` records). Around 0.01% only provide information on reporting policy violations (`iodef` records). 0.63% of domains deploy both.

Reporting Policy Violations—`iodef` Records. Several records that have been configured prevent contact because of misconfiguration. About 3.79% of `iodef` entries are invalid due to an invalid schema [357] as neither `mailto` nor `http[s]` is present. 17 entries with unknown schemas still contain a colon, eight of these use a non-existent schema (`mailinto:`, `mail:`, etc.), six are typos with missing or

Tabelle 5.4.1: The CA strings that appear in more than 10% of the CAA records in our dataset as well as their relative occurrence as the *only* CA string.

	issue		issuewild		
CAA String	Overall	Single	Overall	Single	CA Count†
`letsencrypt.org`	86.95%	17.70%	85.28%	6.43%	1
`digicert.com`	62.58%	1.31%	77.94%	1.36%	4
`comodoca.com`	45.68%	0.39%	73.73%	0.34%	12
`pki.goog`	33.27%	0.91%	51.10%	0.06%	1
`globalsign.com`	32.01%	0.55%	31.13%	0.63%	2
`sectigo.com`	21.52%	1.04%	25.37%	1.79%	21

† Count of unique CAs (by Subject Organization) in our dataset that match the CAA string in the first column

switched letters, and three contain characters that break the formatting. Among the remaining invalid entries, 924 are likely email addresses and 58 are HTTP endpoints. One record only contains a sequence of 27 numbers, which is unlikely a phone number because it is too long. Among valid records, nearly all `iodef` records contain email addresses (> 99%), 42 in total an HTTPS URL, and 16 domains have both.

We contacted domain owners with invalid iodef records. Since many email addresses were associated with multiple domains, the 924 entries could be reduced to 504 distinct addresses. Nearly 16% of the mails could not be delivered, mostly because the mailboxes no longer exist. While most did not respond, we also got kind responses and learned that at least one case was the result of false documentation of a hosting provider. Others mentioned that their DNS settings were applied automatically by the service provider.

Granting Authorization to Issue (Wildcard) Domain Names — `issue` and `issuewild` Records. Among ≈210k domains with issuance constraints, 97% have records with an `issue` tag and 57% have records with an `issuewild` tag. The overlap is 54%, which leaves 43% that only have `issue` and 3% that only have `issuewild`. 38 domains have malformed entries.

Our dataset shows that compared to six years ago, name owners are more liberal when it comes to allowing multiple CAs to issue certificates. In 2018, 89% of domains only allowed a single CA to issue certificates [213]. Now, in 2024, 28.39% domain owners allow four different CAs to certificate their names, 16.35% 3 CAs, 16.26% 5 CAs, and 10.04% 2 CAs. We even observe one domain with 59 CAA `issue` records and another one with 46 records. Another 59 domains, all ending in `.ba.gov.br`, each have 17 CAA records.

Tabelle 5.4.1 shows the most commonly used CAA strings as well as the number of unique CAs (by Subject Organization) that match this string in our dataset. Clearly, a domain owner grants permission to the operator of the CA infrastructure rather than to a specific CA. For example, ZeroSSL, an Austrian CA, accepts `sectigo.com`, because it uses Sectigo Infrastructure to issue certificates under its own

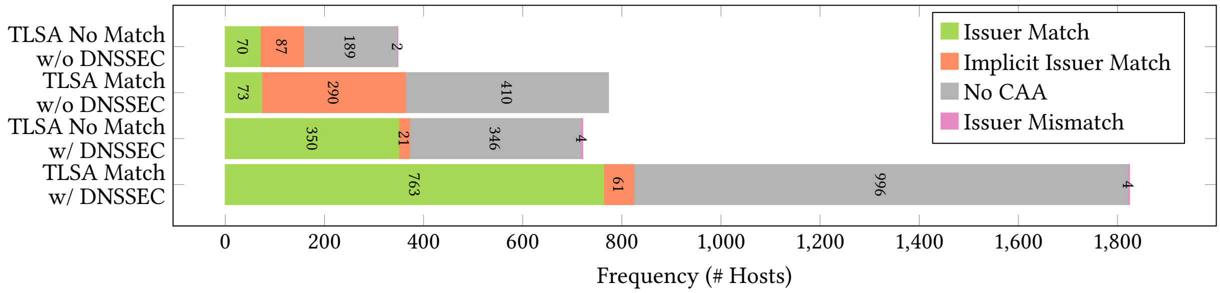


Abbildung 5.4.2: Total number of hosts delivering valid certificates with valid TLSA and CAA records divided by DNSSEC support, and TLSA and (selected) CAA matching status.

brand. At the same time, a CA might accept different CAA strings. In our example, ZeroSSL also accepts `usertrust.com`, another brand of Sectigo.

Records with Non-Standard Tags. We find 353 CAA records with tags not defined in RFC [357]. These tags can be categorized in three types: (*i*) 252 × unrecognized tags as defined in CA/B Baseline Requirements [307, A.1.1.], (*ii*) 50× misspellings such as an extra letter, and (*iii*) 51× malformed formats such as extra quotes.

5.4.3 DANE Deployments

About 0.1% of all unique names that we queried are standard compliant. 98% of these 3678 names provide a valid certificate. The majority (87%) define only end-entity constraints (DANE or PKIX EE), 10% only trust anchor constraints, and the rest both. Although DNSSEC is a requirement for DANE, one out of every third TLSA record set is delivered over 1126 insecure DNS sets. Abbildung 5.4.2 summarizes our findings.

It is noteworthy that only 70% of names match their TLSA records as described below. The case of mismatching TLSA records is discussed in depth in Abschnitt 5.5 using data from CT logs.

Matching TLSA with Invalid Certificates. About 1.5% (53 names) define matching DANE-TA or DANE-EE constraints with invalid certificates (see Abschnitt 5.3). More than half are expired and issued by Let's Encrypt and the remaining rest are self-signed certificates or leaf certificates by miscellaneous (partly not accredited) CAs. More than 71% of TLSA record sets here are secured by DNSSEC and 73% have no associated CAA records.

Matching TLSA with Valid Certificates. 2553 entries provide TLSA records that match the provided certificate and the certificate is valid. However, not all are secured by DNSSEC and only about 46% of entries provides CAA records. Only 190 names in this set have TLSA records that impose a limit on Web PKI certificates, while the rest are DANE-EE and DANE-TA constraints.

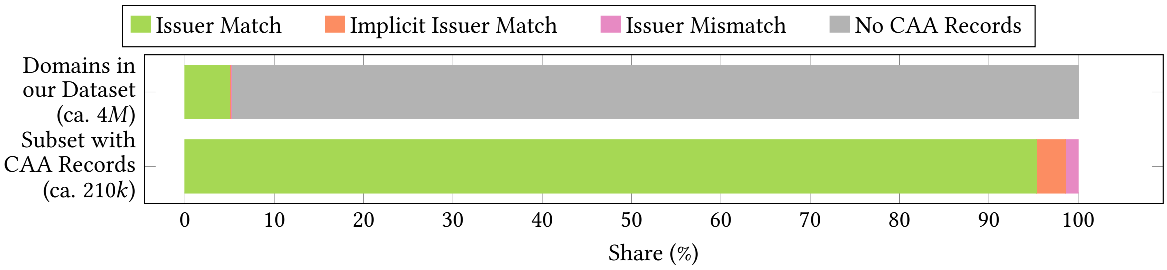


Abbildung 5.5.1: Most certificates with CAA deployment are consistent with their CAA records. We only observed an "Issuer Mismatch" in 2237 domains (1.07% of CAA records).

5.5 Information Consistency

5.5.1 CAA Records vs X.509 Certificates

Based on the results of our CAA matching algorithm, described in Unterabschnitt 5.3.3, we examine the consistency between CAA records and the certificate issuers of the TLS certificates. While the RFC [357] mentions the role of an auditor ("Certificate Evaluators"), CAA records provide information for CAs at the time of certificate issuance and are not required to stay consistent.

CAA Validation Overview. Abbildung 5.5.1 shows the result of our classification for two datasets based on CAA records with the `issue` tag. The upper bar contains all $4M$ domains in our dataset while the lower bar only considers the subset with CAA records. $3.8M$ domains (94.8%) do not have CAA records.

$200k$ domains (4.96%) have certificates consistent with their CAA records ("Issuer Match"). These are 95.34% of the nearly $210k$ domains with a CAA record. Another 0.17% (3.22%) is classified as "Implicit Issuer Match". These domains mostly (93%) have CAA records with the `issuewild` tag but not with an `issue` tag while containing FQDN in their certificate, *i.e.,* the domain has the relevant resource records but only restricts the issuance for wildcard certificates. 6% only deploy `iodef` records. < 1% only have records with unknown tags.

Nearly 91% of domains with CAA records deploy the relevant RR, *i.e.,* the CAA record(s), themselves. For roughly 9% the direct parent domain has the relevant RR. For about 200 domains (≤1%) the DNS hierarchy needs to be traversed further, up to 4 times, which occurred only once.

Only 0.06% of all domains (1.07% with CAA records) have an "Issuer Mismatch", *i.e.,* the string in their CAA records does not match the issuer of their certificate. We found two domains that deployed *only* malformed CAA records.

Relevant CAA Records. For most domains (92.4%) a CAA record with the `issue` tag was the deciding record, *i.e.,* the CAA record that fit the domain we requested from the web server. In 95% of cases the issuer matches, 3.5% have an implicit match, and the remaining 1.5% mismatch. In cases where the

CAA record with the `issuewild` tag was relevant (7.6%), the share of domains with a matching issuer is even higher with 99.4%. 0.6% mismatch, and we do not observe any implicit matches.

Wildcard Certificates for Subdomains. Next, we examine domains with `issuewild` CAA records and a wildcard name for their *subdomains* in the certificate. This is the case for 21% of domains with CAA records in our dataset (44*k*). For 99.5% domains the issuer of the certificate is consistent with their CAA `issuewild` records. 216 (0.49%) have a mismatch, *i.e.,* their certificates should not have been issued in this configuration. 17 (0.04%) forbid issuance of a wildcard certificate with an empty CA string (but have an active wildcard certificate).

Partial CAA Matches. A CAA `issue` record covers a FQDN and a wildcard domain in the absence of an CAA `issuewild` record. However, a CAA `issuewild` record without an `issue` record would only restrict wildcard issuance. If a certificate lists both FQDN and a wildcard domain, CAs should check for CAA `issue` and `issuewild` records. A *partial* match occurs when the issuer of a certificate with a FQDN and a wildcard matches either an `issue` or `issuewild` record, but not both.

For 60 domains, we can match the certificate issuer to the CAA `issue` record but observe a mismatch for the wildcard in the same certificate. And for the reverse—the certificate issuer matches a CAA `issuewild` record, but not the `issue` record—we find 151 occurrences. In no case is issuance restricted by an empty (';') record.

It seems unlikely that domains, which change their CAA records in the time between re-issuance, change only parts of their records and that they do not choose to set an empty (';') record. As such, these are likely falsely issued certificates.

5.5.2 CAA Records vs CT Logs

In Unterabschnitt 5.5.1 we match the issuer of the certificates from web servers against their respective CAA records. For all domain names with at least one CAA record, we query CT logs to fetch all other certificates bound to those names that are valid at the time of measurement. CT logs provide 1*M* certificates for about 191*k* unique domain names—reduced to 766*k* certificates after removing duplicates (*e.g.,* when a CA logs both precert and leaf). To compare the results, we focus on domains with CAA records or mismatching server certificates.

In 98% of cases the CAA matching state is consistent, *i.e.,* the certificates we retrieved from the web server and the certificates we retrieved from the CT logs have the same consistency with the domain's CAA records. Among inconsistent cases, $\approx$ 55% of domains have servers that return a CAA-matching certificate while CT logs contain at least one other valid certificate that does not match the CAA constraints. 30% present a certificate that does not match the domain name (see Abschnitt 5.4) while we find a logged certificate that fulfills the CAA constrains. Abbildung 5.5.2 summarizes the findings.

5.5.3 TLSA Records vs CT Logs

DANE sets constraints on certificates provided by a service endpoint. We take advantage of CT logs to find certificates that are not deployed, but match either TLSA records or the domain name of DANE-

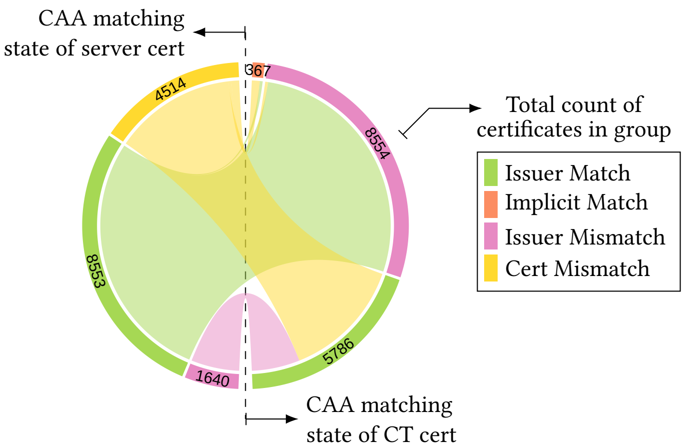


Abbildung 5.5.2: Distribution of inconsistencies between CAA matching state of server (left half-circle) and CT-logged certificates (right half-circle).

enabled services. We explore (*i*) the reasons for the existence of TLSA records that do not match the server certificate, (*ii*) the relationship between valid but not-deployed certificates and TLSA records, and (*iii*) compatibility of CAA records with certs referred to by DANE.

TLSA Records that do not Match the Server Certificate. We observe 1171 names with TLSA records that do not match the provided certificate. Thus, DANE-enabled clients would consider these certificates invalid. We query the `crt.sh` CT database for certificates that match these records and find 1494 certificates (note that a single domain can have multiple TLSA records). Abbildung 5.5.3 depicts how these are distributed with respect to their validity (Unterabschnitt 5.3.3) and relative expiration time.

The majority of mismatching TLSA records references certificates (leaf or CA) that are expired or have been removed from current trust stores. An indication that TLSA records have not been updated to reflect changes of the CA or leaf certificate. We also see TLSA records reference existing and valid certificates which are not deployed.

Undeployed Certs that Match TLSA Records. For 945 domain names with TLSA records that fit their respective web server certificate we find 1171 more matching certificates in CT logs that are *not* included in the certificate chain of the respective web servers. The majority (782) are leaf certificates, and the rest (218) are from intermediate or root CAs.

All leaf certificates are either renewed or expired versions (same public key) of the certificate provided by the server.

Certificates Matching DANE-secured Domains. For all domain names with TLSA records, we query CT logs for matching certificates (subject or SAN) and keep only valid ones (relative to the measurement time). We only fetch leaf certificates and not the complete chain due to limitations in `crt.sh` database API. Thus, we can only authenticate TLSA end-entity (PKIX-EE or DANE-EE) records against logged certificates. A total of 10358 certificates match 3320 unique domain names.

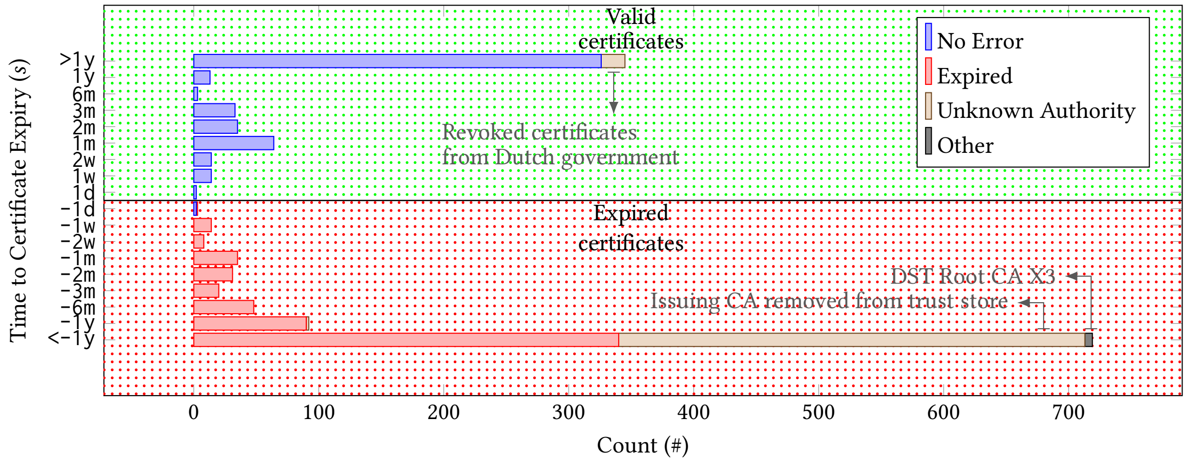


Abbildung 5.5.3: Certificates from CT logs that match TLSA records but are not deployed on the web server, grouped by their relative expiration time and annotated with their validation state.

Tabelle 5.5.1 summarizes for how many certificates can be authenticated by the TLSA records. Notable are cases where a TLSA record matches the server cert but not the CT cert and vice versa (rows #3 to 6). The majority of server-match/CT-mismatches (row #3) are TLSA records that specify a certificate by its fingerprint (and not SPKI), so that even if the same key is renewed, it will not match. We also observe Cloudflare backup certificates (of row #4), where multiple certificates are valid simultaneously at the same time.

5.5.4 TLSA Records vs CAA Records

Most DANE-secured certificates collected from CT logs have the same CAA matching status as their corresponding server certificates. The only notable exception is a set of domain names with matching CAA records and TLSA records referencing a valid certificate that does not match CAA (19 in total). We also observe 19 cases where the server returned a certificate that does not match its domain name, but TLSA records reference certificates that match CAA constraints (9 already expired).

5.6 Discussion

In this section, we discuss our findings comprehensively and provide recommendations for name owners and CA operators.

Lack of DNSSEC deployment defeats CAA purpose. Missing DNSSEC deployment can be abused to trick CAs into misissuing domain-validated certificates [217]. While the CAA standard does not require CAA records to be secured by DNSSEC, the use of DNSSEC is “strongly RECOMMEN-

Tabelle 5.5.1: Number of CT logged certificates for all DANE-secured domain names grouped by authentication status.

#	TLSA Record Authenticates: Server Cert	TLSA Record Authenticates: CT Cert	Same Issuer (Server and CT)	Count
1	✓	✓	✓	959
2	✓	✓	✘	1
3	✓	✘	✓	2,432
4	✓	✘	✘	294
5	✘	✓	✓	70
6	✘	✓	✘	15
7	✘	✘	✓	1,777
8	✘	✘	✘	431

DED" [357]. Yet, we observe a small overlap in the deployment as only ≈ 7% of CAA-enabled domains are also DNSSEC-secured, opening an unnecessary attack surface.

Reused strings in CAA *issuer-domain-name* weaken security. Not only do some CAs accept multiple strings, but some strings are accepted by multiple CAs. We found one string accepted by up to 21 CAs, see Tabelle 5.4.1. While these are likely resellers or managed PKIs, this nonetheless weakens the restrictiveness the system intends to implement. Users are likely unaware of this when they choose their CA and create CAA records. This conflicts with the goal of CAA records.

An authoritative source for CAA mappings can improve deployment and security. We found many invalid strings clearly indicating that CAA records are configured manually, even though supportive tooling (*e.g.,* `https://sslmate.com/caa/`) exists. A well-defined interface to discover legitimate CAA strings of a CA could help avoid typos, ease the setup, and increase the stability of the ecosystem. Furthermore, the current specification and deployment model impacts auditors who must regularly scan the Certification Practice Statements (CPS) of CAs to manually collect all valid CAA strings, which may lead to incorrect and incomplete data sets.

CT Logs can help CAA evaluators. A CA only validates issuance constraints at the time of certification. An evaluator, however, can validate CAA only after certificate issuance and might face CAA records other than what the CA observed. This also explains why the number of CAA mismatches grows when the certificate age increases. This time discrepancy can be reduced to a minimum by also acting as a CT Log Monitor and validate CAA records as soon as a certificate (or its precert) is logged.

DANE can enhance CAA. The CAA standard [357] claims that DANE records are relevant *after* certificate issuance. In fact, we found that DANE records are beneficial along the whole process and can even partly substitute CAA features. A TLSA TA record referencing root or intermediate certificates of a CA, for example, can be used to convey the same semantics as a CAA `issue` tag. Furthermore, a "to be signed certificate" provided to the CA for signing already carries the public key and can be referenced by a TLSA record before certification.

Third party validation can help name owners. Our correspondences with operators (Abschnitt 5.4) reveal that errors in CAA entries are partly due to incorrect instructions or automatic configurations by service operators (*e.g.*, hosting and DNS providers). In addition, lack of feedback (using `iodef`) by issuing CAs in face of misconfigured or malformed CAA records makes it harder for name owners do detect such errors. To address this, we have provided a public validation tool available on `https://caa.secnow.net`, which can validate CAA records, provide comprehensible explanations on effect of each CAA record, and detect errors and suggest solutions.

5.7 Conclusion

In this paper, we analyzed the Web PKI ecosystem with a focus on consistency between CAA, DANE, CT Logs, and X.509 certificates. Ideally, each protection mechanism is correctly configured, and all agree on the set of legitimate certificates of a name. Our results show different deployment behavior. Most alarming, a small share of certificates should not have been issued since they contradict CAA records in place. Furthermore, we found several configuration mistakes that could be prevented by better tooling and explicit best common practices. We argue that there is room for improvement. The CAA record structure itself is simple. Clear information about the CAA strings and better support for configuring CAA records can help to increase adoption and help avoid common errors we identified.

We encourage domain owners to maintain their CAA records and keep them consistent with deployed certificates. Even more, we encourage extending the deployment to include DNSSEC and to consider TLSA records, as these techniques put web security on much firmer grounds. Enabling automated evaluation of issued certificates and consider data of all possible protection mechanisms would be one step towards a consistent and healthy Web PKI.

Acknowledgments. This work was supported in parts by the German Federal Ministry of Education and Research (BMBF) within the project PRIMEnet.

Appendices

5.A Ethical Considerations

Our measurements use common place actions (requesting DNS records, performing TLS handshakes) and were spread out to avoid hitting the rate limits or other protective measures. The information we collect is public and explicitly designed to be checked by the relevant parties. We neither reveal new security vulnerabilities nor do we single out any entity specifically. We individually notified all parties that were negatively impacted by misconfigurations or notable errors.

5.B X.509 Certificates

In the Web PKI X.509 build a chain of trust from a self-signed *Root* certificate over one or more *Intermediate* certificates to a *Leaf*. As shown in Abbildung 5.B.1 the hierarchy is expressed through signatures from the root towards the leaf in one direction and references to the issuer in the other direction.

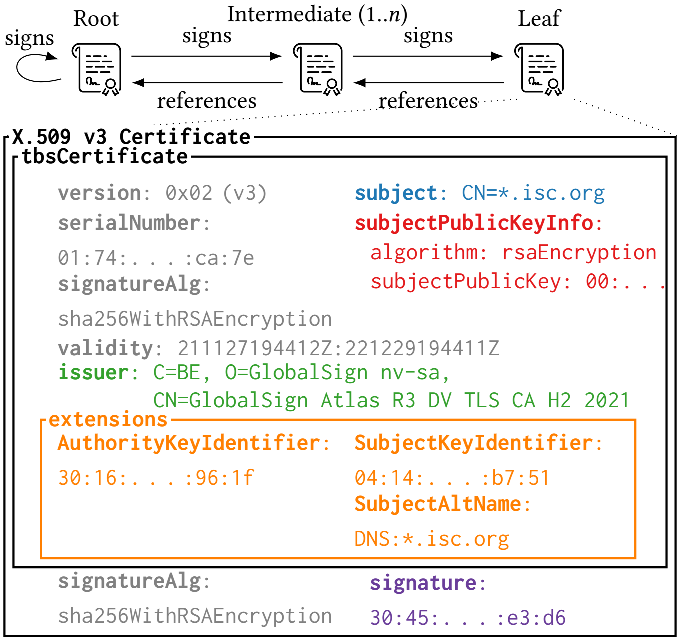


Abbildung 5.B.1: X.509 certificate chains and structure.

5.C Measurement Details

Tabelle 5.C.1 lists when each measurement task described in Abschnitt 5.3 was performed and which tools we used. Tabelle 5.C.2 provides additional information about the DNS queries when collecting X.509 Certificates via TLS (M.2).

5.D CAA Matching Algorithm

The consistency between CAA records and the certificate issuer can be classified into three main groups (colors in Abbildung 5.D.1): No CAA (relevant CAA RR set is empty), Issuer Mismatch (CAA RRs does not match the cert issuer), and Issuer Match. Mismatching cases are further divided to denote if mismatch was caused by empty (";") or malformed CAA `issue/issuewild` values. Implicit Match is semantically equivalent with Issuer Match but specifies the case where relevant CAA RR set is not empty yet has no `issue` or `issuewild` records.

Tabelle 5.C.1: Measurement dates and deployed tools for each task in our pipeline visualized in Abbildung 5.3.1.

Task	Name	Date	Tool
T.1	Query AA Records	2024-04-07	dig
T.2	Check ports 80/443		nc
T.3	Follow HTTP/HTML Redirects		libcurl
T.4	Connect in Browser (Puppeteer)	2024-04-12	Puppeteer
M.1	Query DNS Records	2024-04-12	JavaScript
M.2	Collect X.509 chains		
M.3	Fetch certificate from CT logs	2024-04-24	Go

Tabelle 5.C.2: List of DNS resource records that we collect for a given domain name. Subdomain column denotes the prefix added to the domain name for the query; `@` denotes empty label.

#	Subdomain	Type	Description
1	`@`	SOA	Zone apex
2	`@`	A	IP address(es)
3	`@`	CAA	CAA RRs
4	`_443._tcp`	TLSA	DANE
5	`_validation-contactemail`	TXT	DV E-Mail contact
6	`_validation-contactphone`	TXT	DV Phone contact

To match if a certificate matches an issuer denoted by a CAA `issue/issuewild` record, we have composed our own approach that maps a CA string identifier to a CA based on its subject name, organization name, or other features. For example, for the following CAA RR:

```
example.com          CAA 0 issue "web.com"
```

we assert that the CA certificate carries ‘Network Solutions L.L.C.’ as organization name in its subject field.

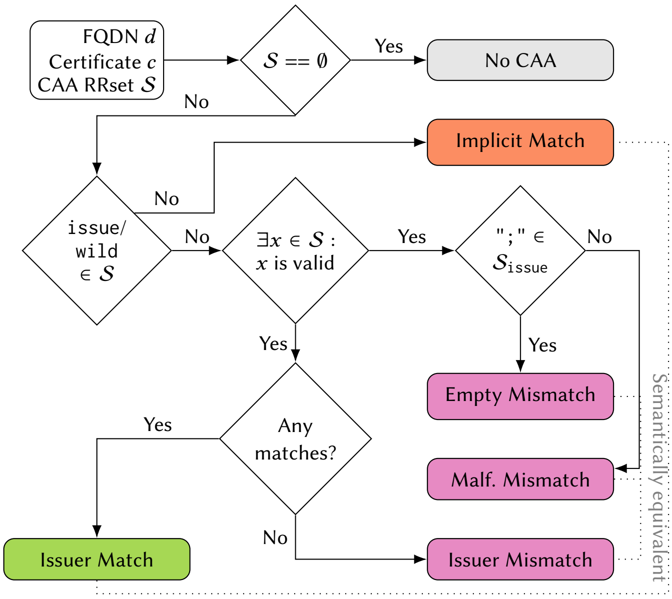


Abbildung 5.D.1: The algorithm we use to check the consistency between a certificate c and relevant CAA resource records set S for a given domain d.

Kapitel 6

Security of Critical Services in the WWW: Measuring Namespaces, DNSSEC, and Web PKI

Abstract

To verify the applicability of our model presented in Kapitel 2, we investigate the online presence of *Alerting Authorities* in the US, selected *German Emergency Service* providers, and *UN member states*. We observe partially enhanced security relative to global Internet trends, yet find cause for concern as only about 6% of unique hosts cater to secure resolution. About 46% of investigated organizations use shared certificates with 1% of all organizations having no or invalid certificates. Two thirds of organizations are not uniquely identifiable and as such lack the basic requirement of trustworthy communication.

6.1 Measurement Method and Data Corpus

In this section, we describe our framework to identify the Web presences of organizations and analyze their domain names and Web certificates. Abbildung 6.1.1 summarizes our measurement method from preparation phase to data gathering and analysis.

6.1.1 Identifying Web Presences

Identifying the Web presence of an organization is challenging since there is no mapping system that unambiguously provides the URL of an organization. Search engines, however, provide at least a set of potential entry points. We now describe our approach to identify the primary website of an organization. Our goal is to reduce manual actions.

Our framework queries and scrapes the Google search engine. Combining the name and territory of operation (*e.g., Fresno Police Department CA*) significantly reduces the result set. Using the organizations under investigation in this paper, each query yielded between 8 and 28 results.

Since the results are not necessarily ranked to have the official URL first, we exclude popular but incorrect results based on a list of inapt domain names (*e.g.,* social media sites and yellow pages). The topmost remaining URL is then selected as primiary website for the organization.

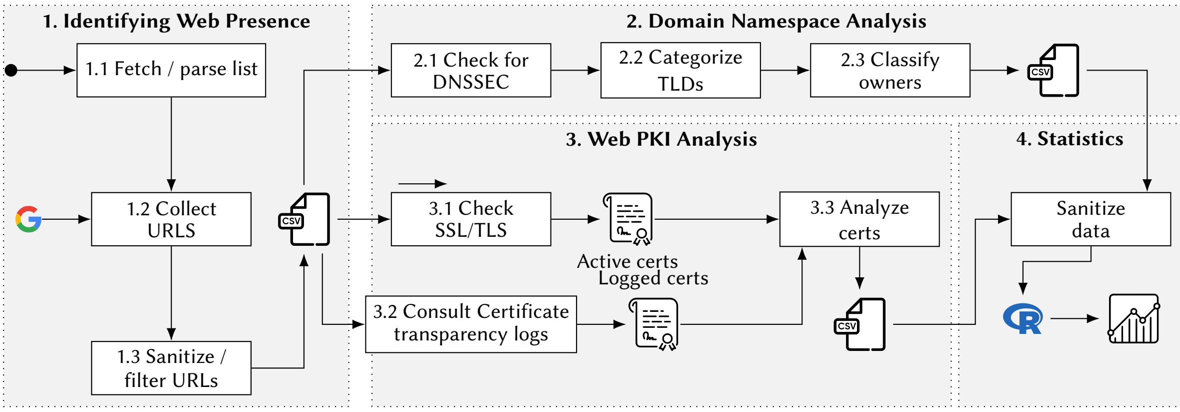


Abbildung 6.1.1: Toolchain to gather and analyze data about organizations included in our three data-sets

Our framework then provides a list of collected URLs that can be manually checked to remove any mismatches and correct falsely associated URLs which were not detected automatically, *e.g.,* same URL for homonymous counties in different states. In our case, we had to adapt 1%-4% manually, for details see Unterabschnitt 6.1.4.

Finally, for each URL (*e.g.,* `https://www.fresno.gov/police`) our framework extracts the FQDNs (*e.g.,* `www.fresno.gov`) and path segments (*e.g.,* `/police`).

6.1.2 Analysing Domain Namespaces

In the second phase, we identify domain namespaces, categorize them, and check for DNSSEC.

To analyze top-level domains and service names separately, our framework splits second-level domains (SLD) from TLDs (or more specifically the public suffix). For example, for '`www.ci.tracy.ca.us`', '`ca.us`' is the TLD and '`tracy`' the effective SLD.

Our framework categorizes TLDs into generic, sponsored, *etc.* Based on a list of predefined keywords (see Tabelle A.I in Appendix) each Alerting Authority is mapped to a *field of activity*, *i.e.,* either Public safety, Governmental, Law enforcement, Military, or Educational. For other organizations such Ministries of Foreign Affairs, the field of operation is given by definition.

We consider a domain name DNSSEC-enabled, if there exists at least one signed and verifiable `A` record for that name. Furthermore, we calculate the distance from the domain name to the next DNSSEC-enabled parenting zone, *i.e.,* the nearest zone cut with a verifiable `SOA` record. For example, `www.hamburg.de`, an insecure name, is two zone cuts away from the next DNSSEC-enabled zone as it is an alias for `e29654.dscb.akamaiedge.net.` under the zone `dscb.akamaiedge.net.`, with the next DNSSEC-enabled zone being `.net` (`akamaiedge.net.` is insecure). This distance indicates how many zones between the zone in question and a DNSSEC-enabled zone exist that need to enable DNSSEC as well so a trust chain can be established.

The results of our analysis on domains names are presented in Abschnitt 6.2.

Tabelle 6.1.1: Example entries and total count for our datasets

	Entries		Example		
List	Total	w/ URL	ID	Territory	Name
US-AA	1805	1789	200313	NC	Burke County
DE-EM	800	798	BOS006	Hamburg	Deutsches Rotes Kreuz
MFA	193	169	MFA024	Brazil	Ministério das Relações Exteriores

6.1.3 Collecting Web PKI Certificates

We investigate the current and historic adaption of Web PKI certificates by finding certificates that match domain names under consideration.

To study the current state, we use OpenSSL version `1.1.1l` CLI to fetch complete certificate chains, perform validation, and verify revocation status using stapled Online Certificate Status Protocol (OCSP) [345], manual OCSP [448], or Certificate Revocation Lists (CRL) [297].

For our historical analysis, we use CT logs [388], [214]. We access the publicly available database provided by *Sectigo* under crt.sh, which audits 92 log servers from 20 organizations in February 2022. For any given host name, our toolchain queries the database for certificates that include either (*i*) the host name, (*ii*) a wildcard covering the host name as in the subject name, or (*iii*) include the host name in the list of subject alternative names (SAN). The results of our analysis are described in Abschnitt 6.3.

6.1.4 Data Corpus

We apply our method to analyze the security profiles of web services of three stakeholder groups. Each group has been selected due to its critical communication role in a specific context:

US-AA All "state, local, tribal and territorial" *Alerting Authorities* (AA) with access to FEMA's *Integrated Public Alert & Warning System* (IPAWS) [335].

DE-EM German Emergency Service providers consisting of police and fire departments, city hall portals, and German Red Cross chapters of the biggest 100 cities in Germany.

MFA Ministries of Foreign Affairs of all countries represented at the United Nations, as used by the *Diplomatic Pulse* platform [371].

Our `US-AA` dataset is based on the list of "complete" and "in process" Alerting Authorities provided by FEMA [334] in 2021-09-10. Each entry represents an organization by a unique ID, a name, and a territory of operation (including unincorporated territories) for which we derive URLs, fetch certificates, *etc.*

The `DE-EM` dataset is based on the names of the biggest 200 German cities and the German terms `{polizei, feuerwehr, deutsches rotes kreuz, stadt}` (*i.e.,* Police, Fire department, German Red Cross, and City). These combination of these terms guides the identification of Web presences.

The `MFA` list is provided by the *Diplomatic Pulse* [371], a search engine for official press releases of UN member states. Some provided URLs have been outdated and the list was partially corrected for this work by manual verification and replacement of missing or wrong addresses. Tabelle 6.1.1 represents an example entry from each dataset alongside the total entry count.

Our measurements were carried out from October 2021 to January 2022. To prevent bias from specific vantage points, *e.g.*, limited access due to geo-blocking, we conducted our measurements from various vantage points in Europe and the US. Overall, we identify 1789 Web presences for the `US-AA` dataset, 798 entries for the `DE-EM` dataset, and 193 entries for the `MFA` dataset.

6.2 DNS Namespace Analysis

In this section, we aim to answer the following questions by studying domain names:

1. Does each service provider have its own dedicated domain name?
2. How do service providers integrate in the global DNS namespace?
3. Do service providers secure their names using DNSSEC?

The first question is concerned with how an organization maintains its online presence without introducing unnecessary dependencies. Lack of a dedicated name, for example, leads to dependence on someone else for authentication and data security as X.509 certificates are bound to domain names. The second question aims to investigate the preference regarding TLDs to take advantage of recognizability (*e.g.*, governmental organization under `.gov`) and security (restricted vs. non-restricted TLDs). The last question considers measures taken into account to secure names against threats such as spoofing or DNS hijacking, which can also lead to impersonation and phishing. Tabelle 6.2.1 summarizes our findings.

6.2.1 Dedicated Domain Names

We consider an entity to have a dedicated DNS name either if it has its own effective SLD, or has been assigned a sub-domain under the namespace of its parent organization or any generic service provider that is not shared. For example, the *Tehama County Sheriff* (`tehamaso.org`) has its own dedicated name whereas *Apache County Sheriffs Office* (`www.co.apache.az.us/sheriff/`) does not.

Out of the total 1,788 collected URLs for the `US-AA` dataset, 1,747 unique domain names exist, showing that in some cases multiple entities are subsumed under the same domain, *e.g.*, different agencies all under the domain name of a single state. For the `DE-EM` dataset, there exist 595 unique domain names for 798 collected URLs. All 169 collected URLs for the `MFA` dataset are under distinct domain names.

The share of dedicated names varies largely among different fields of operation. Ministries of Foreign affairs have their own dedicated name. Among the US Alerting Authorities all educational entities

Tabelle 6.2.1: Top-level domains in use by organizations studied in this work

	TLD		Registration		Registry		Domain share (%)			Domain count (#)			DNSSEC support (#)		
Type	Label	DNSSEC	Restricted	Fee/year	Name	Country	US-AA	DE-EM	MFA	US-AA	DE-EM	MFA	US-AA	DE-EM	MFA
gTLD	`.com`	✓	✘	< 15 $	Verisign	US	18.23 %	0.25 %	0 %	325	2	0	12	0	–
	`.info`	✓	✘	< 15 $	Afilias [1]	US	0.05 %	0 %	0 %	1	0	0	0	–	–
	`.net`	✓	✘	< 15 $	Verisign	US	4.53 %	0.25 %	0 %	81	2	0	2	0	–
	`.org`	✓	✘	< 15 $	PIR	US	24.94 %	0.62 %	1.18 %	446	5	2	31	0	1
							47.76 %	1.12 %	1.18 %	854	9	2	45	0	1
ccTLD	`.<ISO-3166>`	(✓)†	(✘)	–	–	–	4.19 %	88.97 %	28.99 %	75	710	49	0	44	7
ccSLD	`.<state>.us`	✓	✓	–	Neustar	US	13.98 %	0 %	0 %	250	0	0	5	–	–
	`.*.<ISO-3166>`	(✓)	✓	–	–	–	0 %	0 %	69.23 %	0	0	117	–	–	8
							13.98 %	0 %	69.23 %	250	0	117	5	–	8
sTLD	`.edu`	✓	✓	77 $	Educase [2]	US	0.50 %	0 %	0 %	9	0	0	–	–	0
	`.gov`	✓	✓	‡ 0 $	CISA [3]	US	32.32 %	0 %	0.59 %	578	0	1	69	–	0
	`.mil`	✓	✓	–	DoD NIC	US	1.23 %	0 %	0 %	22	0	0	14	–	–
							24.06 %	0 %	0.59 %	609	0	1	83	–	0
geoTLD	`.eu`	✓	(✘)	< 10 $	EURid	BE	0 %	0.25 %	0 %	0	2	0	–	0	–
	`.hamburg`	✓	✘	≈ 50 $	Hamburg Top-Level-Domain GmbH	DE	0 %	0.12 %	0 %	0	1	0	–	0	–
	`.nrw`	✓	✘	≈ 30 $	mmx.co [4]	US	0 %	9.52 %	0 %	0	76	0	–	0	–
							0 %	9.89 %	0 %	0	79	0	–	0	–
									Total	1788	798	169	133	44	16

[1] Acquired by Donuts Inc.

[2] subsidiary of Verisign, [3] Previously managed by *General Services Management*, [4] Minds + Machines GmbH operated by GoDaddy

† See ICANN TLD DNSSEC support [369]

‡ Fees were dropped from 400 $ after takeover by US CISA in 2021

Datasets: US Alerting Authorities (`US-AA`) / German emergency management (`DE-EM`) / Ministries of foreign affairs (`MFA`) (total of 6) and over 90% of governmental entities (628 out of 673) have dedicated names. In contrast, ≈ 23% of public safety entities (197 out of 852), and less than 30% of law enforcement (68 out of 216) and military organizations (6 out of 19) have their own names. In the case of German Emergency providers, 97% of the municipality portals (198 out of 200) and German Red Cross chapters (194 out of 199) are served by dedicated names, followed by 59% of fire departments (118 out of 200), and 23% of police departments (46 out of 199).

6.2.2 Namespace Structure

To analyze namespace structures, we group various TLDs and country code second-level domains (ccSLDs) as follows:

gTLD: generic top-level domain (*e.g.,* `.org`).

ccTLD: country code top-level domain (*e.g.,* `.us`).

ccSLD: country code second-level domain (*e.g.,* `gouv.fr`).

sTLD: sponsored top-level domain (*e.g.,* `.mil`).

geoTLD: top-level domains for "geographical, geopolitical, ethnic, social or cultural representation" (*e.g.,* `.eu`).

Each TLD group features different properties. In general, there are little to no delegation limits and naming conventions for names under gTLDs or ccTLDs, except for the `.us` namespace. Under the `.us` ccTLD more than 3,000 names are reserved and unavailable for public registration. This DNS sub-tree has a rigorous structure on the second, third, and fourth level. The structuring reflects the "political geography" (see RFC1480) and defines a number of reserved names for designated organizations or purposes and territory of operation (*e.g.,* county or city). Sponsored TLDs (`.edu`, `.gov`, and `.mil`) impose stricter eligibility requirements and thus have an advantage over gTLD names as only eligible registrants are granted domain name ownership in these namespaces. The sTLDs, however, are mainly limited to the US so that non-US entities are forced to make use of local counterparts under their respective country code SLDs, *e.g.,* `.ac.uk` for academic institutes in the UK or `gouv.fr` for the French government.

The majority of `US-AA` entries are located under gTLD namespaces followed by sTLD, ccSLD, and ccTLD namespaces (see Tabelle 6.2.1). It is noteworthy that the `.us` locality namespace (part of `.us` ccSLD) exhibits a relatively low penetration among AAs. When considering the canonical forms `[`**`ci`**`,`**`co`**`].<locality>.<state-code>.us` for cities or counties, we observe only 9 out of 66 actual cities that use this naming pattern. Similarly, for names that include the term *county*, there are only 218 of 696 cases using the canonical form.

In contrast to Alerting Authorities, `DE-EM` and `MFA` organizations face more limitations in choice of appropriate and restricted namespaces. For instance, none of the entries in `DE-EM` are under restric-

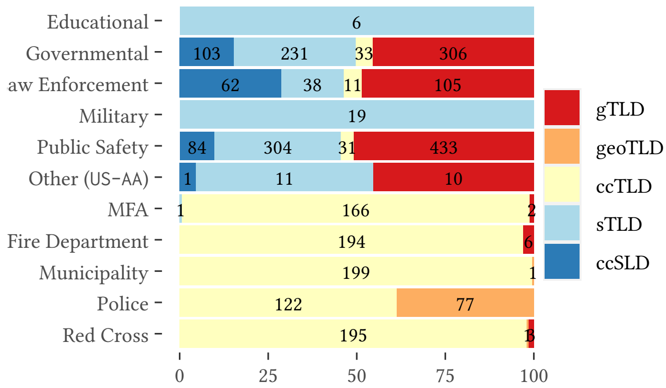


Abbildung 6.2.1: Distribution of TLD types per operation territory

ted namespaces (≈ 90% under `.de`), and the majority of MFAs resolve to `ccSLD` namespaces of their countries (*e.g.,* `.gov.af`).

Finally, we examined whether the specific choice of top-level domains of an organization correlates with its field of operation. Abbildung 6.2.1 depicts how widely used various TLD types are in different fields of operation. The majority of Alerting Authorities opt for gTLDs with the exception of educational and military organizations that exclusively take advantage of restricted TLDs (`.edu` and `.mil` respectively). `DE-EM` and `MFA` organizations primarily choose `ccTLD` namespaces.

6.2.3 DNSSEC Deployment

All investigated TLDs, except a number of ccTLD and ccSLD namespaces, are DNSSEC-enabled (see Tabelle 6.2.1). Nonetheless, we only observe 7.4%, 5.5%, and 9.4% of entities that are part of DNSSEC-enabled zones in `US-AA`, `DE-EM`, and `MFA` datasets. It is noteworthy that 77.6%, 70.6%, and 35.5% of entities in respective lists are directly under DNSSEC-enabled zones and only need to activate DNSSEC on their own authoritative name servers, while 13.1%, 1%, and 53.2% are respectively 2 or more zones deeper in the DNS hierarchy and need at least one predecessor zone to enable DNSSEC before they can. In total, only ≈ 6% of unique hosts that we analyzed are signed and secure, which is still a higher DNSSEC penetration compared to the longitudinal study by Chung *et al.* [45], who measured 0.6% for `.com` and 1.0% for `.org` domains.

Only about 12% (65 out of 557) of unique `.gov` names support DNSSEC. This is surprising for two reasons. First, `.gov` names are mandated to implement DNSSEC. Second, the DNSSEC penetration is much lower than the ≈ 90% DNSSEC penetration among selected governmental organizations (sample set of approximately 1200 `.gov` SLDs) as measured by NIST [425].

6.3 Web PKI Analysis

The ecosystem of Web PKI revolves around X.509 certificates. We investigate the deployment and characteristics of certificates to answer the following questions:

1. To what extent do entities embrace the Web PKI?
2. How and for what purposes (identification vs encryption) are certificates utilized?
3. How is the historic landscape of X.509 shaped among organizations in our dataset?

6.3.1 Current Deployment of Web PKI

We first want to gain a better understanding of the current deployment of Web certificates by analyzing a snapshot of TLS deployment across public servers from our datasets.

The highest TLS deployment is among German emergency services ($\approx$ 99%), also with the lowest number of invalid certificates (< 2%). Comparably, $\approx$ 98% of US Alerting Authorities support TLS with only 2% erroneous certificates. Among Ministries of Foreign Affairs, 15% lack TLS support and nearly 15% of the certificates are unverifiable. Failed verifications are caused by server misconfigurations, self-signed or expired certificates. For validation, we use OpenSSL trusted root certificates (on Ubuntu Linux). In contrast to other Web PKI studies, we see less invalid certificates on average in our sample; Chung *et al.* [42] observed 65% on average considering the global IPv4 space in 2016, and Durumeric *et al.* [67] found $\approx$ 13% measuring Alexa 1M top domains in 2013.

6.3.2 Certificate Usage

Although the technical basis of all X.509 certificates is the same, the semantics of a certificate are defined by the governing validation policies and how a certificate is in use. A Web PKI X.509 certificate must at least fulfill domain validation, *i.e.,* authenticate the ownership of a domain name. For OV and EV certificates, the identity of the certificate holder (to various degrees) can be authenticated as well. Identification assurances, however, can be diluted through certificate sharing. Except EV certificates, both DV and OV certificates allow wildcard names as subject alternative names (SAN) to avoid enumerating all FQDNs under the control of the certificate holder. In practice, the SAN extension also allows sharing a certificate among different hosts. For example, in 2019, the US federal government issued OV certificates with more than 600 specific and wildcard SAN entries. Certificate sharing, on the one hand, defeats the purpose of identification and, on the other hand, expands the attack surface and increases operational costs so that if one of the hosts is compromised or the certificate is revoked, every other host necessarily also needs to be configured with a new certificate (sometimes called "fate-sharing").

To detect certificate sharing, we count the number of unique domain names in the SAN section of a certificate while ignoring `www.*` subdomains. We consider any SAN count greater than 1 or the existence of wildcard names as an indication of a shared certificate. In the case of EV certificates and regardless of the number of SANs, the certificate is considered dedicated as the certification authority is mandated to ensure that all alternative names are representations of one and the same entity. This is not necessarily the case for OV certificates as we discuss below.

Tabelle 6.3.1: Proposed security model and measurement results, which exhibit specific security features per data set.

Security Model						Measurement Results		
Security Dimensions				Assurance		Unique Hosts [#]		
#	Resolution	Transaction	Identification	Score	Profile	US-AA	DE-EM	MFA
1	✘	✘	✘	0	○	93	27	11
2	✘	✘	!	2	○	68	0	27
3	✘	!	✘	1	○	781	709	30
4	✘	!	!	3	○	713	18	85
5	✓	✘	✘	2	○	2	2	1
6	✓	✘	!	4	○	1	0	4
7	✓	!	✘	3	○	43	40	6
8	✓	✓	✘	4	○	0	0	0
					Total inadequate:	1701 (95%)	796 (99%)	164 (98%)
9	✓	!	!	5	◐	87	2	5
10	✓	✓	!	6	◐	0	0	0
					Total weak:	87 (5%)	2 (1%)	5 (2%)
11	✓	✓	✓	8	●	0	0	0

Measurable security dimensions (✘/0 : insecure, !/1 : partial, ✓/2 : secure) determine Assurance Scores ($resolution + transaction + 2 \times Identification$) aggregated into Assurance Profiles (● strong, ◐ weak, ○ inadequate).

Tabelle 6.3.2: Number of unique hosts per Assurance Profile grouped by field of operation.

		Assurance Profile		
Dataset	Field of operation	○	◐	●
US-AA	Educational	6	0	0
	Governmental	646	28	0
	Law Enforcement	206	10	0
	Military	7	12	0
	Public Safety	819	33	0
	Other	18	4	0
MFA	Ministry of Foreign Affairs	164	5	0
DE-EM	Fire Department	200	0	0
	Municipality	200	0	0
	Police	197	2	0
	Red Cross	199	0	0

● strong, ◐ weak, ○ inadequate (see Tabelle 6.3.1)

For those organizations behind an SSL/TLS enabled server, we observe that only a small fraction is represented by an EV certificate, $\approx$ 2% for `US-AA`, 1% for `DE-EM`, and $\approx$ 3% for `MFA` entries. As we discuss later in Section 6.4, this is likely due to a decision by major browser vendors to abandon visual cues for different validation types altogether [75]. OV certificates amount to one third for entries in `DE-EM` and `MFA` datasets each, and 22% for `US-AA`; 84%, 82%, and 90%, respectively, using shared certificates. The remaining majorities are represented by DV certificates with a portion of 40% (`US-AA`), 41% (`DE-EM`), and 57% (`MFA`).

We observe common deployment models that make use of shared certificates. Most prominently, Cloudflare uses its own CA to issue an OV certificate to itself, *i.e.,* 'Cloudflare, Inc.' as organization name and `sni.cloudflaressl.com` as common name, and includes a customer domain name as SAN. In this model, despite the organization validation, the certificate cannot be used to identify the service provider, *e.g.,* Wilco County in Texas under `www.wilco.org` provides a certificate that does not relate to its infrastructure provider Cloudflare. Another model, observed in the `US-AA` dataset, is when a certificate is shared among different customers of a multitenancy cloud provider (dubbed as "cruise-liner certificates" by Cangialosi *et al.* [33]). For instance, Granicus, a cloud provider for governmental organizations in the US, uses Let's Encrypt DV certificates to create a single certificate for many customers at once. Here, not only is identification ambiguous but also confidentiality is jeopardized by shared server keys, leaving customers insufficiently isolated from each other. It is worth noting that Let's Encrypt certificates only allow up to 100 DNS type SANs. The disadvantage of such an approach compared to the former model is that a single compromised or revoked certificate affects multiple un-

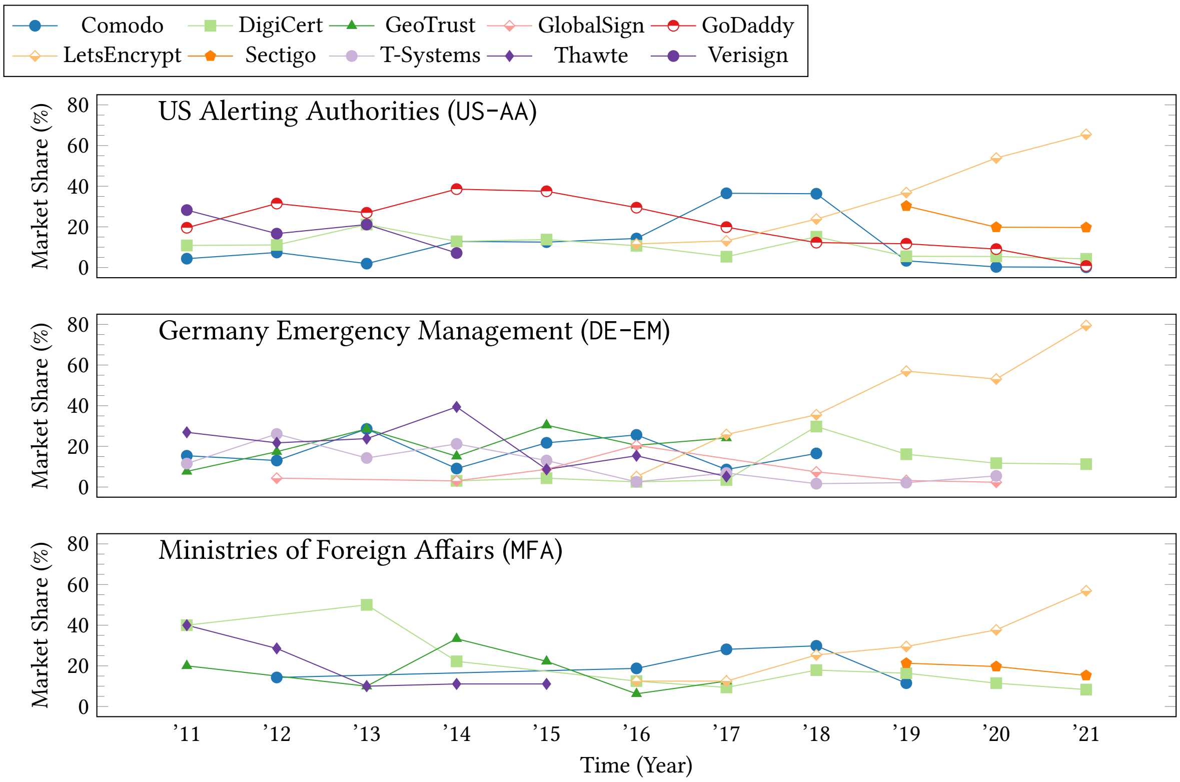


Abbildung 6.3.1: CA Market share per year grouped by dataset

related customers, often called "fate sharing." Beside cloud service providers, individual organizations also commonly make use of shared certificates to have a single certificate to subdomains that are all managed by a single technical entity.

Tabelle 6.3.1 combines our findings from this Section and Abschnitt 6.2 to reveal different combinations of DNS and X.509 certificate characteristics, linked to different levels of assurance. In Tabelle 6.3.2, we group our results by organization types. Due to low DNSSEC deployment, lack of TLSA records, popularity of open TLDs, and pervasiveness of DV certificates, the majority of hosts fall under the inadequate Assurance Profile.

6.3.3 Historic X.509 Certificate Landscape

The historic analysis of X.509 certificates collected from Certificate Transparency logs helps us to gain a better understanding of security policy changes related to examined entities and CAs. We span eleven years. It should be noted that the total number of organizations providing publicly logged certificates changes for each year. We consider this in the following by normalizing the results either with respect to the number of unique hosts or the total number of certificates valid per year.

Certification Authorities. In addition to common regulations, certificate authorities implement and follow their own set of policies. From the perspective of relying parties, *i.e.*, Web users, such policies are opaque and as long as a CA is included in a user's trust store, it is considered trustworthy. For

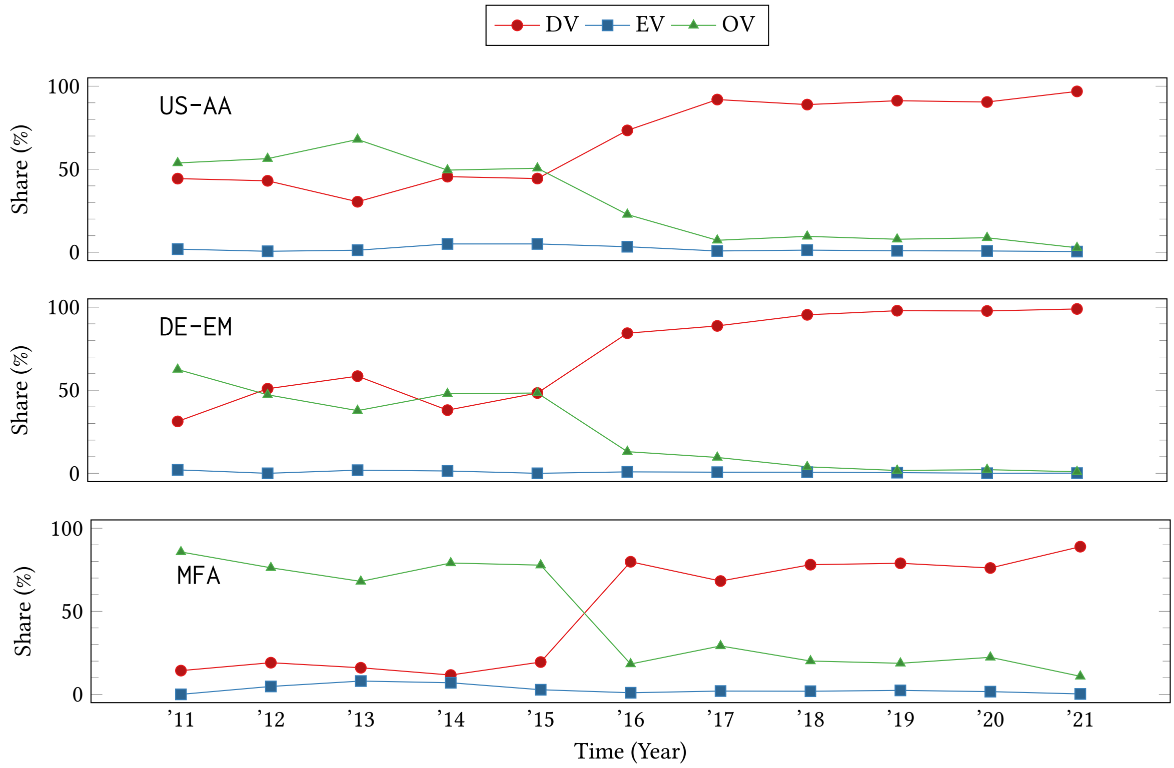


Abbildung 6.3.2: Development of certificate types in the past decade

the subscribers, however, these policies among other factors such as fees, offered certificate types, and operation costs are decisive in choosing an appropriate CA.

We focus our analysis on certification authorities with a market share of at least 20% in any year during the past decade. The calculation of market share is based on the number of *covered* subscribers. We use the term *cover* to differentiate from issuance: if a subscriber, for example, received a certificate from a CA valid from 2014 to 2016, we consider it to be covered by that CA for 2014, 2015, and 2016. Further, if a CA issues multiple short-lived certificates (*e.g.,* 90 days) for a subscriber within a given year, we only count that host as covered once in that year by the issuing CA.

Abbildung 6.3.1 depicts our findings in terms of relative market share development in the past decade. Compared with the CA market share for the Alexa 1M top domain list throughout the past decade [1], [67], [111] we observe parallels, such as decline of the GoDaddy market share and rapid gains of Let's Encrypt, as well as discrepancies that cannot directly be explained due to dynamic nature of and fluctuations in the Alexa top list. This figure also highlights two factors that were evidently decisive for the organizations in our datasets in their choice of CA: convenience and cost. GoDaddy (among US-AA) and T-Systems (among DE-EM), for example, which have been the market leaders at some points in time, provide Web hosting and domain name registration in addition to certification services in convenient packages. In contrast, Let's Encrypt, which has surged to the top in the short period after its public offering, offers automated DV certification at no cost.

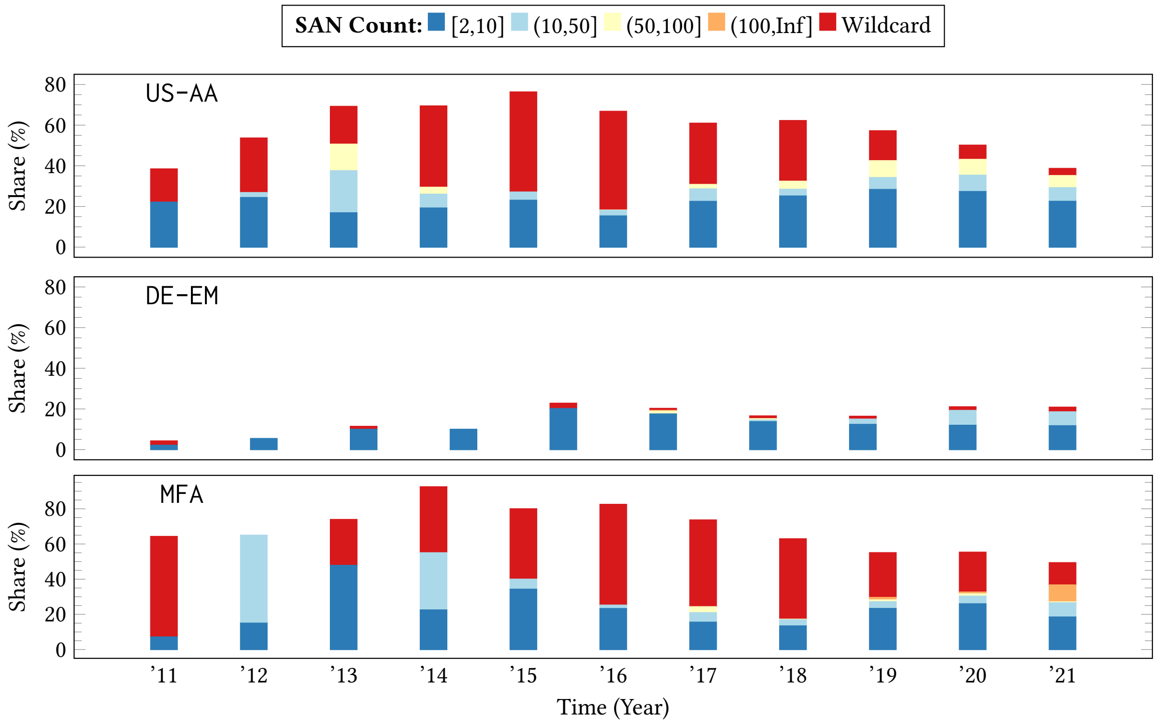


Abbildung 6.3.3: Share of host names represented by certificates with more than 2 unique SAN entries (excluding EV certificates)

Validation types. In Abschnitt 2.3 we discussed the role of OV/EV certificates for secure identification. Although the current deployment of investigated entities (see Unterabschnitt 6.3.1) reveals a low penetration of such certificates, historically a higher share of entities made use of OV/EV certificates as depicted Abbildung 6.3.2. The surging popularity of free and automated DV certificates (see Abbildung 6.3.1) has evidently led to a decrease in usage of OV/EV certificates.

Certificate Sharing. Multitenant Web hosting and security service providers make use of shared certificates as discussed above. Abbildung 6.3.3 depicts the development of certificate sharing based on our analysis of historic data. Here, we also notice an increasing number of certificates shared across hosts which *do not* belong to the same logical entity. Most critically, also among OV certificates, we see service providers obtaining certificates under their names and listing the hosts of their customers as SANs. This practically defeats the identification purpose of the certificates. At the time of writing, we observe cases of such certificates listing SANs that clearly belong to separate entities, *e.g.*, `mo.gov`, `asap.farm`, and `incapsula.com` within the same certificate. In this very specific case, records from the *Wayback Machine* archives show that `asap.farm` has previously belonged to Missouri Department of Agriculture but it was never removed from the certificate even though the domain name registration was transferred to another entity.

6.4 How to Proceed Further?

Our results draw a rather alarming picture of how security measures of Web-based communication are implemented and integrated. We now discuss some possible reasons for the observed deficiencies and suggest alternatives.

Possible reasons for lack of security. Trustworthy Web communication is complex because it involves consumers as well as infrastructure operators such as CAs, ISPs, and browser vendors. CA practices and guidelines diverge [53], some automated DV certification services are susceptible to impersonation attacks [205], [217], and some root CAs do not restrict certification scope for their intermediate CAs [67]. The high fraction of invalid and shared certificates might relate to carelessness regarding the Web PKI trust model (self-signed certificates) or additional (not only financial) configuration [133] and certification costs. Higher costs and complexity are also the reason why certificate owners rarely make use of OV/EV certificates.

DNS registrars do not always offer DNSSEC by default or free of cost [44] and not all ISPs bother to operate DNSSEC-aware recursive resolvers that properly verify signed DNS records [45], [253]. Missing DNSSEC support [369] can partly be traced back to its complexity, *e.g.,* correct implementation of key transitions [186]. This might also be the reason why some domain names under `.gov` do not support DNSSEC even though DNSSEC is mandatory under this TLD.

Provide comprehensive security assessment to end users. The complexity of secure communication must no longer be hidden from end users. Rather, there is now a more pressing need than ever before for users to be educated in better understanding the semantics of domain names [201], [206] and Web PKI certificates and their practical use. Instead of the recent tendency by browser vendors to remove security signals, we must improve visual security cues and expand them *in meaningful ways* to different levels in Web browsers. Confusing TLS warnings [7] should be avoided and EV indicators should not be eliminated but enhanced [75]. Offering alternative CA trustworthiness assessment measures beyond the standard binary trust model [37], [71], [108] could be an additional option. The security model proposed in this article illustrates a clear example of a multi-dimensional framework that is designed to quantify this complex landscape. Our model is well suited to being visually integrated into a concise mechanism to present meaningful signals to end users.

Integrate routing security. Our security model should next be enhanced to consider routing security. Threats such as route hijacking expose communication to eavesdropping and interception. When traffic is intercepted, attackers may undermine end-to-end security by using a rogue certificate or outright prevent Web services by dropping traffic. Such a Denial of Service (DoS) cannot be protected by end-to-end encryption at all.

Integrating routing in the assessment of security is important to provide a comprehensive view. Similar to the verification whether a domain name is protected by DNSSEC, an enhanced assurance score should be developed to include whether IP prefixes of Web servers and/or DNS name servers are secured by the Resource Public Key Infrastructure (RPKI).

Providing information from lower layers and granting control over those layers to the application layer can allow implementing extensions directly in browsers, enabling users to choose settings to reach higher assurance scores independently of preconfigured browser or operating system behavior. One approach could be advanced Path-aware Networking (PAN) combined with *geofencing* where Web applications select Internet paths that are under a specific jurisdiction.

Leveraging content object security. Current Web security relies on end-to-end security during transmission. This poses a challenge for two reasons. First, security channels need to be established per application endpoint, and, second, it introduces security dependencies when content distribution is provided by third parties (*e.g.,* Content-delivery Networks). One way to overcome these challenges is to secure the data itself instead of the communication channel. This is known as *object security* and has found attention in IoT (*e.g.,* OSCORE) and Information-Centric Networking (ICN). In this security model, data is enhanced with cryptographic features that enable self-certification and/or self-authentication, and is protected both while in flight and while at rest.

6.5 Conclusion and Outlook

To showcase the applicability of our proposed model (see Kapitel 2) in practice, we analyzed three classes of Websites that are relevant to the public, Alerting Authorities in the US, selected emergency management organizations in Germany, and input for the Diplomatic Pulse, a search engine for official press releases of UN Member States. We measured the deployment of common technologies providing potential security features (namespaces, DNS(SEC), and Web PKI) for these Websites and embedded them in our model.

We uncovered deficiencies and discussed alternatives while emphasizing that common solutions are not necessarily technical but operational as well as political. Protecting Web-based communication, specifically in context of critical infrastructure and public safety, entails addressing operational and policy challenges on national and international levels and calls for commitment of all stakeholders from service providers to intermediate infrastructure operators and browser vendors alongside policy-makers.

Acknowledgments. We would like to thank the Innovation Cell in the United Nations Department of Political and Peacebuilding Affairs (UN DPPA) for providing us the list of sources used in the Diplomatic Pulse database. We thank Jakob Bork for helping with the verification of German Emergency Service providers.

This work was supported in parts by the German Federal Ministry of Education and Research (BMBF) within the projects *Deutsches Internet-Institut* (grant no. *16DII111*) and *PRIMEnet*.

Appendices

6.A Categorization method

Tabelle A.I: Regular expressions applied on an AA name to categorize its field of operation (in order of application).

Category	Regular expression
Military	`[^[:alnum:]]fort|^fort|army|missile|base|pfpa`
Governmental	`county|counties|city|commission|borough|town| village|parish|authority|council|government| national|aviation|correction`
Educational	`university`
Law Enforcement	`police|sheriff|investigation|patrol|intelligence|'homeland security'|'law enforcement'`
Public Safety	`911|'9-1-1'|emergency|ema|eom|ohsep|fire| safety| communication|dispatch`

Kapitel 7

A Call to Reconsider Certification Authority Authorization (CAA)

Abstract

Web security relies on X.509 certificates. Fundamental to the underlying security model is the correct issuance of a certificate, which has been repeatedly challenged in the past. This led to the development of various safeguards to prevent misissuance or to mitigate the deployment of illegitimate certificates. In this paper, we argue that legitimacy of certificates should be placed first, but that the current solution standardized by the IETF and required by the CA/Browser Forum is broken by design. We take a close look at CA Authorization (CAA) from conceptual and operational perspectives. We identify design flaws, such as implicit semantics, loose policy scooping, and lack of verifiability, as well as temporal constraints that undermine CAA effectiveness in mitigating risk of misissuance. Our CAA monitoring method identifies misissued certificates, confirmed by a major CA. Based on our findings, we discuss best practices to deploy CAA and propose modifications to improve the current specification and guide future alternatives.

7.1 Introduction

Web security relies on X.509 certificates. Fundamental to the underlying security model is the correct issuance of certificates. This trust model of the Web is challenged by unrestricted certification [86]. Certification Authorities (CA) are generally allowed to certify any arbitrary resource, *i.e.,* to bind a public key to a domain name by creating a certificate. If certification occurs without the consent of a name owner, the misissued certificate can be used to impersonate resources of that name owner. Causes for misissuance are manyfold. Rogue or compromised CAs [10], spoofed DNS records or compromised name servers [217], and malicious traffic rerouting [25], [347] have been part of attack vectors in the past.

There are two principal directions to harden the security of the certificate issuance process. (*i*) *Prevention* by enabling a CA to verify whether the issuing request is valid. (*ii*) *Mitigation* by enabling the name owner or any third party on behalf to identify and revoke an incorrectly issued certificate. Both directions have been covered in standardization and deployment. Certificate transparency

(CT) [389] makes certificates public and DNS-Based Authentication of Named Entities for TLS (DANE TLSA) [363] binds certificates to services available below a specific domain name. Prevention is specified in CA authorization (CAA) [357], which allows a name owner to note in the DNS which CA is allowed to issue a certificate.

In this paper, we argue that both prevention and mitigation are important, but that prevention should be the first-class citizen because it addresses a root cause (Abschnitt 7.2). Unfortunately, the CAA IETF standard, which is the solution the CA/Browser Forum agreed on to be mandatory, is flawed. We revisit design decisions and deployment behaviors (Abschnitt 7.5) to refine on-going standards and inform our community to guide fundamentally different approaches (Abschnitt 7.6), hoping that these insights prevent common pitfalls in the future. Our data-driven approach (Abschnitt 7.4) is based on more than 4.6M unique certificates from CT logs, which we test whether they conform to CAA policies. Our major findings read:

1. Implicit semantics overshadow expressiveness.
2. Underspecified syntax allows misinterpretation.
3. Loose policy scoping raises security risks.
4. Opaque procedures defeat reliability and trust.

7.2 Background

Certificate Authority Authorization (CAA) [357] is a DNS-based approach that allows a domain name owner to specify which CA is allowed to issue certificates for the namespace under her control. Dedicated `CAA` DNS resource records contain this information. Checking CAA records before issuing a certificate was made mandatory for CAs by the CA/Browser Forum in 2017 [300].

When a CA verifies a certificate issue request for a domain name, the CA needs to perform two steps. First, the CA needs to find DNS material that is considered for CA authorization. Second, based on the obtained DNS material, the CA checks whether it is authorized to issue a certificate for the domain name.

A CA starts searching for DNS material at the given domain name (*e.g.,* www.example.com). If no CAA records are found, the CA iteratively traverses the parent hierarchy (*e.g.,* example.com, then com) until CAA records are found or the root zone (`.`) is reached. For aliases, the canonical name (denoted by a `CNAME`) record is first checked and if no CAA records are found, the parents of the alias and not of the canonical name are visited.

A CA can verify whether it is authorized to issue the requested certificate based on the CAA model, which provides properties (also called *tag*) and values as part of a CAA record. The IETF standardized `issue` and `issuewild` to constrain issuance and `iodef` to contact the name owner in case of policy violations. Two additional tags are defined by the CA/Browser Forum [304] to verify ownership of a name via phone (`contactphone`) or e-mail (`contactemail`). The CAA specification also allows non-standard tags to introduce new semantics and functionalities. If a CA encounters an unknown or unsupported tag, it either rejects issuance if its critical flag is set, otherwise ignores the flag. If only non-critical unknown tags or only tags other than `issue` or `issuewild` are found for a domain name, it is interpreted as open for authorization to all CAs.

The values of `issue`, `issuewild`, and `iodef` are fully specified. The value of `issue` and `issuewild` contains a CA domain name (*i.e.,* 'issuer-domain-name') to identify a CA and a list of optional parameters, *e.g.,* to restrict issuance to specific account holders. CAs choose their own identifiers and list them in their Certification Practice Statement (CPS). It is possible to constrain issuance to multiple CAs or completely forbid issuance by having an `issue` or `issuewild` without any valid CA identifier. The `iodef` value must be a URL using `mailto`, `http`, or `https` as scheme.

CAA does not repel misbehaving CAs. Similar to other security mechanisms, CAA cannot prevent records from being ignored. Third-party Evaluators [357, §1], however, could audit CAs and degrade CAs in case of misbehavior, possibly leading to a healthier ecosystem in the mid-term. Nevertheless, we argue that there are further serious concerns because CAA specification [357] and deployment agreements [307] suffer from four blind spots:

1. **Implicit semantics** mask misconfigurations and reduces expressiveness of CAA policies.
2. **Boundless policy scoping** allows parent zones to define policies that overrule child zones and lead to misissuance from the child perspective.
3. **Ambiguous CA identifiers** can (unintentionally) lead to overly permissive policies.
4. **Non-verifiable and temporally unrestricted policies** make CAA-based CA auditing unreliable if not impossible.

These four properties practically defeat the primary goal of CAA, *i.e.,* constraining issuance, as well as its secondary goal, *i.e.,* enabling *Evaluators* to audit CA issuance behavior, as we discuss in the following.

7.3 Threat Model

CAA is based on the unprotected DNS, so it inherits all of its security shortcomings, *e.g.,* susceptibility to record spoofing. Furthermore, if an attacker manages to compromise the name servers for a domain name, it can modify or remove CAA policies altogether. Finally, CAA does not protect against malicious or misbehaving CAs that simply disregard CAA policies. In this paper, we consider these aspects as out-of-scope, and assume the following for our threat model: (*i*) target domain name servers and respective DNS records are intact and authoritative, and (*ii*) CAs correctly implement and respect CAA.

We consider three concrete attacks concerning spoofing, elevation of privileges, and repudiation. In our model, the actors are name owners, operators of public recursive resolver, CAs, and an active

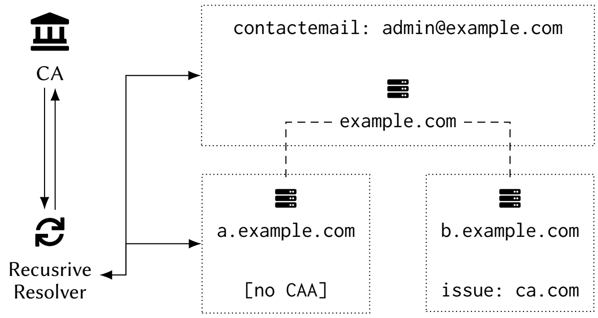


Abbildung 7.3.1: Overview of entities involved in CAA validation in a namespace with two delegations (`{a,b}.example.com`).

attacker as adversary. Abbildung 7.3.1 depicts an example. We consider an active attacker that cannot manipulate or compromise name owner infrastructure but other name servers and recursive resolvers.

Spoofing name owner. Assume an attacker that aims at manipulating CAA policies for a given domain name to authorize or deny a specific CA, or to insert its own contact information for domain validation, *i.e.,* using `contactphone` or `contactemail` tags. If the domain name has its own CAA policies, an attack is not feasible without compromising the target name servers or manipulating DNS message distribution. The prior requirement contradicts our assumptions (secure target domain name server, and good CA), the latter could be countered by enabling the (optional) DNSSEC.

If the target domain does not have its own CAA policies, a viable attack vector is to convince or compromise a name server along the path to the root zone, or even take advantage of insecure CAA policies on those name servers. In such scenario, regardless of security measures taken by a name owner, spoofing is possible. This follows from CAA specification to search for CAA records beyond a DNS zone boundaries (see Abschnitt 7.2) and does not require CAA policies for a domain name to be defined by its respective owner.

A similar case occurs for a name alias pointing to a canonical name using a `CNAME` record. Here, only the owner of the canonical name can define CAA policies for the aliased resources. The attack vector then transfers to the canonical name, *e.g.,* by targeting CDN operators that commonly use canonical names for their customers.

Two possible solutions exist for this issue. The CAA lookup strategy could be sharpened to remain confined to the actual name owner. Alternatively, a subdomain could be designated for CAA records that eliminates the recursive CAA lookup algorithm. We discuss these approaches in depths in Unterabschnitt 7.5.2.

Elevation of privilege. Assume an attacker that wants to authorize further CAs for a given domain name beyond those intended by the name owner, *e.g.,* to take some advantage of issuance, such as a weak validation procedure. To reach this goal, an attacker can take advantage of the fact that CA identifiers are not assigned in a coordinated or centralized manner (*e.g.,* through IANA), and that the CAA specification does not require uniqueness or exclusive use of identifiers. Respectively, an attacker could convince a name owner to add or use some identifier which also authorizes another, unintended CA (see Tabelle 7.5.1). We discuss this issue in details in Unterabschnitt 7.5.3.

This attack is deployment related and can be mitigated by (*i*) coordination among CAs when choosing their identifiers, or by (*ii*) provision of a public index that makes CA identifiers transparent to name owners.

Repudiation. CAs document the issuance procedure together with relevant CAA records and are even mandated to do so in cases where issuance was prevented by a CAA record. However, in case of a misissuance, these records cannot be used to prove or refute wrongdoing by the CA as DNS records are neither signed nor carry a validity period. It is impossible to assert their authenticity or validity.

If an attacker, for example, manages to compromise DNS resolvers used by a CA, or if a CA procedure suffers from a flaw, the records collected by CAs cannot be used as forensic material. This can basically be solved by mandating DNSSEC when using CAA and also obligate CAs to record DNSSEC chains along CAA records when processing a certificate application.

7.4 CAA Deployment in the Wild

We want to assess how the conceptional weaknesses of CAA reflect in operational practice and perform CAA measurements using a novel approach based on CT logs. Our measurements mimic what we expect a CAA Evaluator would do, which introduces strict requirements in terms of accuracy and flexibility. Unfortunately, little is currently known about CAA Evaluators.

Our method aims at (*i*) analyzing as many certificates as possible for each trusted CA, and (*ii*) reducing the time discrepancy between certificate issuance and the collection of corresponding CAA records. For reproducibility reasons, we restrict our measurements to public data, which prevents us from using private DNS zone files.

Certificate Collection. Recent work confirmed that all major CAs submit their certificates to at least one CT log [79]. Compared to active TLS scans (*e.g.,* using top lists), scanning CT logs provide a more complete picture of issued certificates as it also covers certificates which are not actively deployed on any server, *e.g.,* backup certificates [384]. We scan all active CT logs compliant with the Chrome CT policy to cover as many certificates and CAs as possible. In June 2024, 10 logs operated by 5 different companies were active. We use the library provided by Google [354] to retrieve certificates from the logs. For each CT log, we regularly check for new precertificates and fetch these. We collect precertificates instead of final certificates since storing precertificates is mandatory whereas final certificates are optional.

Timely DNS queries. Maintaining the same set of CAA records as observed by CAs at the time of issuance is challenging because certificates are not instantly available in CT logs after submission. When a certificate is merged and made publicly available depends on the maximum merge delay (MMD). Without access to historic DNS records, we argue that this remains the best effort in reconstructing the same CAA records as observed by the issuing CA.

For all new certificates, we extract all subject alternative names (SAN) of type DNS and expand each domain name to all its parents up to TLDs. For example, `www.example.co.uk` is expanded to `{www.example.co.uk, example.co.uk, co.uk, uk}`. We then query CAA DNS records for all names and store only certificates with at least one SAN that has a corresponding CAA record. The result is a mapping of domain names to a set of CAA records as well as several other attributes such as the source of record (*e.g.,* parent domain), provided the source is a canonical name.

To avoid stale data from third-party caches, and as recommended by RFC 6844 [356], we retrieve data directly from authoritative name servers instead of recursive resolvers.

Mapping between CA identifiers and CA names. To verify whether a certificate was issued by a legitimate CA, we need a mapping of CA identifiers (as denoted in CAA records) to CA names (as denoted in X.509 issuer field). Prior work [79], [213] and the Mozilla Common CA Database provide such a mapping. This data, however, is neither up-to-date nor complete. We extend the Mozilla list by inspecting CPS documents from various CAs.

Next, for each certificate, we consider all domain names included in the SAN and check whether CAA records comply with the CA that issued the certificate. Note that in contrast to prior work, our point of departure are certificates and *all* the included domain names and not a single domain name and the corresponding certificate deployed at a web server.

Measurement Setup and Data Corpus. We queried CT logs over a period of 2 week (2024-06-13 to 2024-06-26) and collected more than 5.4M certificates. About 85% (total of 4.6M) are unique certificates, because a single cert is logged in multiple logs as required by browsers. Among multiple instances of the same certificate, we choose the one that was first seen in any of the logs since it represents freshness. From these certificates we extract 5.8M unique domains, of which 34% are wildcards.

7.5 Potential Design and Operational Pitfalls

CAA aims to be cost-effective through low-barrier deployment requirements and simple validation processes [357, §5.2.]. This is true compared to DNSSEC [444], [446], DANE [363], MTA-STS [399], and other DNS-based security protocols. Such *simplicity*, however, can be delusive as identified in prior work [213] and discussed among CA operators [301], [302], [339]. In the following, we discuss pitfalls caused by concessions to accessibility and simplicity of the protocol, investigate how these pitfalls are reflected in current CAA deployment, and propose solutions.

7.5.1 Implicit Semantics

Problem. The CAA specification allows to express different semantics using the same syntax or implementing the same semantic using different syntaxes. For example, `issue` can be used to (*i*) constrain issuance if its value is a CA-recognized domain name, (*ii*) forbid issuance if its value is empty (*i.e.,* `";"`), malformed (*e.g.,* `"%%%%%"`), or is not associated with any CA, or (*iii*) constrain issuance for wildcard names if `issuewild` is missing. Another example is the case of unrestricted issuance by omitting CAA records altogether *if* none of the parents declare CAA records, or by only listing a tag other than `issue` or `issuewild` with an arbitrary value. It is also allowed to have contradicting CAA policies in a CAA set. For example, to forbid issuance using an empty `issue` tag alongside of another `issue` tag with a valid value.

Such implicit semantics pose two challenges. First, it hinders implementing specific cases, for example defining constraints only for FQDN names, while posing no restrictions on wildcards. Second, it prevents CAs from differentiating between intentional configurations and accidental misconfiguration (*e.g.,* by typos). Consequently, definite feedback cannot be provided to name owners.

Deployment. We observe 811 cases that permit certificate issuance and at the same time contain an empty (`";"`) `issue` tag. In additional 18 cases, malformed values (*e.g.,* `"@"`, `"\;"`) coexist with valid identifiers. In other cases, we find the same erroneous, yet syntactically correct, values used at different domain names, indicating that the name owners have copied the values without realizing its semantics. For instance, we observe 1,319 unique domain names using `0issueletsencrypt.org` (incorrectly includes flag and tag within the value) with 829 using the following exact set to issue constraints: `{0issueletsencrypt.org, comodoca.com, digicert.com, letsencrypt.org, pki.goog}` (note the correct value for Let's Encrypt). Although the CAA specification explicitly ignores such cases [357, §4.2.], specifically for values that hint at a typo rather than an intentional malformed value, we diagnose that the semantic is not fully understood by name owners and can lead to unwanted results.

In our dataset, there are also cases of unintentional lifting of constraints due to unknown tags. Two domains use `isssue` tag (note extra "s") with a value of `letsencrypt.org`, which evidently was meant to constrain issuance, but in practice lifts all restrictions. Another domain uses the aforementioned erroneous tag alongside the correct `issue` tag. Similarly, there are cases where `issue` was misspelled (*e.g.,* `issed`) next to a valid `issuewild` records, thus only limiting issuance for wildcard domains. We also observe confusion by an issuer adding the CA identifier only in `issuewild` but not the `issue` tag, leading a CA to consider itself to be authorized to certify FQDNs despite CAA constraints. We reported this incident to the CA (GoDaddy), which confirmed the failure[1].

Solution. First, the semantics of each tag must be uniquely confined, *i.e.,* `issue` is limited to FQDN domains, `issuewild` only applies to wildcard domains, and existence of (or lack thereof) one does not impact the other. Second, a constant value should be standardized to denote a no-issue policy instead of empty or malformed values. This way typos (as discussed below) or other unintentional errors are explicitly detected and not open to misinterpretation. Finally, contradictory policies should be considered as errors.

7.5.2 Boundless Policy Scoping

Problem. The DNS namespace is partitioned in zones. Each zone comprises a set of authoritative data for all the names in its namespaces [410]. A zone owner can further delegate parts of his namespace

[1] See `https://bugzilla.mozilla.org/show_bug.cgi?id=1904748`

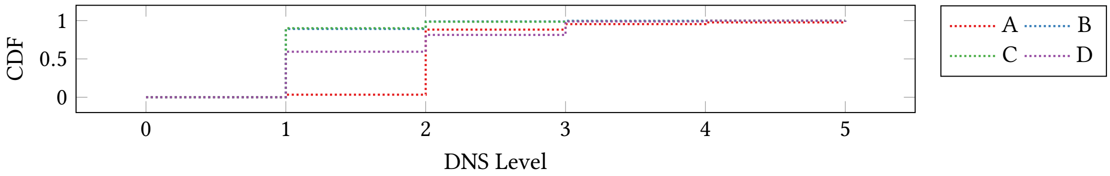


Abbildung 7.5.1: Distribution of domain names by DNS level at which relevant CAA records were found. Group A are delegated domains w/o CAA, B are alias domains w/o CAA, C are alias parents w/ CAA, and D are alias domains w/o CAA but with an alias parent with CAA.

to others and create new zones. Delegated zones are independent of their parents so that the parent zone cannot define records for names in a delegated namespace without retracting the delegation.

The CAA specification implicitly breaks this authoritative barrier by allowing CAA records to be inherited from *parents*, which can be under a different authority. Consequently, the attack surface is expanded beyond the owned name servers. For instance, if parent and child define contradictory policies, or there is a parent applying less strict policies, an attacker can try to suppress or spoof all CAA queries in the query chain until the desired level is reached. Also the attacker could compromise a name server in the hierarchy to impact the entire namespace. It should be noted that not only constraining policies (*i.e.,* `issue` and `issuewild` tags) are inherited, but also contact information that are used for domain validation [307, Appendix A], *e.g.,* as specified by `contactemail` tag [303]. This allows an entity to be granted a certificate for a domain name which it does not control.

A similar situation arises for aliases denoted by a `CNAME` record, which do not allow other records to be present except DNSSEC crypto. Here, a CA first asks the authority of the canonical name for CAA records, implying that only a potentially different authority can define CAA policies for a domain name that is not delegated. The zone owner is left without any means to control CAA certification control.

Deployment. More than 777k domain names in our dataset are subject to CAA policies defined by a separate zone. We classify these into four categories (ordered by frequency of occurrence):

A *Delegated domain w/o CAA.* In more than half of all cases, the CAA is defined by a parenting zone for a subdomain that is delegated.

B *Alias domain w/o CAA.* About 45% of the cases show alias domains with CAA records sourced from a parent, *i.e.,* the authority of the domain name and CAA source differ.

C *Alias parent w/ CAA.* 1,672 instances inherit CAA and the relevant CAA records originate from a parenting domain name which is an alias for a domain name in another zone.

D *Alias domain w/o CAA and alias parent w/ CAA.* As a special instance of the former case, the domain name is itself an alias without CAA records, but we find a parent that is also an alias and its canonical name (from another zone) provides effective CAA records (1, 080 instances).

In total, there were 108 domains for which a parent in another zone lists email addresses. If the CA supports domain validation using CAA `contactemail`, the parent can apply and successfully retrieve a domain validated (DV) certificate for such a subdomain in another zone. In all instances it is another authority that defines CAA policies for a namespace. Abbildung 7.5.1 depicts the distribution of DNS hierarchy levels at which CAA records were found.

Solution. The effective scope of CAA records must be confined to a given DNS zone. A naive approach would adapt the algorithm used to find relevant CAA records [357, §3.] to stop at the zone apex and not at the DNS root. This, however, does not solve the problem of aliases. The conceptually clean solution taken by DANE [363] dedicates subdomains for `TLSA` records. Thus in case of canonical names, the referred authority has no control over DANE records. Analogously, CAA records should be listed under a designated `_caa` subdomain, *e.g.,* `_caa.example.com`.

7.5.3 Ambiguous Identifiers

Problem. The value of `issue` and `issuewild` tags is called 'issuer-domain-name' and identifies the CA which owns that name. This identification, however, is not unique as the same domain name is also accepted by any 'party acting under the explicit authority of the holder of the issuer-domain-name' [357, §4.2.]. Such authorized party can be a subsidiary, a reseller, or even another CA organization. Ambiguous identifiers defeat the purpose of CAA by preventing name owners to authorize a specific CA. Two cases are plausible. First, a name owner unknowingly authorizes a third-party CA. This happens when independent CAs accept the same identifiers, *e.g.,* DigiCert also accepts Amazon identifiers [320] as it vicariously runs some intermediate CAs for Amazon. Second, a name owner unknowingly authorizes all CAs running on the same infrastructure. This is the case for resellers, managed PKIs and alike. For example, DKB, a German bank, has its own dedicated CA certificates[2] managed by DigiCert, but lacks a dedicated identifier. Hence it uses 'digicert.com', which authorizes all DigiCert subsidiaries, such as Thawte or QuoVadis.

Deployment. Without additional information, it is not possible to detect whether name owners are aware of identifier ambiguities. In our dataset, nearly every fifth CAA source redundantly lists `comodoca.com` and `sectigo.com` as values for `issue`, whereas Sectigo CPS explicitly states that the former identifier is deprecated. We also observe up to eight different CA organizations that share the same identifier. Note that the number of CAs sharing identifiers is higher than what we observed in our measurements. Alone `sectigo.com` is accepted by Gandi (FR), ZeroSSL (CH), GEANT Vereniging (NL), eMudhra Technologies Limited (IN), and at least 17 other CAs from around the world that use Sectigo infrastructure.

[2] See `https://crt.sh/?CN=DKB+CA`

Tabelle 7.5.1: The CA strings that appear in more than 10% of the CAA records in our dataset as well as their occurrence as the *only* CA string.

CAA String	issue		issuewild		CA Count†
	Overall	Single	Overall	Single	
`letsencrypt.org`	89.96%	37.62%	87.26%	5.68%	1
`digicert.com`	38.85%	0.40%	87.26%	0.93%	3
`pki.goog`	36.21%	5.90%	72.22%	8.46%	1
`comodoca.com`	27.84%	0.11%	67.16%	0.03%	6
`globalsign.com`	26.47%	0.02%	44.66%	0.12%	2
`sectigo.com`	25.00%	0.05%	33.48%	0.21%	8
`amazon.com`	11.24%	<0.01%	1.92%	0.04%	1
`godaddy.com`	8.25%	<0.01%	<0.01%	<0.01%	2

† Count of unique CAs (by Subject Organization) in our dataset that match the respective CAA string.

Our data also revealed cases of misissuances (confirmed by the CA GoDaddy), in which all strings matching a pattern were accepted instead of a fixed domain name[3]. Tabelle 7.5.1 gives an overview of the most frequently used identifiers and shows how many organizations match the same identifier.

Solution. The identifiers should reference the CA with whom subscribers directly enter a business relationship instead of the infrastructure operator (*cf.,* the example of Sectigo above). Furthermore, an authoritative and publicly accessible mapping of CAA identifiers to CAs must be provided so that (*i*) name owners can verify the reach of each identifier, and (*ii*) any other identifier be considered invalid. Although Mozilla maintains such a mapping in its 'Common CA Database' [471], it is neither authoritative nor complete [213]. The protocol registry of Internet Assigned Numbers Authority (IANA) is a potential alternative to maintain such a mapping, as was proposed for CAA tags [357, §4.1.].

7.5.4 Non-verifiable and Temporally Unbound Policies

Problem. The CAA specification describes Evaluators as third-party auditors that may use CAA records to detect policy violations [357, §1.]. At the same time, it warns that DNS records may have changed at the time an Evaluator retrieves them and are effectively unreliable in discovering misissuances. In its current state, CAA can only be verified by the name owner and the CA at the time of certification. Even at that time, there is no definitive way for the CA to make sure that CAA records have not been manipulated or suppressed on-the-fly. This issue has its roots in missing (*i*) validity timestamps, and (*ii*) data-origin and integrity verification features.

Deployment. To the best our knowledge, there are no publicly active CAA Evaluators. Using the measurements in this paper, we take the role of an Evaluator. Our proposed method in Abschnitt 7.4 reduces the temporal gap between CA query of CAA records and our own queries. Nonetheless, a time discrepancy as large as the maximum merge delay of CT logs (see Abschnitt 7.4) remains. Abbil-

[3]See `https://bugzilla.mozilla.org/show_bug.cgi?id=1904749`

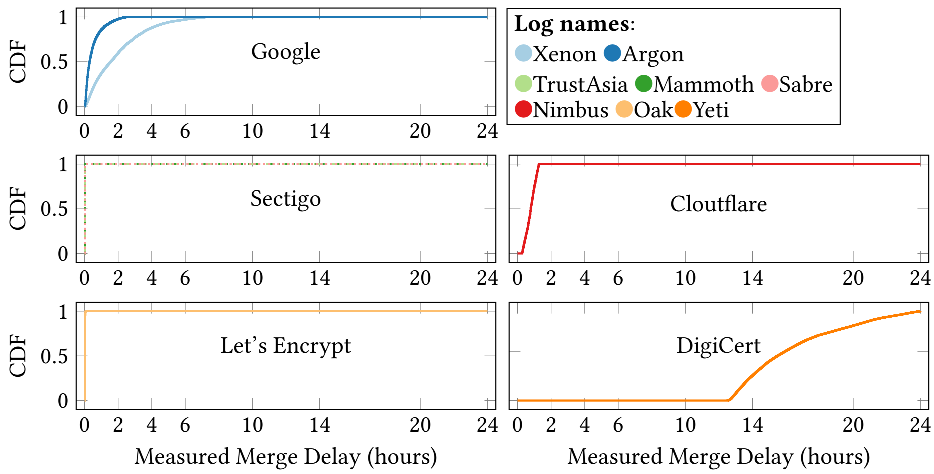


Abbildung 7.5.2: Distribution of measured CT merge delays

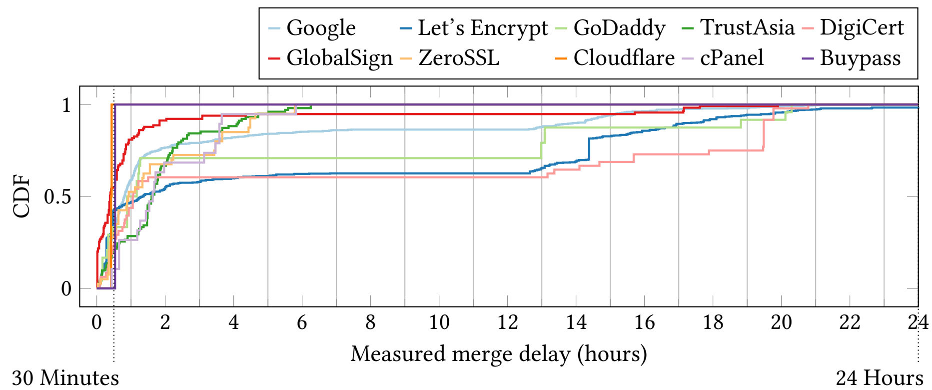


Abbildung 7.5.3: Distribution of measured merge delay for certificates with at least one mismatching CAA record

dung 7.5.2 depicts this merge delay for various CT logs as measured by our scanners. To reduce the risk of manipulated data, we query only authoritative servers using our own recursive resolvers.

To understand the impact of time discrepancy we take at look at cases of mismatching the CAA policies in our dataset. Given that (*i*) we observe only a small fraction of certificates that mismatch their corresponding CAA records, and (*ii*) we are unable to reproduce such cases independently, we conjecture that these are false negatives due to modified DNS records. Abbildung 7.5.3 shows the distributions of these cases with respect to the measured merge delay. We can conclude that any time discrepancy larger than 30 minutes significantly increases the chance of incorrect CAA validation. Regardless of merge delays, CAs can also use CAA records as stale as 8 hours [307, §3.2.2.8], so that in the worst case the temporal discrepancy amounts to $8h + MMD$. Even Sunlight logs[4], a new generation of CT logs that remove merge delays, can fill this gap. Without fine-granular historical DNS data or designated validity timestamps for CAA records, auditing of CAA remains unreliable.

Solution. This problem can be addressed indirectly by enabling DNSSEC or directly by adding validity timestamps to CAA records and signing them. The former approach infers validity of a record from validity period of covering DNSSEC signatures (see RFC4030 [446, §3.]), while data-origins can be validated using DNSSEC keys for that zone. In the latter approach CAA records are extended to (*i*) carry a validity timestamp, and (*ii*) be cryptographically secured, *e.g.,* through a digital signature. This also allows to determine which entity defined a set of CAA policies (data-origin authentication). A signed and temporally constrained CAA record can retrospectively and securely be verified, *e.g.,* by an Evaluator.

7.6 Lessons learned

CA Authorization was designed as a preventive measure against certificate misissuance but falls short due to design decisions, which hinder proper deployment. We highlighted pitfalls and proposed solutions. In the following, we report about lessons learned from a decade of CAA usage and discuss aspects that should guide the design of related protocols in the future.

DNS is a fragile basis for security protocols. DNS is susceptible to attacks impacting its integrity [288], [156], [157], [485]. Any service that relies on the unprotected DNS inherits these vulnerabilities. For example, it has been shown how plain DNS can impact domain validation and consequently lead to certificate misissuance [217]. Although aware of this issue, CAA does not provide an alternative and only recommends using DNSSEC [444]—without making it mandatory—to mitigate threats targeting DNS [357, §5.]. DNSSEC, a set of security extensions, brings integrity and origin authentication alongside authentication of name nonexistence to DNS [444, §3.]. Consequently, DANE mandates the use of DNSSEC. DNSSEC, however, is not widely deployed by name servers and resolvers [45], [186], [188].

[4] See `https://sunlight.dev/`

Solutions orthogonal to DNSSEC have been developed to mitigate DNS security weaknesses, among them Multi-Perspective Issuance Corroboration (MPIC), and trust on first use (TOFU). MPIC verifies domain control from multiple vantage points across the Internet to reduce the risk of retrieving manipulated data. It has been proposed for integration into CA/Browser Forum Baseline Requirements as part of domain validation procedures. TOFU-based approaches rely on a first intact and uncompromised data exchange to establish security parameters for upcoming interactions. An example is MTA-STS [399], which is used against downgrade attacks in STARTTLS protocols (*e.g.,* deployed by mail servers).

Trust must be derived from frequent public audits. To measure effectiveness of a security protocol, it must be auditable. Experiences from the Web PKI show that formal and private audits (*e.g.,* WebTrust or ETSI) are rather blunt tools for preventing or detecting certificate missiuances [10], [405]. In contrast, providing the public with adequate means to perform audits is effective as the success story of CT logs reveals [64], [74], [89], [383].

CAA validation by CAs cannot be audited. CAs do not publicly log CAA record consumption nor even provide them upon request [213], and even if they did, they could not rule out manipulation or spoofing. For DNS-based protocols to be auditable, some form of content object security is required for the underlying resource records to be temporally and spatially decoupled from the name owner—as provided by DNSSEC. Assurances of integrity and data-origin authenticity are covered by the DNSSEC model and possible tampering can be detected. Once object security is reached for DNS resource records, decoupled data can be distributed over established channels, for example as extensions to the Online Certificate Status Protocol (OCSP) [448] or TLS (*cf.,* TLS DNSSEC Chain Extension [327]).

Potential victims must be able to verify measures. CAA has been designed as a light-weight approach to protect against certificate misissuance. It enables the CAs—and only the CAs—to verify their procedures at the time of certificate issuance.

Potential victims of illegitimate certificates, however, are resource owners and service users. An online financial institution, for example, looses reputation if its customers get fooled by an illegitimately certified fraud service. Neither the legitimate service provider nor its users can reliably detect cases of misissuance, which leaves them blind with respect to the efficacy of the CAA protective measures. This blindness of potential victims turns CAA into a toothless security scheme. Instead, CAA merely serves as a detection aid for configuration errors in toolchains of well-behaving CAs.

7.7 Conclusions and Outlook

A reliable and trustworthy certificate issuing process is utmost important to enable secure Internet services. In this paper, we critically revisited CAA, a currently popular DNS-based Internet standard for preventing incorrect certificates. The CA/Browser Forum mandates the deployment of CAA. We found the design of CAA flawed to an extend that prevents proper deployment and proper auditing. Given the importance of the security objectives and that some problems have been found in the wild already in 2018, the lack of progress leaves us puzzled.

We identified several options that require only minor changes to improve CAA in future development, and highlighted principle design choices that should guide the design of improvements. With these insights we hope to have shed light on how to build a more trustworthy certificate ecosystem in the near future.

Ethical Concerns. This position paper draws attention to a security approach that should be improved with respect to design and deployment. When we found incidents in real deployments, we implemented a responsible disclosure policy to solve the problem with the concerned CA before making the incidents public.

Teil III

Suitability of Information-centric Networks

Kapitel 8

On Economic, Societal, and Political Aspects in ICN

Abstract

Information-centric networking (ICN), as an antithesis of host-centric networking, denotes a paradigm shift in communication networks. It introduces names to the network layer and favors de-localized content instead of addresses and hosts. ICN is an attempt to design a network tailored to demands of users who only care about data. The simplicity of this basic premise, however, turns out to be rather deceptive; a pitfall in waiting on the path of ICN to wide-scale deployment. Surely users care about data, but they also care about trust, accountability, private communication, and everything else that the current Internet provides beside mere content. This paper is a first attempt in (*i*) pinpointing the missing non-technical aspects that are crucial to success of ICN as a viable replacement for the Internet, and (*ii*) showing how the realization of these aspects might differ from their current manifestation on the Internet. To study the ripple effect caused by the introduction of names on the network layer, we first introduce a generic abstraction of ICN in terms of three fundamental name bindings. Based on this model we then try to understand how the inevitable reincarnation of a number of economic, societal, and political issues in ICN is going to be different from the current state of affairs on the Internet.

8.1 Introduction

Information-centric networking (ICN) denotes a paradigm shift in networking that aims to address shortcomings of host-based IP networking. The unifying premise of different ICN approaches is that (*i*) Internet's primary use is de facto content distribution [311], and (*ii*) users are only interested in content and barely care about its location within the network [115], [125] (*what* vs *where*). In ICN data is decoupled from its host (more specifically its producer) through independent network-wide labels that cater for content retrieval regardless of its actual location. Respectively, instead of addressing hosts and switching packages among them the network layer directly acts on named data objects (NDO). The spatiotemporal decoupling of data from its producer also enables in-network caching, mobility, and multi-homing by design. Proponents of ICN argue that this new paradigm can solve a number of technical and non-technical issues [244] which have been plaguing the IP networking partly due to the dual purpose of IP addresses, *i.e.,* both as host names and interface addresses [185]. All the same, skeptics have also expressed their concern about practical advantages of ICN and raised suspicion about its superiority to existing solutions [94], [193].

In the past decades the idea of information-centric networks has been accompanying the network research community in form of overlay networks, *e.g.,* TRIAD [311], up to ambitious network layer replacements of IP, *e.g.,* Named-data Networking [115] (NDN). Yet, there are still no wide-scale deployments of ICN, let alone any concrete implementation that could replace the Internet as we know it. The reason for this might be the fact that the Internet is not a mere technical construct but a sociotechnical ecosystem that composes an indispensable basis of our economic, social, and political life [351, Chapter 1]. So any radical change to the Internet, however technically reasonable, will face the invisible inertia of Internet's societal, political, and economic forces that have been shaped by and been shaping the Internet landscape since its conception. Although this has not gone unnoticed by ICN researchers, the study of non-technical aspects are generally limited to specific conceptualization and implementation of ICN [61], [382], [223].

In this position paper, we argue that acceptability (if not success) of ICN directly depends on its capability of adaptability and integration within the broader context of the Internet, *i.e.,* into the economic, societal, and political contexts. Based on a generic abstraction of ICN we identify and discuss a number of fundamental enablers that needs to be addressed before any ICN approach can reach widespread deployment beyond experimental and research settings. Our basic premise throughout this paper can be summarized as follows: by pushing names, specifically human-meaningful names, from application layer further down to the network layer, new challenges are invoked which traditionally have not been the concern of network layer protocol designers.

The rest of this paper is structured as follows: after providing a primer on ICN, introducing a generic abstraction based on name bindings common to all ICN approaches, and giving an overview of naming schemes among select ICN implementations (**§8.2**), we move to our discussion regarding non-technical ramifications of introducing names on the network layer (**§8.3**). We discuss how the paradigm shift of ICN also causes shifts in power structures, trust relations, accountability, and private communication over ICN. Finally, **§8.4** concludes this work and draws an outlook for future work.

8.2 Information-centric networking: a Primer

The basic premise of ICN is that users are only interested in data and care less about where it is located [311], [115], [125]. Respectively, the common denominator of all ICN approaches is to name data instead of hosts (which might provide data and services). In contrast to IP addresses, which are host identifiers as well as locators (addresses)[1], ICN names are location-independent. An ICN name is an indirection, which eventually resolves to a data object. A name can also be bound to other resources, *e.g.,* a public key or a (real-world) identity, just as a domain name can be bound to different resource records.

[1] see Ohlman [185] for pitfalls of ID/locator entanglement and how ICN can help.

Before discussing implications resulted from the paradigm shift of ICN, we start with a general overview of fundamental name bindings in ICN and how various ICN approaches define and secure these bindings.

8.2.1 Fundamental name bindings

Any ICN approach needs to define at least three name bindings (and respective resolution and authentication methods) to cater for name-based data publication and retrieval:

Name to location: maps a name to address(es) where data is available.

Name to data: maps a name to actual bits.

Name to owner: maps a name to an authorized producer.

These bindings allow allocation of names to respective owner, assignment of data with names, and name-based publication and retrieval of data over the network. In the following we will focus only on these binding that evolve around names and consider other derivative bindings out of the scope of this work.

A name binding is considered to be secure if it can be authenticated. Depending on the binding, authentication can succeed through (*i*) self-authentication, (*ii*) self-certification, or (*iii*) through *notarization* by trusted third parties (TTP). A self-authenticating binding is a type of self-sufficient mapping that requires only local information for its verification [83]. For example, using secure hashes as names defines a simple name to data object mapping that can be authenticated locally given that the used hash function is known. Self-certifying bindings rely on digital signatures for authentication given that the public-key of the signer is known beforehand [83], [93]. Finally, authentication can be relegated to a trusted third party that vouches for the correctness of a binding (notarization).

8.2.2 Naming schemes

Various ICN approaches define different naming schemes with different name characteristics as summarized in Tabelle 8.2.1. The decision for specific naming schemes is definitive for how the aforementioned naming bindings are realized and authenticated and expresses a trade-off between contradictory desired attributes. For example, a self-authenticating name provides intrinsic name to data authentication without needing extra namespace management but is non human-meaningful and requires an additional directory or mapping service, *e.g.,* a name resolution server, to translate expressive names into such names. Finally, it should be noted that not all bindings can be self-authenticating or certifying. In the following we discuss how select ICN approaches define and authenticate aforementioned bindings.

Name to Location. On the Internet it is a matter of application layer to map a host name to an IP address, which identifies the host and its topological location within the network. And it is the

Tabelle 8.2.1: Naming and name bindings in various ICN architectures.

	Assignment			Bindings: name to …			
Name	Deterministic	Arbitrary	Structure	Location	Data	Owner	Name syntax*
DONA [125]	✓	✓	Two-tuple	NBR	(Self-auth)	Self-cert	`<principal-pubkey-digest, label>`
PSIRP [386], [468] PURSUIT [476]	✘	✓	Two-tuple	NBR	Self-auth	Self-cert	`<scope-ID, app/rendezvous-ID>`
CONET [57]	✓	✓	Two-tuple	NBR/NBF	Self-auth	Self-cert	`<namespace, name>` (default: `<principal-pubkey-digest, data-digest>`)
CCN / NDN [115], [265]	✘	✓	Hierarchical	NBF	Digest	Signature	`</label>+`
COMET [310]	✓	✘	Flat Hierarchical	NBR	Unspecified	Unspecified	`<alphanumeric-content-ID>` `<domain>@<owner></label>+`
MobilityFirst [249]	✓	✘	Two-tuple	NBR	Self-auth	Self-cert	`<principal-pub-key-digest, data-digest>`
Green ICN [287]	✘	✓	Hierarchical	NBR/NBF	Digest	Signature	`n2n://<name-realm>:<ID>/<content-name>`
NetInf [333]	✓	✘	(Two-tuple)	NBR	Self-auth	(Self-cert)	`ni://[authority/]<alg>;<digest>[?[tag=value]*]`
UMOBILE [404]	✘	✓	Hierarchical	NBF	Digest	Signature	`</label>+ [#tag]*`

* Syntax notation: <> = mandatory, `[]` = optional, `+` = at least once, `*` = zero or more times

responsibility of the name owner to provide for this mapping, *e.g.,* using DNS. On the contrary, ICN explicitly entrusts the network layer with locating data based on its name.

The burden of mapping names to locations on a global scale is addressed in ICN either by introducing (global or local) name resolution services for name-based routing (NBR), a naming scheme that enables name-based forwarding (NBF) without requiring a resolution service, or a combination of both methods. In case of NBR, the most popular solution is to assign names to an administrative entity and let it provision for mapping of names under its control to respective locations or at least to next known responsible resolution service. Examples are principals and hierarchical resolution handlers (RH) in DONA, scopes and the *rendezvous system* in PSIRP mapping rendezvous IDs to forwarding IDs [468], the two tier (global and local) rendezvous system of PURSUIT [477], namespaces and local resolution tables alongside a global *name-system* in CONET to map names to the next hop [57], *domains* and a hierarchical name resolution service in COMET [310], *name-realms* and a centralized distributed NRS in Green ICN which maps these to *network-realms* [287], and finally globally distributed and/or locally decentralized NRS in NetInf which maps hashes to *routing hints* [51], [385]. An alternative is to provide a global and distributed resolution service such as Auspice [222] for MobilityFirst which is a decentralized and federated name service.

Name-based forwarding approaches, on the other hand, require network nodes to maintain some sort of state within the network to be able to perform forwarding without any extra information except the name itself. CCN/NDN is the most prominent example of ICN using NBF with its hierarchical naming scheme that enables prefix aggregation and caters for better scalability.

Name to Data. Name to data binding concerns the resolution of a name to actual bits. For local names the context of this mapping is also local so that the same name can be resolved to different payloads in different local contexts. Global names, on the other hand, require consistent and unique resolution regardless of local context (referential transparency).

A popular approach to realize this binding in ICN is to identify data by its digest. For example, DONA [125], CONET [57], MobilityFirst [249], and NetInf [51], [333] define naming schemes which embed the secure hash of data in its name, thus providing intrinsic name to data bindings. Others, define this binding indirectly over name to owner and data to owner bindings. NDN/CCN [115], [265], for example, mandate all data to be digitally signed (data to owner) and if signer is authorized to use that name (name to owner), the name to data binding is indirectly authenticated.

Name to Owner. Name owner denotes an entity authorized to produce data under a specific name, *i.e.,* create name to data bindings. The name to owner binding is, among others, a perquisite for data-origin authentication (when combined with name to data binding). DNS is an example of global distributed namespace management system that delegates hierarchical domain names to their respective owners. Authentication of domain name delegations succeed through DNSSEC.

Among ICN approaches, DONA [125], CONET [57], and MobilityFirst [249] propose to embed hash value of name principals' public keys into names and sign data with respective private keys for binding authentication (self-certification). Other approaches such as COMET [310] or Green ICN [287] also embed owner identifiers into names but do not specify how binding authentication should succeed, *i.e.,* how to avoid and recognize unauthorized name to owner bindings. NDN explicitly regards [421] namespace management out of the scope network design comparing it to how IP address space management is separated from its architectural design[2].

8.3 ICN Implications beyond Network Layer

The paradigm shift of ICN, *i.e.,* naming data on the network layer, might be an intuitive, simple proposal, yet its implications beyond the technical sphere are far from non-trivial. In the following, we discuss some of these practical implications with respect to the aforementioned name bindings introduced in Abschnitt 8.2.

8.3.1 Names as commodity

As the very existence of yellow pages or the Internet's phone book, DNS, suggests, human-meaningful and expressive names are preferred by people over numeric identifiers, *e.g.,* IP addresses, or even worse, long random digests of data or public keys. As more semantic is interpreted into such names, these presumably rather functional identifiers gain on value and turn into commodities and as such require regulation and protection. Domain names, for example, are protected by trademark laws and national or even international regulations while the most expensive ones worth million times more than their face value [465], *i.e.,* yearly registration fees.

With names at its core, ICN needs to address the issue of namespace management as a perquisite for wide-scale deployment [83]. Namespace management regards on the one hand the allocation of names to owners and name assignments in general, and on the other hand addresses organizational aspects such as dispute resolution and name abuse. Even approaches using deterministically assigned, human unexpressive names, *e.g.,* hashes as names, would sooner or later face the need for a global directory that binds human-meaningful names to underlying ICN names. Some proposed approaches, *e.g.,* COMET [310], already take this necessity into account by integrating a global name directory binding names to underlying identifiers. Others, *e.g.,* DONA [125] and MobilityFirst [249], suggest to use services such as search engines as external directories to discover unexpressive public key hashes, thus, further delegating the problem without solving it.

If we assume that in a future internet ICN names are going to take the role of domain names, we can conclude that they will face similar challenges faced by the DNS ecosystem regarding governance, conflict resolution, and regulation both on national and international levels. If so, we should ask ourselves if growth, establishment, and acceptance of ICN names can be accelerated by an integration in the existing DNS ecosystem as de facto the naming infrastructure of the Internet? Even if the merits of hierarchical naming scheme of DNS is subject to debate, *e.g.,* as MobilityFirst [249] explicitly rejects the

[2] We consider this as a false analogy. The hierarchical expressive names of CCN/NDN are more related to domain names than IP addresses.

single rooted hierarchical namespace and trust model of DNS, the namespace management regardless of the naming scheme can be delegated to ICANN as an established multi-stakeholder organization of Internet governance. This way there would be no need to convince stakeholders to take part in yet a new organization and introduce ICN-tailored policies, regulation, laws, etc.

8.3.2 Power structures in ICN

Castell [36] argues that in a networked society the most crucial form of power is in the hands of those who shape the network and its goals, *programmers*, and those who control the inter-network connections, *switchers*. In other words, *network-making* power regards the capability of involved actors in exercising their will in constructing and manipulating a network. The paradigm shift of ICN causes a respective shift of power along various dimensions which are elaborated in the following in terms of the three aforementioned bindings (Abschnitt 8.2): name to location, name to data, and name to owner.

Power over data retrieval. On the Internet, specifically web communication, it is the name owner that controls how a domain name is mapped to an IP address by manipulating entries of the respective authoritative name servers. Name owner could also control further aspects of name to location bindings, *e.g.,* through dynamic bindings for load balancing purposes. On the other hand, users can exercise control (to some extent) over name resolution through their choice of DNS resolvers. For example, as the Twitter was blocked in Turkey in 2014, people used Google's public DNS to circumvent this DNS-based censorship [460].

In ICN the name to location resolution is delegated to the network and as such remarkable power is shifted from name owners and consumers to the network operators. Beside positive consequences of this power shift, *e.g.,* serving data from the nearest cache, further implications on censorship and net neutrality are non-trivial. For ICN approaches with structured naming schemes (see Tabelle 8.2.1) censorship would be as easy as preventing name resolution based on principal IDs (to censor specific data owners), name-realms, namespaces, rendezvous IDs (to censor complete networks), or to filter specific labels or tags (to censor topics) [13], [319], [134], [242]. Similarly, it is possible to realize a *domestic internet* (see Iran's National Intranet [436] for example), by whitelisting only conforming data producers, namespaces, or topics. Finally, same mechanisms can also be applied to discriminate and prioritize specific types of data flow.

Power over data naming. Names and naming are (if of public relevance) in general subject to regulation, *e.g.,* natural or legal individuals' names. On the Internet, naming regulation is within the authority of namespace owner. ICANN, for example, decides on introducing and delegating new generic TLDs (gTLD) while each TLD owner defines rules on how its namespace is structured and used. For example, the `.us` namespace has more than 3,000 reserved names unavailable for public registration [424] and it is one of the most rigorously structured TLDs with a well-defined structure even on second, third, and forth level domains [315]. In contrast to ICN names, however, there is no direct correspondence between a domain name and the content associated with it so only in very rare cases namespace owners impose content regulations for their delegated names, for example in case of .gov domains [326].

In ICN as the name to data binding might reveal what type of content is carried under a specific names, namespace owners are practically granted fine-granular control over content available within the network. This opens a new field for exercising power just by controlling how and if a specific piece of datum can be named. Blacklisting or whitelisting names would consequently decide on existence of data itself on the network.

Power over name ownership. As names are valuable commodities within ICN, any ICN namespace with a claim of universality will face competition and possible disputes among a wide range of stakeholders. Whereas on the Internet a domain name is rather a convenience, on ICN access to names is surely the most crucial aspect of ICN which practically determines participation on the network: it is impossible to publish before being granted a name. In this sense, the power to allow participation through assignment of names or sub namespaces to owners is in the hands of namespace owners and managers. The biggest challenge, respectively, faced by ICN is to regulate ownership within the global namespace.

The history of DNS and emergence of ICANN as a governing body over the global domain namespace and the power struggles between its stakeholders such as industry and national governments can be a prognosis about the dynamics within the future namespace ecosystem of ICN. ICANN, for example, has been criticized for lack of transparency and its oversight over the root name servers (through IANA) has been considered as a threat to sovereign control of nation states over their respective country code TLDs (ccTLD) [104]. The fact that ICANN is a private and largely unregulated organization that controls the most critical resources on the Internet [351, Chapter 5] is also a point of contention regarding its oversight legitimacy and its power over the Internet a global public good.

8.3.3 Information-centric trust

There is remarkable discrepancy in conceptualization of trust in the technical community and other disciplines[3]. The technical community generally conceptualizes trust in relation to security, *i.e., trustworthiness as security, or trust through security* [180]. Respectively, trust management is formulated in terms of security policies, services, and mechanisms that provide means to assess trustworthiness of a system in terms of reliability, correctness, etc. Here we argue that security alone is an indicator of trustworthiness but does not suffice to establish *trust relations*. Self-signed certificates, for example, can be used for authentication, confidentiality, and data integrity verification, yet are widely considered as not amounting to trust relations (despite strong security assurance) as anyone can simply generate one. We consider a certificate trustworthy only when it's notarized by a trusted party. In this sense, if a trusted third party (TTP) vouches for an entity by signing its certificate, that entity is then considered trustworthy because there exists a trust relation with that TTP in the real world and not because digital signatures provide high security assurances. Here we limit our analysis to trust issues related

[3]See Rousseau et al. [209] for a widely accepted cross-discipline definition of trust.

to trustworthiness of ICN name bindings in terms of both technical and non-technical assurances that amount to trustworthiness and postpone the trustworthiness discussion regarding routing, caching, etc. for future work.

Trustworthy data discovery and delivery. How can a consumer trust an information-centric network with correct resolution of name to location bindings. The short answer is: it can not. The long answer is by verifying name to data bindings. Recall that the basic premise of ICN is that users are not interested in data storage location and are only interested in data itself. Respectively, from a user perspective, it suffices to authenticate name to data and possibly name to owner bindings (see bellow) to verify authenticity of data regardless of network's discovery and delivery strategies.

From the perspective of network operators, however, targeting authentic location of a given name remains an open challenge. On the one hand the overhead of verifying authenticity of data before delivering, just a user might do upon retrieval, is too high to be imposed on the network nodes, and on the other hand lack of network level data authenticity verification can lead to attacks such as cache poisoning by discovering and caching bogus data.

Trustworthy data provision. Trustworthy data provision can be defined as trustworthy correspondence of names and data through authentication of name to data bindings and respectively the capability of detecting possible name spoofing, *i.e.,* binding of bogus data to a name.

If the name to data binding is not intrinsic to the name itself, *i.e.,* self-authenticating or self-certifying, additional measures are required for name to data authentication. Most commonly, this binding is implicitly authenticated through name to owner bindings. In this sense if a producer is authorized to publish under a name, and the data under that name is bound to the respective producer, *e.g.,* through a digital signature, the name to data binding can be considered as authentic. This, however, presupposes authenticating producers which in turn depends on how name to owner bindings are realized (see bellow).

Trustworthy namespace management. Trustworthy namespace management is based on allocation of names to legitimate owners and catering for means to protect name ownership, on the one hand, and provisioning secure means to verify ownership on the other hand. A name or namespace owner is a real-world entity, *e.g.,* natural or legal person, that first needs to be bound to its respective digital representation, *e.g.,* public key, before name to owner binding succeeds. For self-certifying names, where the public key of producer (or its digest) is embedded in the name itself, name to owner authentication simply succeeds by authenticating producer's signature. In other cases, where this binding is not intrinsic, additional infrastructure, *e.g.,* a public key infrastructure (PKI), is required to define and vouch for correctness of such bindings. In either case an additional layer of assurance can be provided for by enabling identification, *i.e.,* by binding public keys to real-world identities. This way the trustworthiness of name owner itself can be evaluated based on existing trust relations in the real world (*e.g.,* name owner is Fox News, I don't trust Fox News, ergo I don't trust this report!).

It should be noted that public-key cryptography presupposes a non-trivial *trust bootstrapping phase* during which public-keys of trusted entities, *e.g.,* trust anchors, are retrieved over a secure channel other than the network itself. The bootstrapping phase solves the chicken or the egg dilemma of asymmetric cryptography: if public-keys are used to sign and secure data transmission over the network, how can these keys be securely transmitted before they can be used to secure the same channel? Common practice is to deliver these keys over separate channels, *e.g.,* as part of an operating system or browser. Another equally important question is who's public key should be included as trustworthy during the bootstrapping phase? In the Web PKI ecosystem, for example, browser vendors, operating system developers, and alike mandate certificate authorities to undergo auditing programs and abide by specific regulations before being admitted in respective *trusted root certificate programs*. Trust bootstrapping phase is almost always neglected by proposed ICN certification solutions, for example by *NDNCERT* [271] in NDN, or name certification services in MobilityFirst [249]. These approaches fail to grasp the workings of current PKIs in their totality, *i.e.,* beyond technical aspects, and lack adequate solution to aforementioned challenges of trust bootstrapping.

Arguably the shortest path to introduce a PKI for name to owner authentication in ICN is to rely on existing infrastructure. Tehrani et al. [83], for example, propose to use DNSSEC for authenticated namespace management in NDN. Alternatively, certification for ICN can be delegated to the existing Web PKI. In any case, existing *trusted root certificates*, *e.g.,* as provided by operating systems, and security practices, *e.g.,* auditing certification authorities, can be reused for ICN without the need to conceptualized and run own infrastructure with extra organizational and regulatory overhead.

8.3.4 Accountability in ICN

Accountability, as an indicator of responsible practices, comprises taking responsibility and accepting consequences, *i.e.,* punishment and compensation of victims [181]. On the Internet, for example, one aspect of network accountability is to identify the origin of network disturbances, *e.g.,* denial-of-service attacks, and hold perpetrators responsible.

Accountability in data discovery and delivery. Granting the responsibility of name to location resolution to the network also shifts the accountability of data discovery and delivery to network operators. Compared with IP networking where the network is unaware of the actual content in transit and is only aware of sender's and receiver's respective IP addresses, any information-centric network has (or at leas can derive) information about the transmitted content based on its name. Given this type of knowledge about content itself, network responsibility and in turn accountability has to be reconsidered for ICN: take a set of entities that provide pirated content within an ICN network. As the network maintains routing or forwarding states in order to discover and deliver such content, the question arises if the network as such or respectively network operators can be held responsible for *contributory* or *vicarious* copyright infringement as it was the case for Napster in early days of peer-to-peer networking (see Penn [192] for details) even though Napster operators neither hosted nor directly distributed pirated material? The answer to this question becomes even more complicated if in-network caches as surrogate content providers are taken into consideration.

Historically, there has been a number of attempts by lawmakers to impose accountability regarding online content and to define liability boundaries for network or service operators. In the U.S., for example, the first attempt to regulate *indecent content*, was *Communication Decency Act* (CDA) (Title V of *Telecommunication Act of 1996* [276]) which was later struck down as unconstitutional by the U.S. supreme court leaving only the so-called *Good Samaritan protection* behind (later codified 47 U.S. Code §230) which relieves users and service providers of their liability regarding information provided by third parties. The very opposite idea paved the way for the *Netzwerkdurchsetzungsgesetz* [56] (NetzDG) in Germany which mandates social media operators to strictly moderate data, *e.g.,* respond to request for deletion of hate speech in a timely manner. Since ICN represents a fundamentally new approach, it is difficult to decide if existing Internet accountability and liability legislation as above constitute precedence for and is also applicable to ICN or new legal frameworks needs to be defined.

Accountability in data inquiry and provision. Through name to data and name to owner bindings, data origin, *i.e.,* data producer, is distinguishable. Data origin authentication is the basis of accountability for data provision. Producers of malicious, pirated or any other *unlawful content* can be identified using their respective digital signatures (non-repudiation) given that their public key is bound to their identity through a trust model (see previous subsection). The use of digital signatures, or more specifically the non-repudiation property, has also been proposed for IP networks to identify malicious hosts [165]. The proponents argue that identification of data producers is a crucial step in fighting against child-pornography, extremist content, and alike. Yet the importance of anonymous publishing and its societal or political role, *e.g.,* in whistle-blowing, is often neglected.

Data accountability is also extensible to data consumption. Just as data providers are hold liable for the content they produce, consumers are accountable in what they consume. The data-oriented abstraction of ICN, however, leaves little room for consumer identification. Respectively, the new class of malicious activity initiated by consumers, *e.g.,* interest flooding [90], [250], remains an open challenge in terms of accountability.

Accountability in namespace and name allocation. A logical consequence of names becoming valuable commodity and in turn subject to regulation and dispute (Unterabschnitt 8.3.1) is the growing concern of name misassignment and abuse. On the Internet, for example, the domain name registrars are mandated by the *2013 Registrar Accreditation Agreement* (between ICANN and registrars) to provide means for domain name abuse reporting and investigation.

8.3.5 Private communication

The very idea of ICN is at odds with private communication both in terms of private as in restricted to a specific group of communicating partners as well as private in terms of secret communication, also referred to as *privacy* in ICN literature [13], [319], [340], [136], [158], [243].

Private as restricted. The data-oriented communication abstraction of ICN which reduces producers and consumers to second class concepts, and the fact that ICN communication is receiver-controlled (reactive) yet receivers can generally neither be addressed nor identified is the main obstacle to private, *i.e.,* restricted, communication. In this sense a producer is unaware of consumers, thus, cannot initiate any consumer-specific data exchange (proactive) while an ideal consumer leaves the task of locating data to the network; it is only the data that matters. It is noteworthy that this deliberate design choice, among others to defy specific types of attacks such as unsolicited receiving of data through DoS attacks, has also paved way for new types of attacks (see for example interest flooding in NDN [90], [250]).

Restricted communication in ICN is, respectively, limited to additional capabilities provided by some approaches. NDN, for example, supports interest signing (consumer authentication), freshness flag for interests (avoiding caches), and forwarding hints (routing manipulation) which can be combined to restrict producers and consumers of a specific data exchange. Other methods, such as including the producers public key in an interest message (as proposed by Ghali et al. [92] to counter content poisoning) can also be used to shape restricted data exchange. Nonetheless, the question of how private, consumer-specific data exchange, *e.g.,* push notifications, can be initiated in ICN, remains an open question.

Private as confidential. Private, *i.e.,* confidential, communication, has been studied more extensively in terms of privacy. Privacy, evidently, is neither an absolute nor a quantifiable attribute. As such, it has been investigated in context of ICN in relation to *information leakage*, *i.e.,* confidential information that can be derived or observed from non-confidential properties. The Achilles' heel of confidentiality are names themselves, specifically expressive and human-meaningful ones which reveal information about the content being carried over the network. Katsaros et al. [118], for example, discuss how semantic names can reveal characteristics of respective content. In general, a number of attacks can cause information leakage through caches [340], [136], leakage through names and name resolution [118], or through network monitoring [13], [319], [158] (see Tourani et al. [243] for an extensive overview). Proposed remedies, *e.g.,* ephemeral or encrypted names, however, almost always run counter to fundamental architecture of ICN or its distinguishing features such as in-network caching.

Private as covert. The name to owner binding alongside the data-oriented security model of ICN appeals to mapping name to their respective owners and securing them through digital signatures. A signature in turn allows authenticating the producer and caters for non-repudiation. This presumably harmless architectural design, however, poses a non-trivial dilemma: anonymity. Anonymity is a dilemma as it poses both positive and negative consequences. In some cases there seems to be overwhelming (even universal) consensus regarding the necessity of anonymity, *e.g.,* for whistle-blowers, whereas in other cases there exists good arguments against it, *e.g.,* child pornography producers.

Various proposals aim to reach anonymity in ICN through forwarding methods similar to Tor's onion routing, *e.g., ANDaNA* [319] for NDN, or multi-tier confidential routing, *e.g.,* virtual anonymity model proposed for MobilityFirst [158]. Similar to proposed solutions for confidential communication (see above), however, applying these remedies also mean giving up other advantages that ICN promises.

8.4 Conclusion

In this paper, we discussed the ambitious reconception of networking through ICN and the resulting far-reaching consequences. We focused on arguments beyond mere technical aspects and considered fundamental economic, societal, and political aspects of a potential ICN-Internet. We have shown that the replacement of IP addresses by names also changes the non-technical landscapes of economic forces, power structures, trust relations, accountability, and private communication. Through this work, we hope to have been able to put the emphasis on the role of non-technical issues for acceptance and deployability of ICN on a global scale.

Kapitel 9

The Missing Piece: On Namespace Management in NDN and How DNSSEC Might Help

Abstract

Names are the cornerstone of every *Information-Centric Network* (ICN), nonetheless, namespace management has been by far neglected in ICN. A global and scalable namespace management approach is a challenge which not only concerns technical, but also requires attention to non-technical, *e.g.,* organizational issues. In this paper, we present both a clear position on namespace management in ICN and preliminary work on a potential solution. We conceptualize a namespace management system for hierarchical names and introduce a prototype for NDN, which leverages existing DNSSEC equipped DNS infrastructure. Based on this, we are able to implement both technical and non-technical aspects of namespace management. We consider lessons learned and pitfalls from decades of the ever-evolving development of domain name system. As the de facto standard namespace management for the Internet, it is an integral orientation factor for both our concept and its implementation.

9.1 Introduction

Names and their management have been considered crucial, right from the beginning of the Internet and its predecessor [408], [409], [466], [467]. Originally, names were administered by John Postel, later, to scale with the growth of the Internet, the *Internet Corporation for Assigned Names and Numbers* (ICANN) was designated for organizational aspects, such as name assignment, policy development and enforcement, and dispute mediation. This enabled the Domain Name System [167] (DNS) to become fundamental for almost all Internet applications. It not only defines a technical solution to resolve hierarchically structured domain names (in the technical protocol) but also provides an ecosystem that ensures proper name management (in the policy organization and community).

As the Internet's most relied upon technology that facilitates namespace management, which matured from research-labs into operational and commercial environments and has gone on to implement real-world multi-stakeholder deployment, the long-established DNS ecosystem is the canonical example that illustrates a clear need for some kind of namespace management in computer networks. In the Information-Centric Networking (ICN) community and literature, which introduces names as first class principles, this topic remains largely ignored. Current work on naming security in ICN mainly focuses on securing name to data bindings [57], [333], [105], [268] and provenance authentication [4], [57], [58], [105], [249], [258], [268]. Existing solutions are respectively limited to technical solutions while overlooking the derivative (but separate) need to attend the policy and organizational aspects.

In this paper, we argue for two positions: (*i*) A successful ICN deployment-model requires clarification of namespace policy and management, and (*ii*) augmenting the design of an ICN solution with the long-term facilities and experiences gained from the DNS ecosystem would allow us to leverage existing structures to address and overcome the significant number of technical, legal, and policy challenges that are inevitable as ICNs achieve operational and industry deployments.

In detail, we revisit the namespace management problem (**§ 9.2**) and present preliminary work to tackle the inevitable (often not only technical) challenges NDN will face in bolstering operational ICN deployment. To reach acceptance among multiple operational and commercial stakeholders, ICN needs to guarantee (*i*) exclusive ownership of names by publishers, (*ii*) the arbitration of rightful holders of named resources, and (*iii*) the verification of the content origin by consumers. To achieve this, we conceptualize (**§ 9.3**) and implement a scalable namespace management scheme while recognizing the fact that namespace management is only effective if both technical and organization aspects are attended [138]. We show the feasibility of our approach by presenting NDNSSEC (**§ 9.4**), a prototype that incorporates *DNS security extensions* (DNSSEC) [444] attestation objects into NDN to prove name ownership.

9.2 Namespace Management

A namespace $\mathcal{N}$ denotes a “set of names from which all names for a given collection of objects are taken” [317, §8]. A name is a unique string in an alphabet referring to some object; the set of all objects to which a name might be *bound* is called the *scope* of that namespace. A namespace, thus, can be formalized as a mapping, *i.e.,* a *functional binary relation*, with $\mathcal{N}$ and $\mathcal{T}$ being respectively its domain and codomain. Such mapping is functional but neither surjective, so that a thing can be left without a name assignment, nor injective to cater for name aliases. In the following, the notation $\mathcal{T}$, standing for set of *things*, is used to denote the scope of a namespace, and $\mathcal{N} \to \mathcal{T}$ denotes the binding of names to things.

Given a namespace $\mathcal{N}$, a (decentralized) namespace management scheme partitions $\mathcal{N}$ into management units, *zones* [330, §6], which are owned and maintained by an authoritative entity. Namespace management binds a name $n \in \mathcal{N}$ to a zone $\mathcal{Z}_i \in \mathcal{Z}$ where $\mathcal{Z}$ denotes the set of possible zones with each zone $\mathcal{Z}_i$ comprising a number of names. The zone owner manages or delegates the management of all or a subset of names within that zone. We use the notation $\mathcal{N} \to \mathcal{Z}$ to denote binding of names to zones, which similar to $\mathcal{N} \to \mathcal{T}$ is functional and left-total so that each name is mapped exactly to a single zone. It is neither necessarily surjective, so that there might be zones which include no names, *i.e.,* are not assigned, nor injective to allow mapping of multiple names to the same zone. Needless to say, manageability is proportional to the number of bound names; using zones reduces management overhead, while preserving the overall size of the namespace and improving the scalability.

Regardless of a formal definition and respective technical means of implementation and deployment, we argue that an effective global namespace management scheme, specifically for computer networks, must also address non-technical issues: not only name ownership must be guaranteed and protected but the management scheme itself must conform to constraints beyond the technical realm. An organization or a company might want to continue using terms, such as trademarks or any other distinguishable and established names associated with them in the real world, in order to remain recognizable in the virtual world. From a purely technical point, such associations are irrelevant and out of the scope of namespace mangement: a point of view, which we consider as short-sighted. In the following, we support our claim by revisiting the development history of DNS in the past decades.

The domain name system (DNS), as the most prominent example of namespace management in computer networking, is a hierarchical distributed key-value database, with $\mathcal{N}$ being the set of all possible domain names, $\mathcal{T}$ being Internet resources, and $\mathcal{Z}$ the set of *DNS zones*. When centralized name management became infeasible, DNS was conceived. The centralized administration was divided into sub-administrations [432], organized hierarchically in a tree structure, and meant to mirror the organizational structure of its managing authority [409]. Domain names were initially conceptualized as purely technical administrative entities allocated on a first-come / first-served basis. They were required only to be registered with the *central domain administrator*, have a designated maintainer, and provide their own name lookup service [432]. As the Internet began its commercializing phase and grew in terms of participating nodes, assigned domains, and users, it became evident that in reality a domain name goes beyond the purely technical context. In 1996, for example, a district court in Cologne, Germany, decided that a domain name corresponding to a city name is *not* subsumed under the naming laws of the German Civil Law. It was argued that a domain name is comparable with "telephone numbers, bank routing numbers or postal codes" and does not necessarily "establish an association between the domain name and its owner" [393]. In less than a year, another district court, this time in Frankfurt, Germany, decided the contrary. It argued that users not only expect to retrieve information *about* but also *from* the municipality of a city under the corresponding domain name while rejecting the previous comparison with phone numbers [392]. It was decided that the domain namespace is not limited to the former interpretation, *i.e., where to send the data* [317], but also regards the latter context, *i.e., indicating the origin* [138] in terms of the real-world owner of that name, and involves naming and trademark laws.

Analogous conflicts have arisen in the overlapping uses of domain names by independent resolution systems (global DNS versus internal DNS namespaces, for example). In cases where names appear similar but have conflicting meanings in different namespaces, name collisions have led to security vulnerabilities [40]. Such disputes and ambiguities (and others) were among the incentives (and types of incentives) for establishment of ICANN with the goal of not only addressing technical, but also non-technical manageability of DNS [419]. The reason is that different *contexts*, related to which a name is interpreted [447], are not always clearly separated or are even intentionally overlapping. It is,

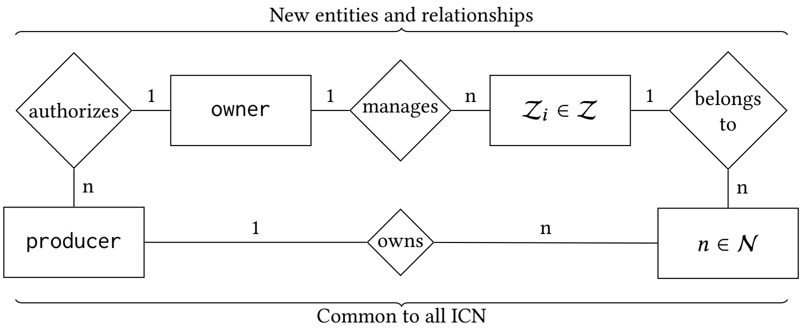


Abbildung 9.3.1: Entity-relationship diagram for a simple name space management framework.

thus, possible for a name or its equivalent variations to be *resolved* to different things in the *context* of different namespaces. The correspondence of different namespaces might cause unanticipated complications that need to be addressed non-technically. Such correspondence is not necessarily self-evident and needs to be acknowledged or established formally.

Finally, there are other less obvious yet non-trivial concerns regarding global namespace management that highlight the importance of its non-technical aspects. For example, depending on the scope of a namespace, having authority over a zone can asymmetrically put the owner in a position of power. Considering a hierarchically structured namespace where each zone apex denotes a subject, *e.g.,* '`/news/world/politics`', the owner of top-level apexes can influence the discourse of each subject, for example, by limiting the delegation only to favorable news outlets and suppressing the rest.

9.3 Conceptual Design

The $\mathcal{N} \rightarrow \mathcal{T}$ binding is the basis of any ICN with $\mathcal{N}$ being the ICN's namespace defined through a naming scheme and $\mathcal{T}$ the data objects within that ICN. Nevertheless, namespace management has not adequately been addressed in ICN. On the one hand, there exists no single authority which is in charge of global namespace allocation and assignment. On the other hand, there is no consensus on how to define ownership/membership of names or amalgamation into zones.

The first step in realizing namespace management in ICN is to acknowledge that the named data is served by different data holders and to codify the inherent subdivision of the namespace into management units, *i.e.,* zones. For the sake of simplicity we assume that zones are non-overlapping and a name only belongs to a single management unit so that the following holds $\forall i, j : \mathcal{Z}_i \cap \mathcal{Z}_j = \emptyset$. Names are then to be mapped to a zone, establishing a functional, $\mathcal{N} \rightarrow \mathcal{Z}$ mapping. Here we limit ourselves to structured naming schemes which allow embedding of zone identifiers in names, *e.g.,* as prefixes in hierarchical naming schemes or as authority field of URI-based schemes. For example, both '`ni://`**`ietf.org`**`/sha-256;Uya. . . _-Q`' (*cf.,* [333]) and '**`/org/ietf`** `/index.html`' contain zone apexes (in bold) of a hierarchically structured namespace directly in the name.

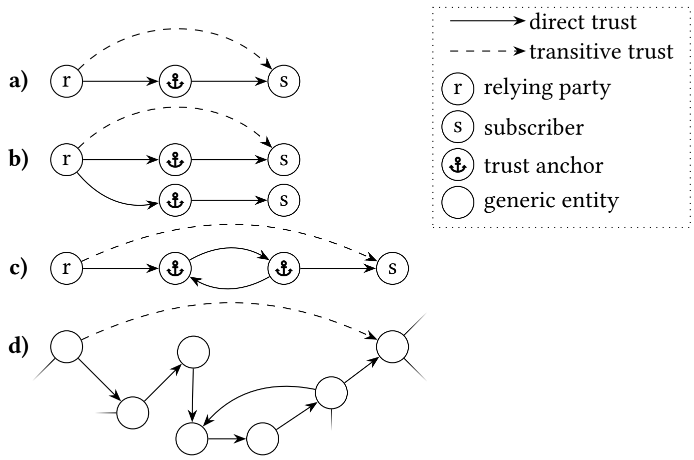


Abbildung 9.3.2: Trust models—a) Basic Trust Model, b) Multiple Trust Anchors, c) Cross Certified Trust Anchors, d) Web of Trust.

Furthermore, a logical authoritative entity is designated to assign and allocate zones to owners, *i.e.,* mapping zones to owners. A zone owner can then delegate management, authorize producers to publish under names belonging to that zone, and provide information about authorized producers. The relationship between zones, owners, producers, and names, as required for namespace management in ICN, is depicted in Abbildung 9.3.1.

Finally, given a binding, it should be possible to authenticate the binding, *i.e.,* to verify its validity, within the context of its respective namespace. In the following, we discuss two methods of securing bindings: the first method allows to authenticate the mapping by its intrinsic properties, whereas the second one relies on third parties to vouch for its correctness and validity.

Self-authentication/certification. A binding is said to be self-authenticating if its verification succeeds only by the bound values and locally available information. For example, if $\mathcal{N} \rightarrow \mathcal{T}$ is defined as $n = h(t)$ with t being a binary object and h a secure cryptographic hash function, the authenticity of the mapping between the name n and object t can easily be verified by calculating hash of t using h and comparing it with n, given that the applied hash algorithm is known. *Self-certification* requires the additional knowledge of a cryptographic key $k \in \mathcal{K}$ to authenticate the binding [93]. The main drawback of such bindings for names is that they diminish readability for humans and prevent aggregation, *e.g.,* in forwarding tables.

Trusted Third Parties (TTP). Beside self-authentication or certification, trusted third parties can be designated to directly vouch for and verify a given binding within a *trust management* framework. Involved parties are identified by their digital credentials, *i.e.,* cryptographic keys $\in \mathcal{K}$, and their actions are constrained by defined policies, while a *trust relation* denotes which authorities may issue credentials [26]. Trust relations are realized through certificates and trust can be established transitively when a *trust anchor* delegates credential issuance to a *subscriber* which subscribes to services of the TTP [473, p. 937]. Trust between a *relying party* r and a subscriber s is then transitively established through the TTP as depicted in Abbildung 9.3.2a. Alternative models may contain multiple trust anchors (Abbildung 9.3.2b) or cross certified trust anchors (Abbildung 9.3.2c). Prominent examples are the single rooted DNSSEC and multiple (cross certified) trust anchors model of Web PKI based on X.509 [297] certificates. A generalization of this approach is the Web of Trust (WOT) where every entity acts both as an authority and a relying party. Trust between two parties is then considered to be established if there exists a trust relation path between them [492] (Abbildung 9.3.2d).

In our concept, we define a simple policy for data publishing under a given zone: a producer is authorized to publish under that zone, if its public key is certified by the zone owner. The producer, in turn, must sign data before publication with the corresponding private key. Respectively, a consumer can verify if the publisher was authorized by the zone owner or not.

9.4 Implementation of NDNSSEC

In our implementation, NDNSSEC, we couple NDN as our ICN core with the existing DNS and DNSSEC infrastructure as the name managing unit. Our prototype extends present NDN tools to allow (*i*) NDN producers to publish and sign data and (*ii*) consumers to verify the data, based on existing DNSSEC data and with almost no manual interaction as long as name ownership is secured via DNSSEC.

DNS has been chosen as it not only adequately addresses technical but also non-technical challenges of a scalable global namespace management—and it is deployed, while NDN has the benefit of a hierarchical naming scheme that matches the hierarchical structure of DNS namespace. It is noteworthy that NDN explicitly excludes namespace management from its architectural design [420] and leaves it for the application layer. Our dependence on DNS is limited to its ecosystem and publicly available data and not necessarily its transport.

The software of our working prototype[1] is based upon *NDN C++ library with eXperimental eXtensions* (ndn-cxx) version 0.6.5 [313].

9.4.1 Securing Bindings

Namespace management requires four bindings (see Abbildung 9.3.1). We implement two of them based on existing built-in data-oriented security mechanisms of NDN, and the other two by integrating the attestation objects of DNSSEC.

Named data packets in NDN carry a name, payload, and metadata including signing information. A packet can be secured either using a hash digest for integrity examination or by a digital signature catering for both integrity and authenticity verification. For digital signatures, the signature block of an NDN data packet includes a `KeyLocator` field which denotes under which name the certificate of the signing party can be retrieved. Certificates in NDN are ordinary data packets which carry signed public keys as payloads.

[1]Source code available under `https://gitlab.com/ndnssec`

In the following, the terms *name* and *producer* respectively denote the packet identifier and the entity responsible for its creation and provision [484]. We use the term *zone apex* to denote a DNS zone origin, *i.e.,* the name at the root of a zone tree [364, §7]. A zone apex is either a conventional fully qualified domain name (FQDN) or its equivalent *reverse slash separated* notation, *e.g.,* '`tools.ietf.org.`' or '`/org/ietf/tools`' (*cf., ndnification* [279, §3]). *Zone owners*, are real-world entities which are authorized to manage the namespace within a zone.

In our selected trust management framework based on DNSSEC, ICN nodes, *e.g.,* producers and consumers, which can be subject to authentication are represented through their cryptographic keys or more specifically through NDN certificates. There exists a (logically) singular trust anchor, the root zone owner, and the remaining zone owners act as its subscribers. Similar to ICN nodes, zone owners, are also represented through cryptographic keys listed as `DNSKEY`s in a zone's authoritative data. Securing bindings succeeds as follows:

Name to Zone. We establish name to zone bindings by prefixing names with zone apexes. As zone apexes are embedded into names, $\mathcal{N} \rightarrow \mathcal{Z}$ mappings are transitively authenticated by authenticating name to producer and producer to zone bindings.

Name to Producer. This binding is already given in NDN. A data packet binds its names to a producer through a digital signature [265].

Producer to Zone. This binding is realized using `DNSKEY` and `RRSIG` records of DNSSEC. The zone owner enlists the public key of producers as `DNSKEY`s to indicate producers which are authorized to publish under its managed zone.

Zone to Zone Owner. DNS zone delegation is leveraged to bind zones to zone owners.

9.4.2 Workflow

Prior to publishing under a zone apex, a producer is required to have its public key served by the DNS zone owner as a `DNSKEY` record. Upon publication, the `KeyLocator` field of the NDN packet is set to `/<SubjectName>/KEY/[KeyId]/[IssuerId]/[Ver]` (see [486] for certificate naming convention in NDN), with `SubjectName` denoting the zone apex, and `KeyId` the digest of the producer's public key calculated the same way digests are produced for DNS `DS` records [446, §5].

Upon receiving a data packet, our extended NDN consumer software first verifies that the content name is prefixed with a zone apex matching the `SubjectName` segment of its `KeyLocator`. Otherwise, the packet is discarded. Then, the `DNSKEY`s under the zone apex are fetched via DNS, their digests are computed, and finally compared with the `KeyId` to find the matching public key. If a match is found, it can be used to verify the signature of the packet. It is also possible for each domain to provide a logical public key resolver (similar to a DNS stub resolver) which is trusted by consumers of that domain: given a zone apex and a public key digest, the resolver verifies whether the key is served by the zone owner as authorized and returns a corresponding public key for that producer as depicted in Abbildung 9.4.1, otherwise it returns an explicit `NACK`. Moreover, a domain-local resolver enables

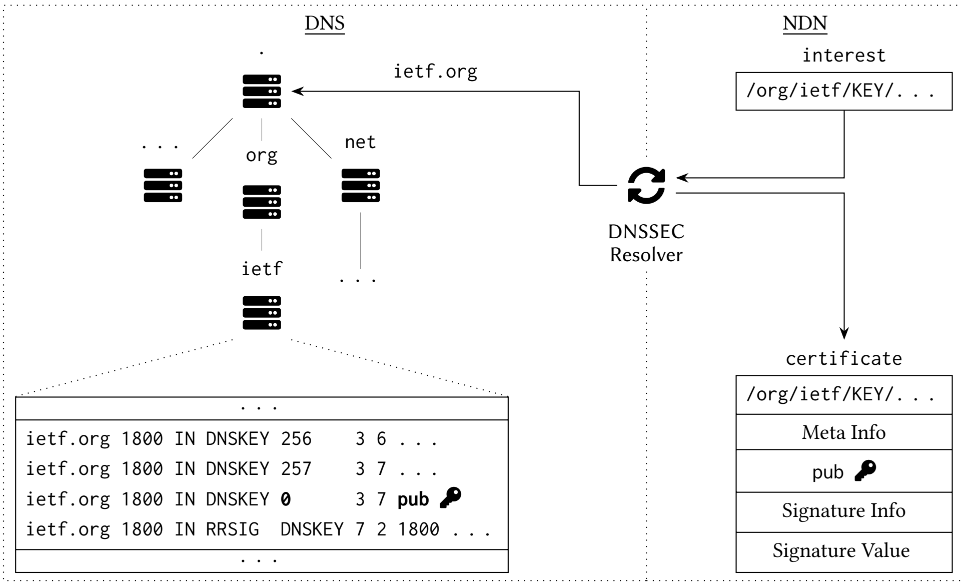


Abbildung 9.4.1: Resolver as bridge between DNS and NDN.

private zone management services for zone apexes which are explicitly used for internal purposes, similar to *internal top-level domains* (iTLD).

Although our prototype relies on the DNS transport to fetch zone information, there are alternative methods to integrate publicly available DNS data into NDN without compromising the integrity of our approach. A zone owner can, for example, publish an *authentication chain* (*cf.,* [454]) as a simple NDN packet which comprises signed DNS record sets all the way from a trust anchor to the zone it manages so that consumers can fetch and authenticate zone records without having to consult the respective name servers. An alternative is to mirror DNS data using NDN-specific distributed key-value databases resembling the DNS approach, *e.g.,* NDNS [279], [4]. Either approach, however, introduces a synchronization problem which can lead to temporary discrepancies between DNS data and their NDN representation. A solution for this is beyond the scope of this work.

9.4.3 Evaluation

Performance evaluation of our proposed approach mainly reduces to evaluating DNS, which has been extensively researched elsewhere and is out of the scope of this work. A comparative evaluation is also not meaningful due to lack of existing alternatives. Instead, we focus on a qualitative analysis of data authentication and the computational complexity of our trust management model by comparing it to NDN's schematic trust [262] and NDNS [279], [4] as alternative trust management frameworks.

The trust policy of DNSSEC can be realized using trust schemata by defining a single trust rule as follows [262]:

Rule	**Data Name**	**Key Name**
`cert`	`(<>*)(<>)<KEY>[id][iid][v]`	`cert(\1,null) | root()`

Tabelle 9.4.1: Complexity of certificate chain verification.

	Level 1 (root)	Level i (interim)	Level n (leaf)	|⚓|
Trust Schema	⚓ $\mathsf{C}_0 \leftarrow \cdots$ —	$\mathsf{C}_i \leftarrow \cdots$ —	C_n	≥ 1
NDNS	⚓ KSK_0, DSK_0	DKEY_i, KSK_i, DSK_i	DKEY_n, KSK_n, DSK_n, $\mathsf{APPCERT}_n$	≥ 1
NDNSSEC	⚓ DNSKEY_0, NS_0, DS_0	DNSKEY_i, NS_i, DS_i	DNSKEY_n	$= 1$

This rule states that for any given certificate, the certificate of its signing entity can be found under the same name but on a level higher in zone hierarchy until a trust anchor (`root()`) is reached. A consumer starts by fetching certificates for each zone, starting at level n, up to a known trust anchor and verifies the certificate chain back to the zone certificate. NDNS proposes an approach resembling that of DNSSEC: the chain verification starts at a known trust anchor and ends with a certificate. Both approaches allow multiple trust anchors to coexist with a high price of i) complicating the trust bootstrapping phase, ii) requiring consumers to redo chain verification, if selected trust anchor does not lead to the certificate under investigation, and iii) increasing management overhead needed to avoid policy collisions. Tabelle 9.4.1 depicts how each approach traverses from a trust anchor to a certificate used to sign a data packet.

Furthermore, NDNSSEC proposes a different method to address certification validity verification and revocation which directly determines the certificate retrieval frequency by a consumer, *e.g.,* when authenticating data streams. Whereas in NDN certificates carry a *freshness period*, comparable to DNS TTL, alongside a *validity period* (also common in X.509 certificates), in our approach, we only use TTL and leave it to consumers to decide on an appropriate policy regarding the validity period of retrieved public keys, *e.g.,* for caching. In NDNSSEC, a public key listed as a `DNSKEY` is considered to be valid and no further revocation mechanisms are required. It should, however, be noted that DNS updates are constrained by propagation delays [88] so there might be short periods of time when retrieved record sets are out of sync.

Comparing to aforementioned approaches, advantages of NDNSSEC can be summarized as follows:

1. **Non-technical policy enforcement** by relying on existing organizations such as ICANN.

2. **Minimal management burden on consumers** by designating a single party as trust anchor (DNS root).

Tabelle 9.5.1: Namespace and trust management in ICN with $\mathcal{N}$ denotes names, $\mathcal{T}$ things, $\mathcal{Z}$ zones, $\mathcal{I}$ identities.

	Authentication					
	$\mathcal{N} \rightarrow \mathcal{T}$	$\mathcal{N} \rightarrow \mathcal{Z}$	Self-auth	Zone Scope	Trust Relation	Naming Example
DNSSEC [444]	✓	✓	✘	$\mathcal{N}$	Decentralized	`tools.ietf.org`
Detti et al. [57]	✓	✓	✓*	$\mathcal{I}$	Decentralized	`<a0914a9. . . 7287a, 0902908. . . 6ec006a>`
Venkataramani et al. [249]	✓	✘	✓*	$\emptyset$	N/A	`8c6a365205c874144. . . 3765a2190404de6a9`
Farrell et al. [333]	✓	✘	✓	$\emptyset$	N/A	`ni:///sha-256;UyaQV-Ev. . . 1aGQAlMO2X_-Q`
Wong and Nikander [258]	✓	✓	✘	$\mathcal{I}$	Centralized	`scheme://authority/page.html`
Afanasyev [279], [4]	✓	✓	✘	$\mathcal{N}$	Decentralized	/level$_1$/. . . /level$_n$/data/name
Mahadevan et al. [155]	✓	✓	✘	$\mathcal{N}$	Decentralized	/level$_1$/. . . /level$_n$/data/name
DiBenedetto and Papadopoulos [58]	✓	✓	✘	$\mathcal{N}$	Decentralized	/level$_1$/. . . /level$_n$/data/name
Zhang et al. [268]	✓	✓	✓*	$\mathcal{I}/\mathcal{N}$	Decentralized	`/arbitrary/ndn/conform/name`
Hamdane et al. [105]	✓	✓	✓*	$\mathcal{I}$	Decentralized	`/producerID/contentID/validity/ver/seg#`

* Denotes *self-certification* (see Section 9.2)

3. **Deterministic authentication** by always guaranteeing a path from the root to a zone owner, otherwise signaling that a zone has not been assigned or delegated through *explicit denial of existence* [479] of DNSSEC .

4. **No additional infrastructure for certificate revocation** by simply replacing compromised or outdated keys through new `DNSKEY` records (cf. *suicide lists* in NDN [486]).

9.5 Related Work

The authors are not aware of any previous work that has elaborated the non-technical policy aspects of namespace management in ICN. In this section, the review of related work is, thus, limited to technical issues, such as authenticating bindings and establishing trust.

Self-certifying names. Self-authentication/certification has been proposed in a number of approaches to secure $\mathcal{N} \rightarrow \mathcal{T}$ relations. In CONET [125], names are structured as $P : L$ where P denotes the public key's digest of a producer and L the cryptographic hash of the label [57]. Similarly, in MobilityFirst [249], principal names are self-certified, *i.e.,* are a digest of the owner's public key, whereas names of data objects are cryptographic hashes of the data itself, *i.e.,* are self-authenticating. Using hashes as names is also proposed in *NetInf* [50] within a URI-based naming scheme where the path component represents the base64 URL encoded digest of the content [333].

Trusted Third Parties. Wong and Nikander [258] leverage a URI-based naming scheme with producer's identity placed as authority segment. A centralized resolution server resolves identities to a public key while mapping the resource path of the URI to the object's metadata, including its digital signature. A distributed solution of the problem provides *NDNS*, an always-on distributed lookup service for NDN [279], [4]. Inspired by DNS, NDNS maps names to a number of records used as routing hints (NS), certificates (APPCERT, CERT), and general records (TXT) [4]. Here, the managed namespace is a subset of NDN namespace, which can bidirectionally be translated into domain names. Similar to NDNS, *key resolution service* (KRS) [155] introduces a distributed resolution mechanism, but for CCN. It can resolve a name into at least a content hash, publisher certificate, or a certificate chain. Both in NDNS and KRS, authenticating $\mathcal{N} \rightarrow \mathcal{T}$ and $\mathcal{N} \rightarrow \mathcal{Z}$ relations succeeds indirectly through publisher authentication. DiBenedetto and Papadopoulos [58] propose to use a resolution service, such as KRS or NDNS, to map a name to the public key of the domain owner. The corresponding private key acts as a key-signing key (KSK) so that any producer with its key certified by the zone KSK is considered as authorized to publish under the corresponding zone.

Identity-based Cryptography (IBC). To prevent storing public keys at third parties, IBC [452] has been proposed. In IBC an identity, *e.g.,* content name, is used to generate the corresponding public key using parameters provided by a trusted *private key generator* (PKG). Zhang et al. [268] propose an IBC based method where a CCN/NDN compatible hierarchical name $n \in \mathcal{N}$ is used as identity. A producer can also prefix a name with its identity $i \in \mathcal{I}$ to denote ownership. Data packets here carry metadata including the producer's identity, establishing a $\mathcal{N} \rightarrow \mathcal{I}$ relation, and the respective PKG public parameters, enabling self-certification of $\mathcal{N} \rightarrow \mathcal{T}$ relations. If names are prefixed with producer IDs, the producer fetches the corresponding key-pair directly from PKG, otherwise a name resolution server (NRS), responsible for key management and policy enforcement, acts as a proxy to the PKG. In either case, consumers only need to fetch the public parameters of the PKG to authenticate data packets. The authors also propose a hybrid solution with domain level PKG/NRS managing *domains* alongside a global PKI for their hierarchical management.

A variation of IBC, namely hierarchical IBC (HIBC), where key generation and identity authentication is delegated among hierarchical PKGs, is leveraged by Hamdane et al. to enhance NDN with a custom naming structure to enable producer identification and authentication alongside integrity verification [105]. In the proposed scheme each segment of an NDO's name n is mapped to a designated PKG which implicitly defines a $\mathcal{N} \rightarrow \mathcal{I}$ relation where $i \in \mathcal{I}$ denotes an organizational entity. Here, having the parameters of the root PKG allows the authentication of credentials issued throughout the hierarchy but suffers from the *key-escrow* problem, *i.e.,* a PKG has or can generate private key of its children.

Comparison. We summarize our observations in Tabelle 9.5.1. All approaches cater for authenticating $\mathcal{N} \rightarrow \mathcal{T}$ and mostly allow for defining zones as collection of names by the same producer or as an independent entity. Self-authentication, either for $\mathcal{N} \rightarrow \mathcal{T}$, $\mathcal{N} \rightarrow \mathcal{Z}$, or both, has been leveraged alongside centralized or decentralized trusted third parties where self-authentication does not suffice. To authenticate TTPs, *e.g.,* resolution servers and PKGs, all approaches use some variation of chain of trust either with domain-local or global trust anchors. Regardless of which method is used to establish trust, a *trust bootstrapping* phase is necessary to establish trust anchors or trusted credentials in general. In contrast to this paper, all methods assume this phase as given without giving further details.

9.6 Conclusion and Research Roadmap

The *technical* mechanics of a namespace are critical to ICNs, but there are many aspects to operational networks and ecosystems that extend beyond just the technical landscape. With over 30 years of experience in global namespace management of over 354 million domains and billions of euros in business per year, the DNS industry has illustrated that policy and management aspects of namespace resolution are critical components. The authors believe that this is the first work to address the *necessary* requirements of namespace policy and management for ICNs, and this is presented through the lens of using the resources that exist in the operational Internet, today. This work underscores the observation that names are not just labels used to identify things, they require policy and context. It is, thus, important to manage names and how they are used. It is also crucial for users to be able to examine if a name is used by its authorized owner, and have policy frameworks to codify "authorization."

In addition to formalizing namespace management for ICNs, we have proposed a concrete solution for hierarchically structuring and assigning the namespace of NDN, based on DNSSEC. We have shown how minimal changes to existing mechanisms of NDN can enable collision-free and scalable namespace management, which addresses both technical and non-technical shortcomings. Finally, as the success of the Internet is driven by community efforts, we propose the following research road map. In the short term, integrating DNS data into the ICN ecosystem without relying on DNS-specific transport is an important step. Such activities should consider the performance impact on the ICN ecosystem in terms of synchronization disparities, which might ensue and affect various aspects of publishing, discovery, and retrieval of data objects. In the long run, there is clearly value in the evaluation of our proposed approach with respect to large-scale user studies, *e.g.*, by modeling real-world settings based on traces from global cloud providers. Upcoming work should also probe the feasibility of NDNSSEC in confined use cases such as disaster scenarios with intermittent connectivity or fragmented networks. Making the right trustworthy public keys locally available without access to the public key infrastructure is challenging but might benefit from ICN in-network caching.

Acknowledgments. We would like to thank the anonymous reviewers and our shepherd Karen Sollins for their concise and constructive feedback. We appreciate the help of Luca Keidel, who provided the first version of our prototype. This work was supported in parts by the German Federal Ministry of Education and Research (BMBF) within the projects *I3* and *Deutsches Internet-Institut* (grant no. *16DII111*).

Kapitel 10

SoK: Public Key and Namespace Management in NDN

Abstract

Named data networking (NDN) enables scenarios where decentralized content distribution based on names is the centerpiece of networking. In this paper, we systematize two requirements to enable trust on a global scale in NDN, namespace management and public key management. We provide a framework to systematically assess and evaluate namespace and public key management systems, and relate their features to DNSSEC and Web PKI, the most prominent and accessible implementations of both building blocks on the current Internet. Our systematization of knowledge of existing approaches in NDN highlights strengths and shortcomings to derive options for future research.

10.1 Introduction

Information-centric networking (ICN) is a paradigm shift in networking that centers around the idea of naming data and having the network layer handle name-based data discovery and delivery. The underlying premise of this idea is that people care about data and not its topological location in the network [115]; an idea which dates back to overlay networks such as P2P networks or the "Internet In-

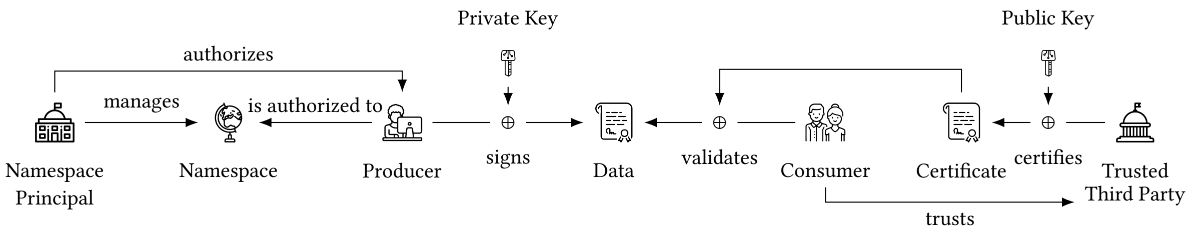


Abbildung 10.1.1: An overview of entities, roles, and relations in namespace and public key management in NDN

direction Infrastructure” (*i*3) [229]. Named-data networking (NDN) [265] is one of the most prominent and currently most lively concrete proposals of ICN.

In NDN, the name-to-data binding, as the foundation of ICN, is implemented by digital signatures over data packets that contain both the name and data in the same packet. However, as NDN names are not intrinsically bound to data, because any arbitrary producer can claim a name and bind data to it, additional effort is required to maintain *referential transparency* such that the relationship between name and data remains unambiguous. To achieve this, two further bindings are required: (*i*) name-to-producer (as name owner) and (*ii*) public-key-to-producer (as content generator). These bindings rely on two buildings blocks. A namespace management system is assumed that allocates namespaces (*e.g.,* `/org/ietf/*`) to producers and enables consumers to verify the ownership of a name, and a public key management system to bind producers (and possibly other roles such as namespace principals) to public keys, to define key life cycle management and trust models. Authenticating a name-to-data binding then succeeds transitively by first authenticating whether the producer is authorized to use that name, and whether the private key used to sign the data is valid and belongs to the producer. How these building blocks are implemented in detail and which further assumptions are made may differ between NDN applications. Considering an NDN ecosystem that allows for the involvement of multiple stakeholders requires interoperable design choices, though.

In this paper, we investigate trust in named-data networking by systematizing knowledge about namespace and public key management in NDN. Our analysis is based on surveying more than 30 NDN applications introduced between 2015 and 2021 at common ICN publication venues. We contextualize design choices and requirements to derive a framework that may serve to compare current and future NDN applications and derive a common understanding of namespace and public key management in NDN.

The remainder of this paper is structured as follows. We start with a brief overview of NDN in Abschnitt 10.2. In Abschnitt 10.3, we discuss the relation of namespace and public key management to name-to-producer and public-key-to-producer bindings in NDN, and propose two taxonomies to better understand these two aspects and their roles in bringing trust to NDN. In Abschnitt 10.4, we study, assess, compare, and discuss pitfalls of existing approaches of namespace and public key management for NDN. We conclude our paper in Abschnitt 10.5.

10.2 Background on Naming and Security Principles in NDN

The main purpose of a named data network is to find data objects based on their names in a given context. Various naming schemes, defining flat or hierarchical namespaces [46], have been proposed by different ICN implementations, each with its own set of unique features [3], [19], [93]. Names, at the core of ICN, are bound to data objects (*i.e.,* metadata and payload) to form *named data objects* (NDO) as depicted in Abbildung 10.2.1. Name-to-data binding is realized by producer signature over an NDO which also caters for data authenticity and integrity. The basic premise here is that a producer who publishes an NDO under a given name is authorized to use that name. A producer is identified by

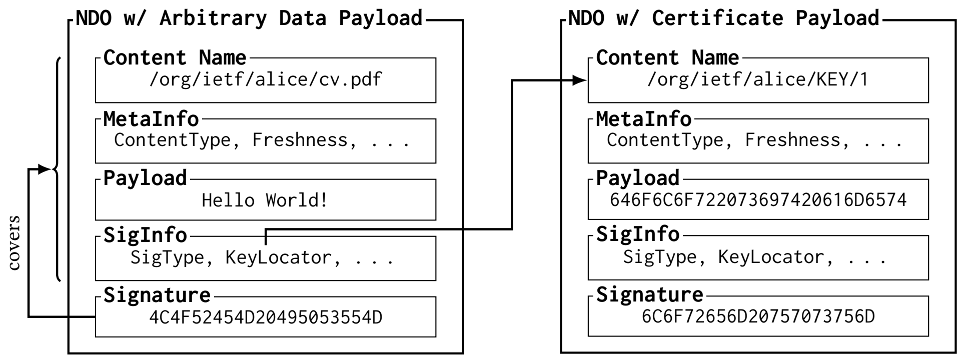


Abbildung 10.2.1: Examples of NDOs with arbitrary payload (left) and certificate payload (right) in NDN.

its signature which can be authenticated by the producer certificate. In this section, we give a brief introduction on naming and namespace in NDN as well as NDN security measures based on digital signatures (public-key cryptography).

Naming and namespace structure. NDN uses a hierarchical naming structure similar to domain names (see Abschnitt 10.A for an overview of DNS and its security extensions). An NDN name is composed of *components* [418] (comparable to labels of a domain name) that are separated by a forward slash. For example, the domain name `ietf.org` can be *ndnified* [279] as `/org/ietf`. NDN does not put any constraints on names or their global uniqueness [490], and explicitly leaves namespace management out of its specification [265]. Applications can freely define their own namespaces by deciding on structure, *e.g.,* number and order of components, and semantics, *e.g.,* using functional or constant name component of names (see Tabelle 10.2.1). Namespaces with a limited scope, *e.g.,* local scope of smart home applications, can be managed by a single entity, whereas global namespaces, *e.g.,* used in routing and forwarding algorithms, need coordination among various stakeholders to avoid naming conflicts. Retrieving data on a global scale requires globally unique names [490].

In contrast to flat namespaces [333], [93], [125], the binding between data and name of the data is neither self-authenticating nor self-certifying in NDN. NDN needs extra measures to authenticate data objects.

Object security to verify data integrity. Digital signatures in NDN build the basis for object security. Object security allows for securing data regardless of its location or transmission channel. The minimal security assurance guarantees data integrity through cryptographic hashes [159]. The security model of NDN relies on this approach by requiring all NDOs to include a signed hash [265], [272] covering an NDO name, metadata, signature information, and its payload (see Abbildung 10.2.1). By computing the digest of an NDO and comparing its signed hash, a consumer can verify that the NDO has not been tampered with, *i.e.,* neither the payload nor its name or metadata has been modified. Note that only a signature by an authorized producer is acceptable as a proof of data integrity.

A producer is identified by its certificate. Certificates in NDN are special data packets that carry a public key as payload and are identified by the following naming convention: `/<IdentityName>/`

Tabelle 10.2.1: An overview of selected CCN/NDN applications and their namespace and key management requirements, based on surveying research published at ACM ICN ’15–’21

		Namespace Requirements			Key Usage		
	Name	Prefix	Functional Components	Name Format†‡	Confidentiality	Authentication	Access Control
Routing and forwarding	NLSR [112]	Network name	Site and router names	/<network>/<site>/<router>	–	Routing messages	–
	LSCR [109]	Network name	Site, router, msg type	/<network>/<site>/<router>/**LSCR**/ **LSA**/<typeID>	–	–	–
	SNAMP [5]	Global prefix	–	/<network>/<site>/	–	Link objects	–
	MNDN [176]	1. Global prefix 2. Name server	– DFZ prefix	/<network> /**GNRS**/<DFZ-prefix>	–	Link objects	Zone mappings
	KITE [270]	Global prefix	Tracing segment	/<network>/<traceSeg>	–	Trace interest	–
	LEO NDN[148]	Satellite location	–	/<baseNS>/<satID>	–	–	–
Sync	ChronoSync [273]	1. Broadcast space 2. –	Sync interest Sync reply	/<broadcast>/<appName> /<producerID>/<appName>	Sync data	–	Sync group
	PSync [267]	Multicast space[1]	Sync interest and reply		–	–	–
	MMORPG Sync [169]	Game ID	Game instance	/<appID>/<gameInst>	–	–	–
Security	ICN-based MIS[14]	Identity	Application ID	/<idPart1>/<idPart2>/<appID>	–	Identities	Data
	CCN-AC[134]	Anonymizer domain	Parameters	/<anonDomain>/[<encName>|<cmd>]	Interest/Data	Anonymizer/Caches	Interest/Data
	NDN OCSP [202]	1. Query service 2. Update service	– key ID and Update commans	/<ocspNS> /<server>/<keyID>/<cmd>	–	Services	Update service
	NDN-ABS [199]	–	ABE public params	/<baseNS>/**ABE**/<public-params>	Data Packets	Producer	Consumer
Diagnostic	NCMP [161]	–	Command and params	/<baseNS>/**register**/<cmd>	Result	Requester	Server
	NDN-Trace [122]	Trace prefix	Parameters	/**Trace**/<pathType>/<traceType>/< name>/<nonce>/<FaceID>	–	–	–
	DNMP [179]	Network root	Parameters	/<root>/dnmp/<params>	(Replies)	Commands	Commands and replies
RPC	NFaaS[132]	Execution prefix	Parameters	/**exec**/<appClass>/<func>/<input>	–	Requester	Functions
	RICE [130]	1. Function name 2. Instance name	parameters state	/<func>/<params> /<instID>/<func>/<state>	(Input)	Results	(Server)
	CFN [131]	Node name	Parameters	/<nodename>/<framework>/<params>	–	–	–
	IceFlow [135]	Application name	Dataflow parameters	/<app>/<actor>/<instance>/data/< partition>/<object>	–	–	–
IoT	Multi-Source IoT [9]	Multi-source ID	producer/device ID	/<baseNS>/<msINT>/<deviceID>	–	–	–
	Keyword-based ICN-IoT [15]	–	Function name and hashtags	/<prefix>/<func>/<hashtags>	(Data)	(Users)	(Devices)
	IoT On-boarding [47]	Network root	Commands and parameters	/<root>/<cmd>/<params>	Data Packets	Devices	–
	IoT QoS [100]	–	Traffic class	/<baseNS>/<trafficClass>	–	–	–
	FW Update IoT [99]	Deployment ID	Globally unique vendor / device class	/<deplID>/<vendorID>/<devCls>	–	Vendor identity	–
Misc	NDN-RTC[101]	–	Steam and packet params	/<baseND>/**streams**/<streamID>/< threads>/**frames**/<packetType>/< frame>/<dataType>	–	Producer	–
	CNS [39]	Organization	1. Administrative ID 2. Incident reponse ID	/<org>/**authorities**/<adminID> /<org>/**incidents**/<incidentID>	–	–	–
	TCP/ICN [168]	1. Forw. Proxy prefix 2. Rev. proxy prefix	TCP payload TCP headers	/<prefix>/<tpc4tuple>/<seqNo>/<wrapNo> /<prefix>/<tcpHeaders>/<nonce>	–	–	–
	NDN DeLorean [263]	–	Chronicle Tree	/<service>/<type>/<state>/<index> <digest>	–	Service timestamps	–
	NDNFit [489], [264]	Trust root	User, device, and app IDs	/<root>/<user>/<dev>/<app>	Content	Producer	Userspace
	HPNM [260]	–	Video encoding parameters	/<baseNS>/**SVC**/<name>/<params>	–	–	–

† For the sake of simplicity name formats only represent most significant prefixes of names and omit the rest (*e.g.*, segment component)

‡ Notation: <> denotes a functional component that can be composed of multiple specific name components, [] denotes alternative components (separated by |), and constant components are in **bold** type.

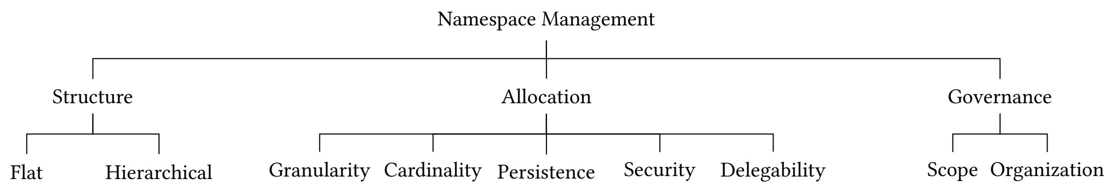


Abbildung 10.3.1: A taxonomy of namespace management

KEY/<KeyId>/<IssuerId>/<Ver> [416]. The IdentityName is comparable to subject name of X.509 certificates [297], [372] and serves to identify the certificate holder. Every signed data packet includes a KeyLocator field [417] that contains the name of the certificate which can be used to authenticate data signature. For certificates, the KeyLocator field contains the certificate name of the issuer. Note that a certificate is published under the namespace of its owner denoted by the /<IdentityName> prefix. This implies that the certificate issuer is authorized to generate and sign NDOs under the special prefix /<IdentityName>/KEY that it does not own.

Establishing trust in NDN resembles SDSI [443] by putting the burden of defining trust anchors and trust relations on each involved entity [272]. It is noteworthy that this approach fundamentally differs from the most prominent deployment of public key infrastructure (PKI) on the current Internet, the Web PKI (see Abschnitt 10.B for an overview). Beside authentication, public key cryptography is used by NDN applications for encryption and access control purposes (see Tabelle 10.2.1).

10.3 Problem Statement: Trust in Named Data Networks

Authenticating name-to-data bindings in NDN presupposes authenticating name-to-producer and public-key-to-producer bindings. These two bindings build the basis of trust in NDN by allowing a data consumer to verify that a name is not spoofed, and an authorized producer has generated the content bound to that name. Achieving a common understanding of trust in NDN requires consensus among producers, consumers, and other involved nodes on the network regarding the context that defines these bindings.

10.3.1 Name-to-producer Binding and Namespace Management

Binding a name to a content producer requires authorization of a producer to publish data under that name. In contrast to self-certifying or self-authenticating names, names are arbitrary in NDN and an external mechanism is needed to verify the ownership of names. This includes namespace management to allocate names to their legitimate owners (see Bechtold [21]) and functions that enable consumers to securely validate name assignments. In Abbildung 10.3.1, we propose a concise taxonomy of namespace management adapted from prior work [21], [46], [121], [447], [458] based on *structure*, *allocation*, and *governance*. After defining each aspect, we also briefly discuss the pitfalls in each category that need to be considered when designing namespace management for NDN.

Structure. A namespace structure can be *flat* or *hierarchical*. DNS, for example, uses a hierarchical naming structure which is divided into sub-namespaces or zones. Each zone can define its own set of structuring rules. For example, the `.us` top-level domain (TLD) defines a rigorous structure up to fourth level in its hierarchy [314]. NDN uses hierarchical names not only to improve interest forwarding and data delivery (*e.g.,* through prefix aggregation) but also to cater for semantically meaningful names, *e.g.,* as the basis for its security approach (see Abschnitt 10.2). In addition to its syntactical structure, a namespace can be structured along semantic meanings of name components. For example, ChronoSync [273], a synchronization protocol for NDN, defines a broadcast namespace (*e.g.,* `/ndn/broadcast/chronos/chat`) to synchronize the state of messages published by producers under their own namespaces (*e.g.,* `/wonderland/alice/chronos/chat`).

Many NDN applications drive the namespace structure to ease the development of applications (*e.g.,* see Thompson, Gusev, and Burke [240]). Using functional name components [423] to convey specific semantics leads to larger implication in global settings, though, in particular when functional components are shared among multiple producers. In our previous example, ChronoSync, having a broadcast namespace requires different producers to be authorized both to a broadcast namespace and their own namespaces at the same time. If names-to-producer binding succeeds through certification of a producer public key (as we discuss in the next section), either all producers must share the same key to announce their local state changes under the broadcast namespace, or have their public key certified both by the broadcast namespace owner and the namespace principal of their own namespaces.

Allocation. Allocation describes how names are allocated to namespace principals in terms of *granularity* (*i.e.,* allocation unit), *cardinality* of namespace to principal allocations, validity period (*persistence*), *security* of allocation method, and possibility of delegating a namespace to a third party (*delegability*). In DNS, for example, a complete sub-namespace, a zone, is delegated to exactly one zone owner, who can further delegate any part of its own namespace. DNS allocations are not persistent, and zone ownership can vary during time. The security extensions of DNS (see Abschnitt 10.A) are used to cryptographically secure delegations. On the top level of DNS hierarchy, *e.g.,* `.gov`, allocations might be subject to specific eligibility requirements and rules defined by the namespace governance. Similar to zone delegations in DNS, In NDN a complete namespace denoted by a prefix is allocated instead of single names.

Governance. We consider the set of rules, policies, and politics around names and naming as namespace governance. To illustrate this, we refer to the history of DNS and its evolution from local `HOSTS.TXT` files to a globally distributed database governed by the *Internet Corporation for Assigned Names and Numbers* (ICANN). During this process the management of domain names, which mainly used to be of technical character, was enhanced with policies agreed upon among various stakeholders, and national and international laws to regulate naming on the Internet, *e.g.,* to address naming conflicts due to trademark violations. Names in DNS, or more specifically namespaces in terms of zones, are al-

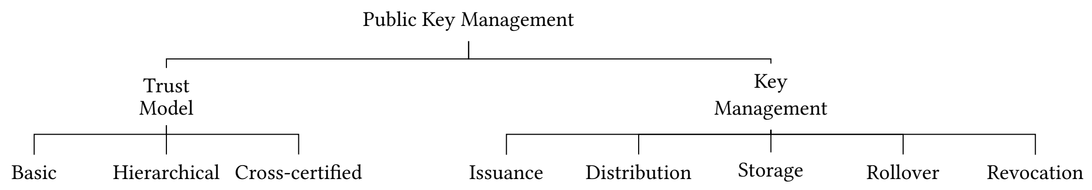


Abbildung 10.3.2: A Taxonomy for Public Key Management

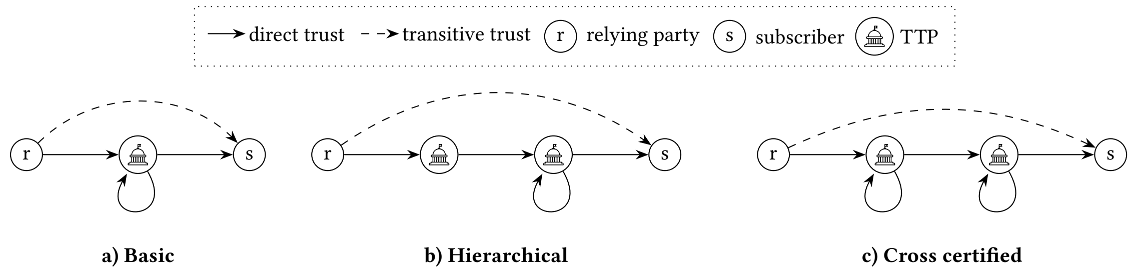


Abbildung 10.3.3: Public key management trust models (loops denote trust anchors)

located from ICANN to registries to a single logical principal, the zone owner. Khare [121] argues that such one-to-one mappings create a degree of scarcity with political consequences, observed during the DNS reform and emergence of ICANN. The importance of ICANN and more specifically its role with respect to transparency and accountability in establishing trust on the Internet is also acknowledged in a report by the Global Commission on Internet Governance [351]. Beyond DNS, Bechtold [21] argues that designing and controlling a namespace in general brings about "politics, policy, and regulation". Due to its similarity both in terms of structure as well as semantics to the DNS, we argue that an NDN namespace also requires proper governance to enable Internet-scale scope (see our previous work [83] for a thorough discussion).

Occasionally, it is required to have a set of names, addresses, *etc.* confined to a restricted context, *e.g.,* a local or private network. The NDN forwarding daemon (NFD), for example, uses the `/localhost/nfd` namespace for message exchange as part of its management protocol [422]. Such namespaces must also be defined as part of a namespace management governance.

10.3.2 Public-key-to-producer Binding and Public Key Management

Digital signatures build the basis of the NDN security model. As producers are bound to public keys through certificates, securing NDOs presuppose public key management. We propose a taxonomy for public key management based on *trust model* and *key management* (see Abbildung 10.3.2).

Trust Model. One of the main challenges in public key cryptography is the establishment of trust in the public key. Various trust models have been proposed to avoid manual out-of-band key exchange and verification by introducing trusted third parties (TTP) that generate and store key pair (*e.g.,* key escrow systems) or certify authenticity of a public key (*e.g.,* certification authorities). Here, a relying party (RP), *i.e.,* an entity that relies on the information included in a certificate to make a decision, for example, a data consumer, only needs to trust a limited number of TTPs and assess the trustworthiness of any public key based on policies of this trust model. Abbildung 10.3.3 depicts various trust models adapted from [375].

To establish TTPs, a relying party first needs to retrieve a set of trusted keys, denoting trust anchors (TA), in a bootstrapping phase. A trust model can define how trust is then deferred from a TA to any other TTP [26]. In DNSSEC PKI [444], for example, a single TA (root zone) is defined but RPs can decide to manually trust any other zone as their TA, whereas in Web PKI any CA that fulfills a set of requirements can be accepted as TA by operating system or browser vendors as maintainers of a *trust store* [349] (see for example Mozilla CCADB [471] or Apple root Certificate Program [285]). In contrast to DNSSEC, CAs in Web PKI are not limited in their certification so that a single compromised CA can issue certificates for any arbitrary domain name and any identity. To counter this, Certificate Transparency (CT) logs [214] were introduced so that anyone can track which CAs issues certificates for which domain names and subjects. Similar approaches to CT logs have been also proposed for NDN [38]. However, as we see later in Abschnitt 10.4, public key management solutions for NDN which unify the role of namespace principal and CA for a given namespace, are less dependent on such certificate logs. This is due to the fact that in such cases the relationship between a certification entity, *i.e.,* namespace principal, and the namespace it is authorized to certify is unique and unambiguous. So if an entity other than the namespace owner issues certificates for that namespaces, those certificates are by definition invalid and that entity is evidently misbehaving. Certificate logs here can, nonetheless, be used by namespace principals to detect misbehaving parent principals, *e.g.,* when a namespace principal delegates the same sub-namespace to two different owners.

A trust model also defines how valid trust chains between trust anchors and any arbitrary key can be constructed. In DNSSEC, there exists only a single verifiable path from any given key to a TA which is constructed from that TA, *e.g.,* root, to respective zone along the delegation path. In contrast, the cross-certification model of Web PKI with multiple independent trust anchors allows multiple valid trust chains that cannot be reconstructed by an RP without additional information from the subscriber. To address this ambiguity, a web server provides the complete trust chain to its client. Trust chains in NDN are built from bottom to top through iterative following of `KeyLocators` and as such pose a surface for availability attacks in which an adversary can trick an RP to start building a trust chain that does not terminate with a TA or contains loops, as we discuss in more detail in Abschnitt 10.4.

Key Management. Procedures and policies to create, distribute, store, rollover, and revoke keys are referred to as key management [394]. Most commonly, keys are generated by their owners and are only certified by a trusted entity. Key escrow systems in which a TTP is also responsible to generate private keys are an exception to this rule. In such systems the TTP generates both private and public keys practically defeating the non-repudiation property of digital signatures and creating a single point of failure in case of compromise.

Key distribution defines how RPs can retrieve public keys. In DNSSEC, keys of a DNS zone are bundled together as a `DNSKEY` RRset, in Web PKI, the web server provides its certificate (signed public key) to clients, CAs provide repositories that contain certificates of intermediate and root CAs, and browsers or operating systems maintain *trust stores* that contain all trusted CAs. In NDN, a certificate needs to be retrieved by its name just like any other NDO. The `KeyLocator` field of an NDO defines the name of certificate that can be used to authenticate that NDO.

Key rollover refers to the procedure of transitioning an old key with a new one while maintaining the existing trust relations. In Web PKI, rollover for leaf certificates are instantaneous while trust anchors transitions succeed out-of-band by trust store maintainers despite procedures defined for Root CA key rollovers in RFC 4210 [411, §4.4.]. Key rollovers in DNSSEC are defined for both TAs [464] and other keys [413].

Finally, in case of key compromises, the issuer or the key owner revokes the key to avoid its further use. Although keys can be cached and used as long as they are not expired, a relying party still needs to regularly check if a key is not revoked. Revoking keys in DNSSEC can succeed instantaneously but might lead to inconsistencies due to cached keys, specially for KSKs for which the respective `DS` record must also be updated in the parent zone (see Osterweil [186] for a thorough discussion). Wang and Xiao [256], for example, propose a method to roll over KSKs in emergency situations without introducing temporal inconsistencies. In Web PKI, the revocation status of a certificate can be verified through Certificate Revocation Lists (CRL) [297] or the Online Certificate Status Protocol (OCSP) [448]. Analogous solutions for NDN has also been proposed such as suicide directories [486], distributed OCSP for NDN [202], or the ledger-based revocation approach *CertRevoke* [261].

Improper key rollover and revocation can have negative security implications. RFC 7583 [413], for example, discusses how to roll over DNSSEC keys without introducing inconsistencies in the network considering possibly cached records and propagation delays. In NDN certificate validity is temporally bound. And the validity of a certificate is bound by the validity of other keys in its trust chain. The validity of a data packet, consequently, is determined implicitly by the validity of its producer certificate. Such temporal dependency not only impacts producers and consumers but also intermediate and caching nodes. Rolling over or revoking a key equals invalidating all data signed by that key. If a trust anchor or intermediate CA rolls or revoke their keys before they are expired, an adversary can use old, yet valid, keys for replay attacks and disturb availability. Even data and certificates that are cached on the network can lead to failed authentication in case of ill-timed revocation and rollovers. The validity period of NDOs in general also impacts caching. In NDN, caches keep an NDO for at least until its *freshness period* (a relative value) expires. It is, however, reasonable to purge an NDO from caches as soon as it expires even if it still is considered to be fresh.

10.4 Namespace and Key Management Approaches for NDN

In this section, we review different namespace and public key management schemes proposed for NDN, describing each according to the taxonomies introduced in Abschnitt 10.3 and summarized in Figures 10.3.1 and 10.3.2.

10.4.1 NDN Public Key Management (PKM)

Overview. In an NDN technical report, Yu [486] proposes a public key management method for NDN which also allows integrating namespace management based on naming semantics of certificate issuance and ownership. The author defines the NDN certificate format (preceding the current format [416]) and procedures to issue, verify, and revoke signatures and keys. Here, a certificate name is formatted as `/<identity>/KEY/<keyID>/<ver>`. The `identity` component serves to identify the public key owner, *e.g.,* `/org/ietf/alice`. The process of certification implicitly delegates the `/<identity>` namespace to the certificate owner to publish its own data, *e.g.,* revocation packets, as discussed below. To allow for multiple keys to be bound with the same identity, each certificate name is carries a `keyID` which (similar to a `DNSKEY` key tag) can be used to distinguish keys of the same owner. A version number is also included in a certificate's name in case the same public key is re-signed. The identity prefix is divided from key id and version number with a constant `KEY` name component. This approach has been (partly) implemented by NDN applications, *e.g.,* NDNfit [489].

Challenges. Similar to the current naming convention for NDN certificates, this approach requires that the issuer is authorized to publish under the namespace that actually belongs to the key owner, *e.g.,* under `/org/ietf/alice/KEY` prefix. Two possible alternatives can be imagined: (*i*) the issuer is both the namespace principal and the certification authority of the parenting namespace, *e.g.,* `/org/ietf`, or (*ii*) identities are prefixed with a namespace delegated to the issuer, *e.g.,* `/letsencrypt/org/ietf/alice`. Whereas the latter approach requires a separate infrastructure for namespace management, the former solution (as used in NDNFit [489]) entangles namespace management and public key management, unifying the role of namespace principal and certification authority. Here, namespace delegation succeeds through certification, *i.e.,* the owner of `/org/ietf` namespace can delegate `/org/ietf/alice` to Alice by signing her certificate under `/org/ietf/alice/KEY` namespace (see Abbildung 10.2.1). Alice, as her own namespace principal and CA, can in turn delegate any name under her namespace. Note that although certification and key management are defined along the hierarchical naming structure of NDN, the author does not further specify name and certification delegations from the root, *i.e.,* the '`/`' namespace.

The pitfall of unifying the role of namespace principal and CA is that any data producer who owns its dedicated namespace must also take care of key management. To take the burden of certificate distribution from the certificate owner, the author foresees *certificate hosts* which act as always-on, long-time caches for certificates. An owner can then designate certificate hosts to distribute its certificate(s) and is also responsible to make sure that the latest certificates are always stored by the certificate hosts. Yet, key life cycle management remains a duty of its owner. To revoke a certificate, its owner publishes a revocation certificate, *i.e.,* a self-signed version of the certificate, under the same name but with an appended `REVOKED` name component. Certificate owner must then actively respond to interests for the revocation status, *e.g.,* interests for `/org/ietf/alice/KEY/1/1/REVOKED`, using a signed NACK if the certificate is not revoked. The fact that the key owner and not the issuer is responsible for revocation and issuing signed NACKs can be advantageous for an adversary. Consider that

Eve has gained access to Alice's private key. Alice now wants to revoke its key but Eve can suppress revocation certificates if she manages to publish signed NACKs before interests for revocation status reach Alice. An alternative approach proposed by the author is to have a global "suicide directory" to register all revoked certificates resembling a combination of Certification Revocation Lists (CRL) [297] and Certificate Transparency Logs [214] of the Web PKI.

Beside revoking keys, single signatures can also be revoked in this approach. Given an NDO, the signer needs to regularly publish "signature status" under `/<identity>/SigStatus/<hash>/<ts>` with `hash` being the digest of signed NDO and `ts` the publication timestamp of the status. The content of the status denotes if the signature is revoked. Producers can delegate this process to a third party through a special certificate extension which is not further detailed by the author. The downside of this is that publishing any NDO obligates its producer to regularly publish a respective signature status *even* if no consumer is currently requesting that NDO. An alternative would be to extend the `SigInfo` of an NDO (see Abbildung 10.2.1) to contain a validity period similar to the validity period of DNSSEC signatures (`RRSIG`; see Abschnitt 10.A). Although this alternative could increase the network traffic, it would bring two advantages: (*i*) the producer needs to sign or re-sign data only if it is actively consumed, and (*ii*) the validity period of an NDO would define a hard and absolute deadline for caches to remove the NDO.

Finally, given an NDO, the process of validation succeeds as follows:

1. Verify that NDO name shares the same prefix with the certificate of the signing party as denoted in `KeyLocator` (assuming that certification and namespace delegation are unified)

2. Fetch certificate and verify that it has not expired, *i.e.*, the `ValidityPeriod` holds

3. Verify that the certificate is not revoked either by:

   a) Consulting the suicide directory or

   b) Confirming that no revocation certificate exists (signed NACK with an un-revoked signature; see above)

4. Verify that the NDO signature is not revoked

5. Validate the NDO signature using fetched certificate

This procedure is then repeated for any intermediate certificate (an NDO itself) by following the certificate chain until a trust anchor is reached. The validation starts with an NDO, moves along the delegation path, and ends with a trust anchor. To avoid multiple round trips to authenticate a single packet, the author introduces *key bundle* which contains all certificates required to validate an NDO. A key bundle, however, does not obviate the need to verify revocation status of included signatures.

Tabelle 10.4.1: Simplified trust schemata and exemplary names for /org/ietf namespace with a single trust anchor

Rules

#	Name pattern (data or certificate)		Name pattern (certificate)
1	<org><ietf><KEY>[id]	→	/org/ietf/KEY/1
2.1	<org><ietf><admin>[user]<KEY>[id]	→	<org><ietf><KEY>[id]
2.2	<org><ietf><admin>[user]<KEY>[id]	→	<org><ietf><admin>[user]<KEY>[id]
3	<org><ietf>[user]<KEY>	→	<org><ietf><admin><KEY>[id]
4	<org><ietf><~>([user])<>*	→	<org><ietf>([user])<KEY>[id]
5	<org><ietf><public><>	→	<org><ietf>[user]<KEY>[id]

Instances

#	NDO Name		Key Locator
1	/org/ietf/KEY/1	→	/org/ietf/KEY/1
2	/org/ietf/admin/bob/KEY/1	→	/org/ietf/KEY/1
3	/org/ietf/admin/eve/KEY/1	→	/org/ietf/admin/bob/KEY/1
4	/org/ietf/alice/KEY/1	→	/org/ietf/admin/bob/KEY/1
5	/org/ietf/~/alice/cv.pdf	→	/org/ietf/alice/KEY/1
6	/org/ietf/public/party.pdf	→	/org/ietf/alice/KEY/1

10.4.2 Trust Schema

Overview. Based on the observation that different applications might need to define different trust relationships, Yu *et al.* [262] propose "trust schemata", a name-based trust management system for NDN. The idea of trust management can be traced back to Blaze, Feigenbaum, and Lacy [26], who considered existing trust mechanisms, *e.g.*, PGP, as too narrow and inflexible and introduced a language to express security policies, to validate if a credential fulfills a policy, and to "defer trust" to third parties. As an established part of NDN security specification, schematized trust in NDN is a trust management system that defines both namespace as well as public key management in terms of trust rules and trust anchors. A trust rule is a mapping that defines the relation between an NDO name and the certificate name of its signing party. Namespace authorization in this approach are defined by rules that map non-certificate name patterns to certificate name patterns, and public key management by rules that map certificate name patterns to another certificate name pattern. Trust rules can be linked together to form a trust chain ending with a trust anchor. A trust anchor is denoted when a name pattern is mapped to a (set of) specific certificate name(s). A trust anchor here denotes both a namespace principal and a CA.

An example of trust schema is provided in Tabelle 10.4.1. In this simple scheme, Rule #1 defines a single trust anchor that controls /org/ietf namespace. Rules #2 and #3 define administrators as CAs for the producers authorized to names under *user* (`/org/ietf/~/[user]`) and *public* (`/org/ietf/public`) namespaces. And finally, Rule #4 and #5 authorize producers to publish under these namespaces.

Advantages. The main advantage of trust schemata is their expressiveness in defining namespace and key management policies. For instance, in our previous example, an administrator can only certify public keys and is not authorized to any namespace to publish non-certificate NDOs, while a non-administrator user may publish NDOs under specific namespaces but cannot delegate sub-namespaces or certify other users. Furthermore, as namespace and key management are not coupled, a single entity (using a single certificate) can be authorized to multiple namespaces (Rules #4 and #5).

Challenge: Key Revocation. Decoupling namespace and key management require careful design of trust rules to cater for certificate (or signature) revocations while avoiding potential security issues. In the following, using the example of revocation methods proposed by NDN PKM (from the same author, see Unterabschnitt 10.4.1), and CertRevoke [261], we discuss challenges of catering to key revocation when designing trust schema.

We start with a simple trust schema as depicted in Tabelle 10.4.1. When using NDN PKM, we see that a certificate owner is not allowed here to publish a revocation notification, thus, leaving the burden of revocation to the issuing CA. The pitfall, however, is that a (compromised) CA can revoke a certificate that it has not issued and still fulfill the trust rules. For self-referencing certification rules (*e.g.,* Rule #2.2), this property allows revocation of peering CAs as well. To elaborate this we use the example of instances in Tabelle 10.4.1. It can be seen that Alice cannot revoke her certificate by publishing a self-signed certificate under `/org/ietf/alice/KEY/1/REVOKED` as she is not authorized to publish under that namespace. She also cannot publish signed NACKs to indicate that her certificate is still valid. At the same time, Eve can publish a revocation certificate for Alice (Instance #3) despite that fact that Bob has issued the certificate (Instance #4). Eve can also even revoke Bob's certificate (Rule #2.2.) or suppress revocation certificates with signed NACKs in response to revocation status requests. Adding an extra trust rule to allow for self-signing revocation notifications as follows:

```
(<>*)<KEY>([id])([ver])<REVOKED> → (<>*)<KEY>([id])([ver])
```

would solve these issues without giving any unnecessary privileges to a certificate owner. Note that such rule still does not solve potential availability attacks discussed in Unterabschnitt 10.4.1. Another solution is proposed by CertRevoke is to explicitly include the revoker identity in a revocation packet and then utilize fine granular trust schema rules to limit certificate revocations to their exact issuers or owners. For example, the revocation message for a certificate under `/org/ietf/alice/KEY/1/issuer1` (see Abschnitt 10.2) can only carry the name `/org/ietf/alice/REVOKE/1/[issuer1|self]` and be signed by the issuer (if last component is the `issuer1`) or the owner (denoted by `self` name component). The downside of CertRevoke is that to check the revocation status, a relying party needs send out two messages (one with issuer ID and one with `self`) to make sure that neither has revoked the certificate.

Challenge: Authentication. Assuming that revocation procedures are defined, authenticating an NDO using trust schemata is similar to the procedure discussed previously (see Unterabschnitt 10.4.1) but instead of step 1, a relying party need to verify that the name of an NDO and its `KeyLocator` match at least one trust rule.

Furthermore, no a priori known authentication path may lead to unstable behavior. As the trust chain is built from leaf to root (in contrast to DNSSEC) but the authentication the other way around, an adversary can deceive a relying party to construct a trust chain that conforms to trust rules but do not end with a trust anchor. Self-referenced trust rules even allow creating loops and increase the length of the trust chain. This poses a serious attack surface for availability. For example, an adversary can publish an NDO under Alice's namespace signed with a fake certificate `/org/ietf/alice/KEY/2` attributed to Alice. This certificate can then be signed by an administrator whose certificate is signed by another fake administrator and so on. A relying party would then start authenticating the fake NDO by constructing a trust chain that never terminates. This problem can be solved by bundling the trust chain with an NDO (see key bundle in Unterabschnitt 10.4.1).

Challenge: Scoping and Synchronization. Finally, Trust schema suffer from two additional major shortcomings. First, the limited scope. It is not possible to delegate trust management, *i.e.,* having another entity define trust schema for a sub-namespace. Consequently, a relying party need to fetch the set of trust anchors and trust rules for every namespace that it aims to validate NDOs from. Second, no possibility of synchronization. If the trust schema of a namespace is changed at some point in time, there is no way for a relying party to know which set of rules apply to which set of NDOs. There is no temporal binding between an NDO and respective trust schema.

10.4.3 NDNSSEC

Overview. Using the example of DNS and observing changes in the past decades, Tehrani *et al.* [83] argue that namespace management, specially on an Internet scale, requires attending to organizational and political aspects. Based on this, the authors suggest using the DNS namespace for NDN to take advantage of existing infrastructure without a need to establish organizations, policies, *etc.* to attend non-technical aspects (*e.g.,* an equivalent of ICANN). Additionally, the proposed solution uses the DNSSEC PKI to secure namespace bindings and is also used for key management in NDN. Acquiring a namespace in NDNSSEC then becomes equal to a DNS zone delegation. For example, the zone owner of `www.ietf.org` is also the namespace principal of `/org/ietf/www` on NDN. Note that, although approaches to distribute keys and other data using structures similar to DNS has already been proposed before (see for example NDNS [279] and KRS [155]), NDNSSEC emphasizes on taking advantages of established technical and non-technical aspects of DNS and DNSSEC [83].

To realize key management, the role of namespace principal and CA for a namespace are unified. To authorize producers to publish NDOs, a namespace principal includes respective keys in its zone's `DNSKEY` RRset (see Abschnitt 10.A). In this approach, the name of an NDO and its `KeyLocator` share the same prefix that is the NDN equivalent of fully qualified domain name of the zone apex. The `KeyLocator` field uses a simplified NDN certificate naming (see Abschnitt 10.2) with `KeyId` being the digest of a `DNSKEY` and empty values for `IssuerId` and `Ver` name components. To fetch keys over

DNS, a consumer can either run a local DNS resolver [81] or rely on a designated trusted third party (TTP) for key retrieval and its translation into NDN certificates. The latter solution allows having namespaces which are only valid within a private network [83], but also allows the local resolver to publish certificates under namespaces that it does not technically own. Regardless of how DNS records are introduced to NDN, *i.e.,* through a bridge resolver or an authentication chain, the lifetime of a key is defined by its covering signature, *i.e.,* `RRSIG` (see Abschnitt 10.A). Respectively the certificates that are generated by the bridge resolver must reflect the validity period of the `RRSIG` that covers the containing key.

Challenge: Maintaining Trust Chains. To avoid retrieving DNS records over the Internet, the authors suggest that zone owners publish the complete authentication chain for their zones (similar to key bundles in Unterabschnitt 10.4.1) or to mirror DNS data using NDNS [4]. The former option, however, requires zone owners to publish a new chain every time any of the included records in the chain is invalidated (recall that DNSSEC signatures have a validity period) or a key is revoked. Furthermore, a zone owner must regularly scan for changes in its parenting zone as DNS(SEC) does not define a method to signal such changes. Authentication chains, just like certificates, must also be enhanced with a validity period which expires no later than the earliest expiration data of all involved signatures.

Key revocation in NDNSSEC is defined as removing a key from the DNS zone. And signature revocation for certificates is not required for keys, as each DNSSEC signatures (`RRSIG`) already carry a validity period that semantically corresponds to a "signature status" (see Unterabschnitt 10.4.1). Another advantage of DNSSEC is the set of well-defined key rollover procedures [413], [464] that also define rollover procedures for trust anchors that spares the need for out-of-band key exchange after an initial trust bootstrapping.

Challenge: Authentication and Authorization. Given an NDO, the authentication process is similar to the previously discussed procedure (see Unterabschnitt 10.4.1), but instead of step 2 and step 3, either a bridging resolver fetches and validates a certificate or an authentication chain containing the key is retrieved and validated. In the latter approach, the `KeyLocator` field carries the name of the authentication chain instead of a certificate.

The major drawback of NDNSSEC is caused by the fact that not single records, *e.g.,* a single `DNSKEY` record, but the set of same-type resource records are signed together. To validate if a producer is authorized to publish under a namespace, thus, the keys for all authorized producers must be retrieved to validate their covering signature. This poses a major scalability issue as the number producers for a namespace increases. To authorize a single producer to multiple namespaces, its public key must be replicated to all respective zones. Another shortcoming of NDNSSEC is its dependence on DNS and a lack of specific solution on how to mirror DNS data in NDN ecosystem.

Tabelle 10.4.2: Summary of namespace management approaches in NDN based on taxonomy in Abbildung 10.3.1

		Allocation						Governance	
	Structure	Granularity	Cardinality	Persistence	Security	Delegability		Scope	Organization
NDN PKM [486]	Hierarchical	Subspace	1-n	Transient	TA Signature	✓		Local	Centralized
Trust Schema [262]	Hierarchical	Arbitrary	1-n	Permanent	TA Signature	✘		Local	Centralized
NDNSSEC [83]	Hierarchical	Subspace	1-n	Transient	DNSSEC PKI	✓		Global	Centralized
IBC [268] & HIBC [105]	Hierarchical	Subspace	1-1	Permanent	PKG Signature	✘		Local	Centralized

10.4.4 Identity-based Trust

Overview. Zhang *et al.* [268] introduce a method to use NDN names, or more specifically their string representation, to generate cryptographic key pairs using *identity-based cryptography* (IBC) [452]. Here, a 'private key generator' (PKG) is in charge of issuing private keys for identities. In this approach, the producer must first consult a 'Name Registration Service' (NRS) to register a desired name. If the registration succeeds and the producer is authorized, the NRS also retrieves the corresponding private key (generated using producer's identity) from the PKG and forwards it to the producer. Namespace management in this approach is centralized and handled by the NRS which also coordinates private key issuance with the PKG. Relying parties can then use the public parameters of PKG to generate public keys for identities. An identity is included as metadata in an NDO, alongside the NDO name containing the PKG parameters, and can also be part of an NDO name. However, if the identity is not explicitly part of the name, *i.e.,* no unique name-to-producer binding is given, an authorized producer can publish NDOs with valid signatures under any arbitrary name.

Challenge: Issuance and storage. Key escrow, *i.e.,* the circumstance that an entity other than the key owner generates and stores keys, is a common weakness of all IBC approaches (see Abschnitt 10.3). Among the proposed solution, certificateless signature scheme has recently also been proposed for NDN [113].

Challenge: Scalability. Having only one instance of NRS and PKG, is an obstacle to scalability in this approach. To address this issue, the authors propose a hybrid scheme where local instances of NRS and PKG are responsible for smaller sections of the global namespace. The public parameters of each sub-namespace is signed by a set of CAs as part of a PKI that is trusted by relying parties, *e.g.,* DNSSEC. An alternative to address scalability is proposed by Hamdane *et al.* [105] based on hierarchical IBC (HIBC). Here, instead of a single global PKG or a set of domain-local PKGs, a root PKG is designated, and key issuance can be delegated to subordinate PKGs. A public key is then calculated from a given identity combined with the identity of its ancestors.

Challenge: Key Revocation. Key revocation in IBC is its main weakness. As a unique string is always mapped to a unique public key (given a set of public parameters), and keys, public or private,

Tabelle 10.4.3: Summary of public key management approaches in NDN based on taxonomy in Abbildung 10.3.2

		Key Management				
	Trust Model	Issuance	Distribution	Storage	Rollover	Revocation
NDN PKM [486]	Hierarchical	Namespace Principal	Owner / cert hosts	Owner	✘	Owner
Trust Schema [262]	Hierarchical	Designated CA	Owner	Owner	✘	✘
NDNSSEC [83]	Hierarchical	DNS zone owner	Owner / DNS	Owner	RFC5011 [464] RFC7583 [413]	Issuer
IBC [268]	Basic	PKG	NA	Owner / PKG	✘	✘
HIBC [105]	Hierarchical	PKG	NA	Owner / PKG	✘	✘

are not temporally bound, a key is considered valid as long as an identity is considered valid and vice versa. Revoking a key, thus, means revoking an identity and in turn revoking all data produced and signed by that identity. To address this shortcoming, Zhang *et al.* [268] propose to append a timestamp to identities and to consider that identity as expired as soon as a predefined amount of time spans. For relying parties, however, there is no reliable way to verify if a key is revoked at any point in time (*e.g.,* compared to revocation lists), while a compromised key can sign data as long as the validity window is not expired. In case of HIBC, Hamdane *et al.* [105] note that the revocation problem can be solved by simply revoking the public parameters of the PKG which generated the compromised key. The procedure of revoking a single PKG is not elaborated by the authors, nonetheless, revoking a PKG equals to revoking all keys generated by that PKG.

Authenticating an NDO in the approach proposed by Zhang *et al.* [268] succeeds as follows:

1. Extract the name of NDO containing public parameters from NDO's metadata
2. Retrieve public parameters and verify its validity (signed by trusted PKG)
3. Extract identity name from NDO's metadata and construct corresponding public key using retrieved parameters
4. Validate NDO using generated public key

In the hybrid scheme, instead of step (2), the public parameters are fetched from a trusted pre-defined PKI. Note that this procedure does not consider signature revocations. In the hierarchical approach [105], the `KeyLocator` field contains the name of the NDO with the public parameters of responsible PKG. These parameters are signed by the parent PKG and having the root parameters suffices to generate public key for an identity on any arbitrary level of the hierarchy. To authenticate an NDO, the following method applies:

1. Extract the name of certificate containing public parameters from NDO's `KeyLocator`
2. Fetch certificate and verify that it has not been expired

3. Verify that the certificate and the PKG parameters are not revoked

4. Validate NDO signature using fetched certificate

Similar to the procedure elaborated in Unterabschnitt 10.4.1, this procedure is repeated for all intermediate certificates until the root PKG is reached. Note that signature revocation is neither addressed here.

10.4.5 Summary

The landscape of namespace and public key management in NDN is diverse and far from being standardized. We summarize our findings based on our taxonomies (see Abschnitt 10.3) in Tabelle 10.4.2 and Tabelle 10.4.3 and now discuss comprehensively pitfalls of namespace and public key management in NDN.

Namespace Management. All approaches propose a local namespace, *i.e.,* a context with limited scope, except NDNSSEC. NDNSEC relies on the DNS as its underlying namespace with globally unique names, all other solutions are only suitable for local namespace management due to lack of well-defined name allocation and collision avoidance on a global scale. All but IBC and HIBC cater for allocating names to multiple producers. Identity based namespace management alongside trust schemata define permanent name allocations which cannot be delegated further. To void allocations in (H)IBC, either the PKG must rollover its parameters (both public and private) and consequently void all its generated keys (as proof of name ownership) or a namespace principal (*i.e.,* producer) must acquire a new identity and accordingly a new namespace. For Trust Schema, all allocations hold as long as the local schema is considered as valid. As such, trust rules are defined once and following modifications equal to defining a new Trust Schema. In addition, absence of methods to link Trust Schema with another prevents namespace delegations to other principals with their own set of Trust Schema. As the relation between a name and its owner is not intrinsic in any of the proposed approaches, name allocations are all secured by some form of cryptographic attestations. For (H)IBC, creating a private key from a namespace prefix (identity) is the proof of allocation by the PKG (or indirectly through NRS). For Trust Schema and NDN PKM, a trust chain that ends with a trust anchor (TA) serves to secure name allocations. And in NDNSSEC, all allocations are secured by DNSSEC.

Public Key Management. All but IBC rely on a hierarchical trust model. Beside being advantageous for scalability, a hierarchical trust model which constrains key management can also improve security. For example, in NDN PKM [486], having a CA (same as namespace principal) compromised does not have any negative impacts on other namespaces. In NDNSSEC [83] and NDN PKM the roles of namespace principal respectively DNS zone owner are the same as the CA for that namespace. Trust Schema [262] decouples these two roles and uses certificate names in trust rules to designate CAs for a set of names. In contrast to the other solutions, IBC and HIBC do not certify public keys but generate private keys. This poses a challenge to the non-repudiation property of signature as at least theoretically both PKG and key owner can generate signatures for the same identity. At the same time, public keys in (H)IBC do not require distribution and can be generated by relying parties on the fly using only public parameters of the (root) PKG. It is noteworthy that only NDNSSEC allow for key rollovers (although not discussed by the authors) by relying on DNSSEC methods of rollover [413], [464].

10.5 Conclusion

In this paper, we contributed a systematization of knowledge on namespace and public key management. We surveyed more than 30 NDN applications, proposed at ACM ICN 2015-2021 and related venues, to summarize their assumptions and requirements on naming and security. Lack of consensus on naming and namespace management among existing applications poses a potential cause for conflict when different applications with conflicting naming constraints should run in the same context, *e.g.,* in a global scope. Similarly, missing well-defined public key management, *e.g.,* to establish trust or to revoke compromised keys, poses a serious hurdle to achieve confidentiality, authentication, or access control that are based on public key cryptography. We analyzed current research that introduces approaches to implement naming and key management in NDN. Our comparison was structured along two taxonomies. We observed shortcomings in namespace management regarding authorization (constrained allocation), scalability (lack of delegability), and scope (limited governance policies and procedure).

Open challenges on public key management relate to simplifying bootstrapping procedures (as basis to establish trust relations) and providing procedures for key rollover and revocation (as part of key management). We hope that this paper serves as a systematic guide for both improving existing approaches and highlighting integral features and pitfalls that might arise when designing future solutions.

Acknowledgment. We would like to thank our shepherd Lixia Zhang and the anonymous reviewers for their detailed and helpful feedback. This work was supported in parts by the *German Federal Ministry of Education and Research (BMBF)* within the projects *Deutsches Internet-Institut* (grant no. 16DII111) and *PIVOT*.

Appendices

10.A DNS and DNSSEC

The domain namespace [410] is managed in terms of *zones*, *i.e.,* sub-trees within the DNS hierarchy, and a zone administrator can *delegate* any branch under its sub-tree to another authority. For example, `ietf.org.` ("IETF Trust") has been delegated by the management authority of the zone `org.` ("Public Interest Registry"). DNSSEC [444], the DNS security extensions, introduce data origin authentication, data integrity, and authenticated denial of existence to DNS. DNSSEC secures domain names by binding special RRs that allow for cryptographic verification to domain names. These include public keys (`DNSKEY`), digital signatures (`RRSIG`), and delegation signers (`DS`) to reflect the DNS hierarchy.

DNSSEC keys are used to sign set of RRs of the same type (*RRsets*) instead of single RRs; and the same RRset can be signed by different keys at the same time. When a relying party recursively resolves a name, the RP validates a zone as *secure* if (*i*) the RRsets are signed by at least one valid key included in the `DNSKEY` RRset, and (*ii*) a *chain of trust* can be established from a trust anchor (commonly the root zone) to that zone. The root zone uses a *well-known* self-signed key signing key (KSK) to sign its `DNSKEY`s and uses its zone signing key (ZSK) to sign `DS` records for top-level domains (TLD) such as `.com`. Similarly, when a second-level domain name (SLD) is delegated from a TLD to a DNSSEC-enabled zone, at least one `DS` record is created at the TLD zone upon the request of the delegated zone authority. This iterative procedure creates unique verifiable path from the root zone to any arbitrary DNSSEC-enabled zone. Although there are no constraints for an RP on designating arbitrary zones as trust anchors, in practice RPs use the root zone as the single trusted third party in DNSSEC PKI. A zone can only delegate sections of DNS namespace under its subspace, a zone owner can only certify keys of its immediate child zones.

10.B Web PKI

Based on X.509 certificates [297], Web PKI enables identification and authentication of service endpoints over the web. Here, X.509 certificates are used to bind a *subject*, a real-world identity, to a public key. Originally, a subject was intended to be represented through a *distinguished name*, a unique entry in a global hierarchical directory [372]. Such global directory, however, never came into existence, making distinguished names prone to collision and not suitable for unique authentication or identification of the certificate holder. This shortcoming is not an issue in Web PKI as X.509 certificates are also bound to domain names (as globally unique identifiers). Here, Certification Authorities (CA) certify bindings in a certificate through their signature. A CA can also delegate its authority to *intermediate* CAs, which issues a subscriber a certificate. To validate a certificate, RPs, *i.e.,* web clients, must first acquire certificates for a set of trustworthy CAs through an out-of-band channel. These certificates form the so-called *trust store* and are maintained by browser vendors, operating system companies, *etc.* and not RPs themselves. In general, any trusted CA can certify a public key for any arbitrary subscriber and a single subscriber can be certified by multiple CAs; consequently and in contrast to DNSSEC, there is not necessarily one unique verifiable paths from a given subject or domain name to a CA.

Various solutions have been proposed to remove the ambiguity regarding public keys and responsible CAs for a specific domain name. DNS Certification Authority Authorization (`CAA` RR) [357], for example, allows a domain name owner to explicitly specify which CAs are allowed to issue a certificate for that domain name. Alternatively, the DNS-Based Authentication of Named Entities (DANE) allows describing certificates that are expected to be provided a service endpoint (using `TLSA` RRs [363]). DANE practically extends a DNSSEC trust chain to Web PKI. Finally, an RP validates a certificate as long as there is signature chain from that certificate to a certificate in its trust store.

Kapitel 11

Authenticated Communication in Crises: Toward an Infrastructureless Trust Model for Challenged Networks

Abstract

Natural as well as human-caused disasters and catastrophes easily lead to chaos. Effective crisis communication and informing the public about the ongoing situation can reduce chaos and maintain social resiliency. Communication, however, relies on a working physical infrastructure, which usually is broken in the incident area. Whereas first response teams and authorities benefit from special communication equipment, civilians do not and experience longer periods of being disconnected from the outside world. Even if messages come through occasionally, the communication is too intermittent to allow for channel-based trust models. TLS, for example, requires access to trusted third parties to authenticate data.

In this position paper, we argue that end-to-end communication conflicts with disaster scenarios. We propose an approach that leverages information-centric networking (ICN) to authenticate risk and crisis communication in loosely connected communication systems. Our proposal makes use of spatiotemporal decoupling of data from their producers based on ICN for optimal message propagation in fragmented networks, while introducing a trust bootstrapping phase to enable off-line authentication. The data-oriented security model of ICN is used to tailor a trust model specifically for scenarios during which access to trusted third parties or data owners is not given and messages are relayed through untrusted parties.

11.1 Introduction

Severe disasters and catastrophes almost always lead to communication disturbances by damaging the infrastructure, by causing power outages, or due to delayed and unattainable repairing efforts [183]. Whereas disaster relief teams are in advantage in terms of communication means through special equipment such as satellite phones and long range radio systems, civilians are susceptible of becoming isolated and getting cutoff from external communication. Due to the extraordinary state of the

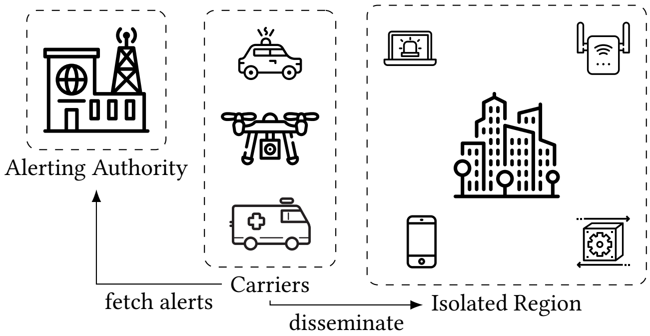


Abbildung 11.1.1: Use case Scenario: Alert Dissemination

situation during such incidents, it is as critical as ever to guarantee *risk and crisis communication* in face of disasters to maintain social resiliency [150], [200].

Effective crisis communication requires, among others, timely information provision from trusted and credible sources. In challenged or fragmented networks, however, it is unreliable to utilize existing Internet infrastructure, as targeted endpoints might be unreachable or down altogether. Simply put, the end-to-end paradigm of the Internet does not adapt well in disaster and crisis response phase. Even if limited communication is made possible, *e.g.,* through ad hoc, delay and disturbance tolerant, or software defined networks, typical security measures such as public key infrastructure (PKI) require access to trusted third parties (TTP), which cannot be guaranteed. Nonetheless, authenticating messages in such scenarios remains a crucial and necessary step.

In this paper, we focus on securing messages, *e.g.,* warnings, issued by authorities which can be authenticated off-line regardless of how a message is retrieved (see Figure 11.1.1).

The remainder of this paper is structured as follows: a brief overview of ICN and respective trust management is given in **§ 11.2** followed by a general concept on how to realize off-line authentication for disaster response in ICN in **§ 11.4**. A concrete solution based on named-data networking (NDN) [265] is introduced in **§ 11.5**. § 11.6 discusses related work, and § 11.7 summarizes our findings.

11.2 Background

Information-centric networking introduces a paradigm shift from *host-centric* networking to information by decoupling data from its producer both in temporal and spatial dimensions. Instead of assigning names to addresses and addresses to hosts to forward packets between them, in ICN the network is responsible for discovery and retrieval of data packets which are identified through unique names. ICN nodes are able to cache content for future provision without relying on the original producer.

The new networking paradigm of ICN requires new security perspectives. In contrast to securing channels between hosts such as in *HTTP secure* (HTTPS) or *secure shell* (SSH), ICN secures content independent of communication participants. A common approach in securing data is to use digital signatures over secure hashes of the data. To achieve authenticity, asymmetric cryptography signs data

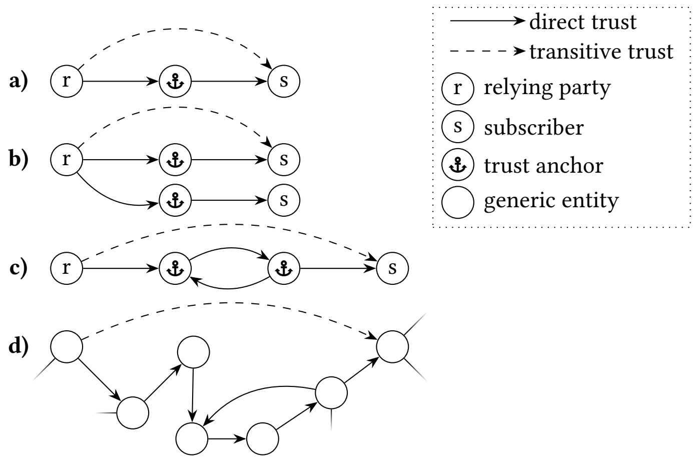


Abbildung 11.2.1: Trust models—a) Basic Trust Model, b) Multiple Trust Anchors, c) Cross Certified Trust Anchors, d) Web of Trust.

using a private key and verifies the signature using the corresponding public key. Public keys are commonly available in form of certificates. A certificate comprises a public key and metadata signed either by the owner (self-signed) or by a trusted third party (TTP) and is leveraged for signature verification and consequently data origin and non-repudiation authentication. To simplify key management tasks and policy enforcement, a *trust management framework* [26] can be used. Such framework defines trust relations by specifying which parties can issue and certify credentials. *i.e.,* TTPs, and enforces policies by putting constraints on actions of credential holders. Respectively, trust is established transitively through a TTP between a *relying party* and a *subscriber*, which uses the services of the TTP [473]. Beside the simple trust model, depicted in Abbildung 11.2.1a, alternative models such as multiple trusted parties, Abbildung 11.2.1b, and cross-certified trust anchors, Abbildung 11.2.1c, are possible. A generalization of these models is given in *Web-of-Trust* (WOT) [492] in which each party acts both as an authority and a relying party, see Abbildung 11.2.1d. Trust is considered to be established (to some degree) between two parties if there can be found a trust relation between them.

11.3 Requirement Analysis

Before eliciting functional and non-functional requirements, the following scenario is used to illustrate a typical use case of our proposed approach: given a geographically restricted disaster-stricken area in which communication channels between authorities and civilians are disturbed or cutoff altogether, it should be possible to (*i*) carry messages from authorities to civilians over multiple hops, and (*ii*) provide authentication mechanisms which are infrastructure-independent and only require information fetched in a preceding trust bootstrapping phase. Abbildung 11.1.1 visualizes such use case in its simplest form.

Stakeholders in this scenario are (*i*) alerting authorities as data producers, (*ii*) off-line and out-of-band data carriers, and (*iii*) data consumers ranging from consumer electronics to IoT devices and local network routers. Functional requirements are respectively defined as follows:

R1 Decoupling data from respective publishers.

R2 On and off-path caching.

R3 Passive and active data discovery.

R4 Data retrieval by its name (and not through its host).

Non-functional requirements are given below:

N1 Compatibility.

N2 Integrability.

N3 Scalability.

N4 Security.

N5 Resilience.

N6 Fault-tolerance.

N1 and **N2** emphasize the importance of compatibility with established and common protocols, *e.g.*, *common alerting protocol* (CAP) [480], **N3** foresees the need for scalability in face of increasing data publishing frequency and consumer count, **N4** denotes the need for mechanisms to secure data, **N5** is derived from the fragile nature of given situation immediately after disaster and crises, and finally **N6** caters for failures in involved networking nodes.

11.4 Concept

The goal of this work is to conceptualize an approach which allows securing messages, their dissemination in challenged networks, and finally providing means for off-line authentication. All functional requirements are already covered when using ICN as networking infrastructure (see Abschnitt 11.2). Regarding non-functional requirements, we focus on **N4**, namely security, while leaving the rest for future work.

We consider a data packet to be secured if the following can be guaranteed: (*i*) data integrity, (*ii*) data origin authentication, and (*iii*) non-repudiation verification. In other words, it should be possible to confirm that data has not been modified during transmission, its origin is traceable, and it can only be originating from its producer and no one else. A producer in this context is a real-world entity, *e.g.*, an alerting authority, identified by its digital credentials. Whereas mere technical solutions can be used to realize data integrity verification (by secure hashes) or non-repudiation attestation (by asymmetric cryptography), origin authentication requires organizational effort to bind digital credentials with real-world identities. To bind real-world identities to public keys, certificates are enhanced with metadata, *e.g.*, using *organization* field in X.509 certificates [297], and the authenticity of the binding can be attested by a TTP.

Interest Packet
Name
Guiders
Nonce
Application Params

Data Packet
Name
Metadata
Content
Data Signature

Abbildung 11.5.1: NDN Interest and Data Packets

As the emphasis of this work lies within off-line authentication, consumers need to fetch *all* necessary certificates during the bootstrapping phase *before* a disaster strikes. This way no on-line communication is required during authentication to fetch missing certificates. This turns out to be a non-trivial task, since a wide spectrum of non-governmental, private, and international entities are involved in disaster management (DM) beside the local major parties, such as police forces, fire departments, and emergency medical services.

Adequate trust bootstrapping would entail two phases: (*i*) a discovery, and (*ii*) a certificate retrieval phase. In discovery phase, a consumer should be able to fetch a list of DM organization which in turn maintain a set of certificates of their authorized producers. The inquiry is context-aware and results in discovery of only spatially related authorities, including both local and global organizations with a mandate for DM in that specific region. Finally, the retrieval phase is an iterative process by the consumer through which the set of certificates for authorized producers is fetched from the discovered organizations. To be practical for consumers, we propose the highest civil protection body of each country, *e.g., Sécurité Civile* in France, *BBK* in Germany, or *FEMA* in the USA, to maintain the list of top-tier DM organizations. It would suffice for consumers to obtain the certificate of civil protection organization securely to initiate the trust bootstrapping phase.

11.5 Proposed Solution

In this section, we introduce a proof of concept that fulfills all functional and non-functional requirements leveraging *named-data networks* (NDN) [265] as ICN core and NDNSSEC [83] for namespace management.

NDN is a popular ICN approach supporting a hierarchical naming scheme. Data packets in NDN are identified by their names, *e.g.,* `/com/example/index.html`, and can be retrieved using *interest messages*. Intermediate nodes forward interest messages to producers or hosts, which may have cached copies of the desired data. Data packets can be secured using asymmetric cryptography, and a signed packet contains all information necessary to fetch the corresponding certificate(s) required for authentication. For this NDN introduces the `KeyLocator` field (part of Digital Signature). A certificate in NDN is an ordinary data packet which carries a public key. To distinguish certificates from ordinary data packets, NDN adds special constraints on the naming of the certificate packets. The standard naming convention for certificates in NDN is as follows: `/<prefix>/KEY/<id>/<issuer>/<ver>` [272] where `prefix` denotes the certificate namespace, `id` the ID of key it carries, `issuer` the ID of the issuer, and `ver` the certificate version. The structure of interest and data packets in NDN is depicted in Abbildung 11.5.1.

NDNSSEC is a security extension which relies on DNS to realize namespace management in NDN. It partitions the global NDN namespace into smaller management units, *zones*, and provides mechanisms to verify if a producer is authorized to publish under a zone or not. To this mean, a DNS zone apex, *e.g.,* `example.com.`, is *ndnified* [279] into its equivalent *reverse slash separated* notation, *e.g.,* `/com/example`, and is used as name prefix for names published under that zone in NDN. The possibility of representing a DNS zone apex in its equivalent NDN notation and vice versa enables the outsourcing of zone management in NDN onto existing DNS ecosystem. This way not only technical aspects of namespace management, but also non-technical issues, such as binding names to real-world identities or solving trademark conflicts, are taken care for by DNS related organizations, such as ICANN. To authorize an NDN publisher to publish under a given zone, the respective zone owner must register a public key of a producer as a `DNSKEY` record under its authoritative name server. Prior to publishing a packet, the producer would sign it using its private key, and set the `KeyLocator` respectively, that is with zone apex as `<prefix>` and its public key digest as `<id>`. To authenticate a packet, consumers consult the DNS to check if any of the listed public keys by the zone owner has been used to sign the packet or not [83].

Our proposed solution presupposes the following as given: (*i*) both data producers and consumers have access to the same logical NDN instance, (*ii*) consumers access the network through a designated application which is already preconfigured with zone apexes, *e.g.,* `/gov/fema`, and certificates of the highest civil protection bodies for each country, and (*iii*) participating DM organizations follow NDNSSEC policies to authorize producers under their respective zones.

The general workflow of our solution comprises three stages as follows:

1. **Trust Bootstrapping**: The optimal phase to initiate trust bootstrapping is prior to disaster, *e.g.,* in disaster preparedness phase. Depending on its current location, a consumer would first need to fetch the list of authorized DM organizations. To this mean an interest message is dispatched for the list of zone apexes of authorized organizations under the name `/ao` prefixed with the namespace of respective civil protection organization. For example the list of authorized DM organizations within the United States can be retrieved by dispatching an interest for `/gov/fema/ao`. Note that the response is not necessarily provisioned by the data owner but any other reachable node which happens to have a copy of that packet. The authenticity of the response can be verified using the preloaded certificates in the application of the consumer.

   Iteratively, the consumer fetches the list of public keys authorized by each organization using NDNSSEC. This step can be done using a single round trip per organization by simply querying all `DNSKEY` records listed under its authoritative name server. Finally, the digest of public key for each entry is calculated and is stored in a local key registry alongside the zone apex and the key itself.

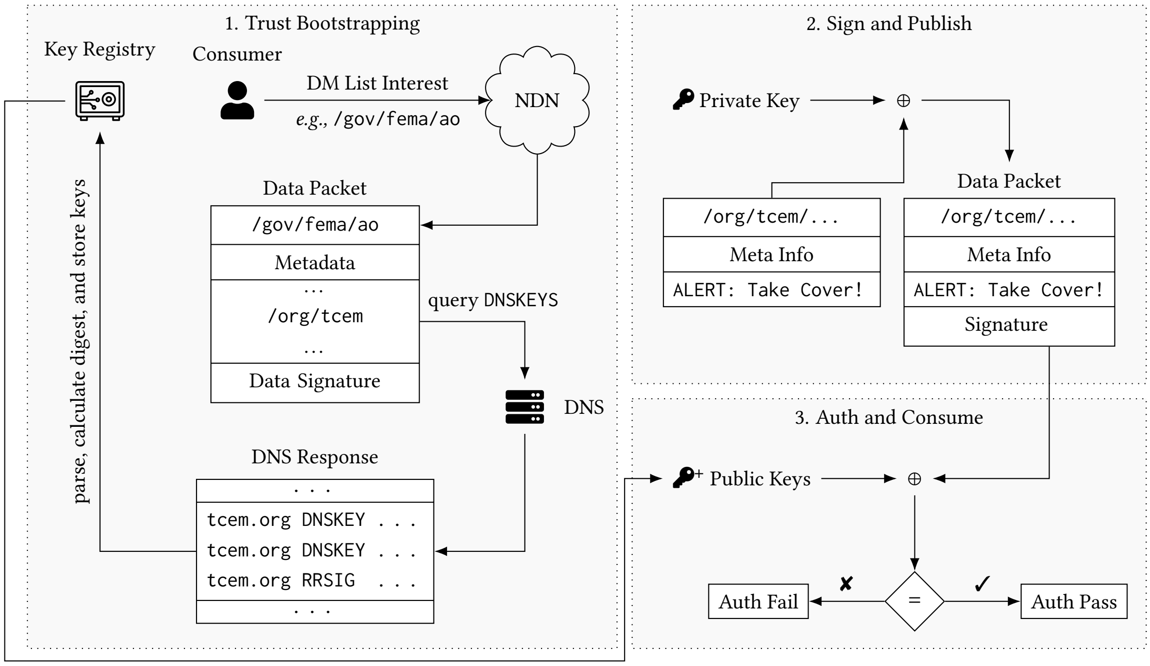


Abbildung 11.5.2: Overview of Workflow: from Bootstrapping to Authentication

2. **Data Publishing**: Published data must conform to the NDNSSEC workflow and policy. An authorized producer must sign its data and set the `KeyLocator` prior to publishing.

3. **Data Consumption and Authentication**: During disaster response it is expected for consumers to retrieve messages from DM organizations relayed through untrustworthy parties. Given a data packet, a consumer uses NDNSSEC to extract the respective zone apex from packet's name and its signature info block. The zone apex is used to search the local key registry, which is propagated during the trust bootstrapping phase, for authorized keys. The key digest included in the packets `KeyLocator` field is used to filter available keys in the registry.

The general overflow is summarized in Abbildung 11.5.2.

11.6 Related Works

The benefits of utilizing ICN in disaster management has been a subject of attention in recent years. Tyson et al., for example, highlight improved resiliency and superior disruption tolerance [245], while Seedorf et al. emphasize the spatiotemporal decoupling and data-oriented security schemes of ICN [220] as beneficial for disaster management. Projects such as *GreenICN* [235] and *UMobile* [212] are a few examples which leverage ICN specifically for disaster management use cases.

At the same time and in spite of widely recognized need for suitable trust models in disaster scenarios [449], [220], [245], only a few approaches have been proposed so far which address both origin authentication and identification. Seedorf et al. [219] propose a decentralized mechanism based on

WOT (see Abbildung 11.2.1d) to bind names to real-world identities in fragmented mobile networks. The objective here is to assess trustworthiness of *on-behalf-of* messages which are received from unknown third parties. Tagami et al. [236] leverage *identity-based cryptography* to avoid the need for persistent access to TTPs for authentication. In this approach, the identity of a producer is used to generate its public key using only public parameters retrieved from a trusted *private key generator* (PKG) (see [452]). A consumer can then generate public keys for an arbitrary producer on the fly, given that the corresponding private key is generated by a PKG known to and trusted by the consumer.

11.7 Conclusion

In this paper, we proposed a solution for message propagation and off-line data-origin authentication in fragmented and intermittent networks. Whereas the end-to-end principles of the Internet are considered as an obstacle, the spatiotemporal decoupling of data from hosts and the data-oriented security paradigm of ICN are enablers for quick message propagation and authentication in scenarios with impaired communication infrastructure.

Our future work comprises the implementation of the proposed solution and its evaluation. An analysis of time and space complexity of trust bootstrapping phase as well as the authentication process is to be undertaken. Furthermore, the organizational overhead for involved parties has to be studied to assess the integrability and interoperability of our approach in existing workflows of DM organizations.

Conclusion and Outlook

In this thesis, we set out to provide trustworthy communication using Web technologies when the data producer is unavailable. This seemingly paradoxical undertaking was made possible by decoupling trust from hosts and binding it directly to data, establishing what we call *object trust*.

We have evaluated existing Web security technologies and modeled a trust framework that integrates these technologies while taking into account the organizational and political aspects of trust (Teil I). Based on our own measurements of Web security protocols (Teil II), we touched on the problems that these approaches face when an end-to-end network path is not given, for example, due to intermittent connectivity. We extended the concept of object security to realize object trust and proposed to use Information-Centric Networks (ICN) as a basis. After evaluating ICN in terms of feasibility and suitability, we investigated how Web technologies can be adapted for ICN and provided a proof of concept (Teil III).

Security and Trust on the Internet

In Chapters 1 and 2, we proposed a model for enforcing trust relations over the Internet. In Kapitel 1, we provided a taxonomy for understanding trust in digital communication, which can also be used to guide practitioners when considering trust relations with users. In Kapitel 2, we discussed how existing Web protocols and infrastructures are sufficient to establish trust relations, and proposed an algorithmic evaluation of security and trust. Our model of trust extends to the three dimensions of resolution, authentication, and transaction security, and relies on TLS, DNSSEC, and the Web PKI.

However, our active Internet-wide measurements revealed suboptimal deployment and usage. In Kapitel 3, we analyzed 15 years of DNSSEC data from more than $10M$ domains using a novel approach to model DNSSEC key transitions and evaluate the security impact. We developed a temporal model to capture key life cycles and transitions, and introduced an approach that can reconstruct the temporal characteristics of keys and transitions from discrete measurement data points. We found large discrepancies with RFC standards, but also a decreasing rate of key management errors over the years. We then extended our measurements to the Web PKI and the integration of DANE and CAA. We developed a systematic measurement approach in Kapitel 4 by reviewing existing literature and drawing on our own experience. In Kapitel 5, we scanned over $4M$ of popular domain names and found that, besides ubiquitous support for TLS and authentication using Web PKI certificates, DNS-based security protocols are rarely used (5% CAA and 1% DANE) and are often plagued by misconfigurations. In Kapitel 6, we increased our focus on trust-critical infrastructure, specifically the Web presence of

US Alerting Authorities, German emergency services, and foreign missions of UN states. Despite the increased need for secure and trusted communications, we are not seeing a noticeable increase in security deployments. On the contrary, we observe a decrease in the use of Extended Validation certificates for identification and an overall low sensitivity to security measures. In summary, we observe that authentication using Web PKI certificates has become ubiquitous, while DNS-based security protocols still have low penetration rates. Among these, DNSSEC and DANE are used reluctantly, as they incur additional costs with minimal to no benefit due to lack of support on user clients. But even CAA, which explicitly aims to reduce operational costs by making DNSSEC optional, suffers from the same fate of insignificance. In Kapitel 7, we took a close look at the CAA specification and its implementation. We found that ambiguous CAA semantics, unbounded policy scoping, ambiguous identifiers, and unverifiability by third parties are among the causes that affect its effectiveness. We find flawed implementations by both users and certification authorities. Our findings are relevant for future security protocols based on DNS(SEC).

With the knowledge of the effectiveness and practical use of Web technologies in realizing online trust relationships, we then moved on to develop a concept for object trust based on Information-centric Networks. We started with a comparative analysis of ICN with respect to the Internet in Kapitel 8. We showed how replacing addresses with names on the network layer requires several non-technical aspects and how these aspects might differ from their current manifestations on the Internet. Specifically, we discussed the implications in terms of having names as commodities, the resulting power structures, the factors that amount to trust, the need to establish accountability, and finally, private communication over the ICN. In Kapitel 9 we showed how the use of DNS(SEC) can bring global namespace management to ICN without the need for a trust bootstrapping phase as postulated in related work. Kapitel 10 extended the scope of our evaluation to public key management in order to have a comprehensive view of trust in the ICN. Based on a survey of over 30 proposed applications, we showed how the existing namespace and public key management as building blocks of trust relationships are unsuitable for global deployment and lack practical measures for key rollover and revocation. In Kapitel 11 we showed how these challenges can be avoided by integrating the existing Web PKI and DNS(SEC) ecosystems into the ICN and enforcing trust relations as we do on the Internet.

Future of Trusted Communication

As we continue to digitize everyday and critical services, the need for trust measures increases. Object security is a suitable candidate for trustworthy data exchange, yet it has its own limitation regarding scalability, adaptation, and effectivity that we discuss below.

Scalability. Our approach to object security focuses on authentication and integrity using digital signatures. Within this framework, scalability is within reach as users need to establish a set of trusted keys in advance, *e.g.,* during a bootstrapping phase. If confidentiality is a concern, the key exchange must be extended to allow the data producer to preencrypt data for each recipient and to allow for spatial and temporal decoupling of data. This requires additional interaction between data producers and consumers beyond a bootstrapping phase. Additional mechanisms are also required if features such as forward secrecy are desired. This is a non-trivial challenge that remains to be addressed.

In addition, issues such as delegated authentication and encryption are scalability accelerators that can be achieved by reducing the burden of security on data producers. DNSSEC muti-signer, for example, is a novel approach to DNSSEC that allows multiple entities to sign DNS records without the name owner losing ultimate control of their namespace. Signed HTTP Exchanges is another example within the Web ecosystem that allows data publishers to allow other parties to securely distribute their content while still allowing users to validate the authenticity, integrity, and confidentiality of the data. The lessons learned from such approaches can be generalized in the context of scalable object security.

Adaptation. A comprehensive approach to online trust requires the involvement of various stakeholders such as ISPs, CAs, browser and operating system vendors, and so on. In the context of object security, this means providing mechanisms and interfaces to retrieve relevant information so that service providers can associate it with packets, and users can make informed trust decisions based on packets alone, without having to implicitly rely on the correct behavior of intermediaries. To facilitate object security, stakeholders need incentives. We discussed the conflicts that may arise when introducing a new paradigm such as object trust, specifically in the context of ICN. Our analysis was of a general nature; future work should investigate incentives and tussles in scoped scenarios with heightened security requirements, such as emergency alerting.

Incentives alone, however are not always sufficient as changes to the large ecosystem of the Internet, even small improvements, will surely face resistance. For one thing, stakeholders may find the proposed changes in conflict with their own interests. In practice, only organizations with a high market share and adequate resources set the tone for policies that impact trust and trustworthiness on a global scale. Political and legal approaches, which essentially force stakeholders to act in a certain way, can counter such power monopolies and are in fact effective tools that enjoy greater legitimacy than policies set by a corporation or private organization. In addition, decisions made through democratic processes tend to take into account and respect cultural aspects that are fundamental to trust relationships. In order to improve trust on the Internet, a more active role for politics is needed in the future.

Effectivity. Finally, there is a need to close the security awareness gap. We have found that users and service providers alike are not always well informed about security measures that can protect them and their services. Online monitoring and notification systems can be critical improvements, especially real-time systems that provide immediate feedback. An integral aspect of this is making the data easy to understand and pointing to concrete solutions.

Abbildungsverzeichnis

Tabellenverzeichnis

List of Acronyms

AA	Alerting Authority
AoT	Act of Trusting
API	Application Protocol Interface
DtT	Decision to Trust
CA	Certification Authority
CA	Certification Authority Authorization
CDN	Content Distribution Network
CLI	Command Line Interface
CN	Common Name
CNAME	Canonical Name
CP	Certificate Policy
CPS	Certification Practice Statement
CRL	Certificate Revocation List
CT	Certificate Transparency
DANE	DNS-based Authenticated Named Entities
DNS	Domain Name System
DNSSEC	DNS Security Extensions
DS	Delegation Signer
DV	Domain Validation
EE	End Entity
EM	Emergency Management
ETSI	European Telecommunications Standards Institute
EV	Extended Validation
FQDN	Fully Qualified Domain Name
HTTP	Hyptertext Transport Protocol
IANA	Internet Assigned Numbers Authority
ICANN	Internet Corporation for Assigned Names and Numbers
ICN	Information-centric Network
IETF	Internet Engineering Taskforce
ISO	International Organization for Standardization
ITU	International Telecommunication Union
IV	Individual Validation
KSK	Key Signing Key
MTA	Mail Transfer Agent
MTA-STS	MTA Strict Transport Security
NDN	Named Data Network
NDO	Named Data Ojbect
NSEC	Next Secure
OCSP	Online Certificate Status Protocol

OV	Organization Validation
PKI	Public Key Infrastructure
PMTU	Path Maximum Transmission Unit
RFC	Request for Comment
RP	Relying Party
RR	Recursive Resolver
RR	Resource Record
RRSet	Resource Record Set
SCT	Signed Certificate Timestamp
SLD	Second Level Domain
SMTP	Simple Mail Transfer Protocol
SAN	Subject Alternative Name
SNI	Subject Name Indication
SoK	Systemization of Knowledge
SSL	Secure Sockets Layer
TA	Trust Attitude
TA	Trust Anchor
TLD	Top Level Domain
TLS	Transport Layer Security
TLSA	TLS Authentication
TOFU	Trust on First User
TTL	Time to Live
TTP	Trusted Third Party
URI	Uniform Resource Identifier
WOT	Web of Trust
ZSK	Zone Signing Key

Literatur

Peer-Reviewed References

[1] J. Aas, E. Rescorla, S. Schoen, B. Warren, R. Barnes, B. Case, Z. Durumeric, P. Eckersley, A. Flores-López, J. A. Halderman, J. Hoffman-Andrews und J. Kasten, “Let’s Encrypt: An Automated Certificate Authority to Encrypt the Entire Web,” in *Proc. of the 2019 ACM SIGSAC CCS*, New York, NY, USA: ACM Press, 2019, S. 2473–2487, ISBN: 9781450367479.

[2] M. Abadi, A. Birrell, I. Mironov, T. Wobber und Y. Xie, “Global Authentication in an Untrustworthy World,” in *Proceedings of the 14th USENIX Conference on Hot Topics in Operating Systems*, Ser. HotOS ’13, Santa Ana Pueblo, New Mexcio: USENIX Association, 2013.

[3] S. S. Adhatarao, J. Chen, M. Arumaithurai, X. Fu und K. K. Ramakrishnan, “Comparison of Naming Schema in ICN,” in *2016 IEEE International Symposium on Local and Metropolitan Area Networks*, Ser. LANMAN ’16, Piscataway, NJ, USA: IEEE, Juni 2016, S. 1–6.

[4] A. Afanasyev, X. Jiang, Y. Yu, J. Tan, Y. Xia, A. Mankin und L. Zhang, “NDNS: A DNS-Like Name Service for NDN,” in *2017 26th International Conference on Computer Communication and Networks*, Ser. ICCCN ’17, Piscataway, NJ, USA: IEEE, Juli 2017, S. 1–9, ISBN: 978-1-5090-2991-4.

[5] A. Afanasyev, C. Yi, L. Wang, B. Zhang und L. Zhang, “SNAMP: Secure namespace mapping to scale NDN forwarding,” in *2015 IEEE Conference on Computer Communications Workshops*, Ser. INCOCOM, Piscataway, NJ, USA: IEEE, Apr. 2015, S. 281–286.

[6] P. Agten, W. Joosen, F. Piessens und N. Nikiforakis, “Seven Months’ Worth of Mistakes: A Longitudinal Study of Typosquatting Abuse,” in *Proceedings of the 2015 NDSS*, Reston, VA, USA: Internet Society, 2015, S. 8–11, ISBN: 1-891562-38-X.

[7] D. Akhawe und A. P. Felt, “Alice in warningland: A large-scale field study of browser security warning effectiveness,” in *Proceedings of 22nd USENIX Security Symposium*, USENIX Association, 2013, S. 257–272, ISBN: 978-1-931971-03-4.

[8] E. S. Alashwali und K. Rasmussen, “What’s in a Downgrade? A Taxonomy of Downgrade Attacks in the TLS Protocol and Application Protocols Using TLS,” in *Security and Privacy in Communication Networks*, R. Beyah, B. Chang, Y. Li und S. Zhu, Hrsg., Cham: Springer International Publishing, 2018, S. 468–487, ISBN: 978-3-030-01704-0.

[9] M. Amadeo, C. Campolo und A. Molinaro, “Multi-Source Data Retrieval in IoT via Named Data Networking,” in *Proceedings of the 1st ACM Conference on Information-Centric Networking*, Ser. ICN ’14, Paris, France: ACM, 2014, S. 67–76.

[10] J. Amann, O. Gasser, Q. Scheitle, L. Brent, G. Carle und R. Holz, “Mission accomplished? HTTPS security after Diginotar,” in *Proceedings of the 2017 Internet Measurement Conference*, Ser. IMC ’17, London, United Kingdom: ACM, 2017, S. 325–340, ISBN: 9781450351188.

[11] A. Anand, M. Kallitsis, J. Sippe und A. Dainotti, “Aggressive Internet-Wide Scanners: Network Impact and Longitudinal Characterization,” in *Companion of the 19th International Conference on emerging Networking EXperiments and Technologies*, Ser. CoNEXT 2023, ACM, Dez. 2023, S. 1–8.

[12] M. Antonakakis, R. Perdisci, D. Dagon, W. Lee und N. Feamster, “Building a dynamic reputation system for DNS,” in *Proceedings of the 19th USENIX Security Symposium*, USENIX Association, 2010, S. 273–289, ISBN: 9781931971775.

[13] S. Arianfar, T. Koponen, B. Raghavan und S. Shenker, “On preserving privacy in content-oriented networks,” in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, Ser. ICN ’11, New York, NY, USA: ACM, 2011, S. 19–24, ISBN: 9781450308014.

[14] T. Asami, B. Namsraijav, Y. Kawahara, K. Sugiyama, A. Tagami, T. Yagyu, K. Nakamura und T. Hasegawa, “Moderator-Controlled Information Sharing by Identity-Based Aggregate Signatures for Information Centric Networking,” in *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, Ser. ICN ’15, San Francisco, California, USA: ACM, Sep. 2015, S. 157–166.

[15] O. Ascigil, S. Reñé, G. Xylomenos, I. Psaras und G. Pavlou, “A keyword-based ICN-IoT platform,” in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, Ser. ICN ’17, New York, NY, USA: ACM, Sep. 2017, S. 22–28.

[16] V. Bajpai, A. Brunstrom, A. Feldmann, W. Kellerer, A. Pras, H. Schulzrinne, G. Smaragdakis, M. Wählisch und K. Wehrle, “The Dagstuhl beginners guide to reproducibility for experimental networking research,” *SIGCOMM Comput. Commun. Rev.*, Jg. 49, Nr. 1, S. 24–30, Feb. 2019, ISSN: 0146-4833.

[17] V. Bajpai, M. Kühlewind, J. Ott, J. Schönwälder, A. Sperotto und B. Trammell, “Challenges with Reproducibility,” in *Proceedings of the Reproducibility Workshop*, Ser. Reproducibility ’17, Los Angeles, CA, USA: ACM, 2017, S. 1–4, ISBN: 9781450350600.

[18] G. Bansal, F. M. Zahedi und D. Gefen, “The role of privacy assurance mechanisms in building trust and the moderating role of privacy concern,” *European Journal of Information Systems*, Jg. 24, Nr. 6, S. 624–644, Nov. 2015.

[19] M. Bari, S. Chowdhury, R. Ahmed, R. Boutaba und B. Mathieu, “A Survey of Naming and Routing in Information-centric Networks,” *IEEE Communications Magazine*, Jg. 50, Nr. 12, S. 44–53, Dez. 2012.

[20] E. Barker, W. Barker, W. Burr, W. Polk und M. Smid, "NIST special publication 800-57," *NIST Special publication*, Jg. 800, Nr. 57, S. 1–142, 2007.

[21] S. Bechtold, "Governance in Namespaces," *Layola of Los Angeles Law Review*, Jg. 36, Nr. 3, 2003.

[22] S. M. Bellovin, "Using the domain name system for system break-ins," in *Proceedings of the 5th Conference on USENIX UNIX Security Symposium - Volume 5*, Ser. SSYM '95, Salt Lake City, Utah: USENIX Association, 1995, S. 18.

[23] N. Bhuiyan und A. Baghel, "An overview of continuous improvement: from the past to the present," *Management Decision*, Jg. 43, Nr. 5, S. 761–771, 2005.

[24] K. Bijlsma-Frankema und A. C. Costa, "Understanding the Trust-Control Nexus," *International Sociology*, Jg. 20, Nr. 3, S. 259–282, 2005.

[25] H. Birge-Lee, Y. Sun, A. Edmundson, J. Rexford und P. Mittal, "Bamboozling Certificate Authorities with BGP," in *27th USENIX Security Symposium (USENIX Security 18)*, Baltimore, MD: USENIX Association, Aug. 2018, S. 833–849, ISBN: 978-1-939133-04-5. URL: `https://www.usenix.org/conference/usenixsecurity18/presentation/birge-lee`.

[26] M. Blaze, J. Feigenbaum und J. Lacy, "Decentralized Trust Management," in *Proceedings of 1996 IEEE Symposium on Security and Privacy*, Piscataway, NJ, USA: IEEE, 1996, S. 164–173.

[27] N. Bos, J. Olson, D. Gergle, G. Olson und Z. Wright, "Effects of four computer-mediated communications channels on trust development," in *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves*, Ser. CHI '02, New York, NY, USA: ACM Press, 2002, S. 135, ISBN: 1581134533.

[28] J. L. Bradach und R. G. Eccles, "Price, Authority, and Trust: From Ideal Types to Plural Forms," *Annual Review of Sociology*, Jg. 15, Nr. 1, S. 97–118, 1989.

[29] D. D. S. Braga, M. Niemann, B. Hellingrath und F. B. D. L. Neto, "Survey on Computational Trust and Reputation Models," *ACM Computing Surveys*, Jg. 51, Nr. 5, S. 1–40, Nov. 2018, ISSN: 0360-0300.

[30] M. Brandt, T. Dai, A. Klein, H. Shulman und M. Waidner, "Domain Validation++ For MitM-Resilient PKI," in *Proceedings of the 2018 ACM SIGSAC*, New York, NY, USA: ACM Press, Okt. 2018, S. 2060–2076, ISBN: 9781450356930.

[31] M. Brinkmann, C. Dresen, R. Merget, D. Poddebniak, J. Müller, J. Somorovsky, J. Schwenk und S. Schinzel, "ALPACA: Application Layer Protocol Confusion - Analyzing and Mitigating Cracks in TLS Authentication," in *30th USENIX Security Symposium (USENIX Security 21)*, Berkeley, CA, USA: USENIX Association, Aug. 2021, S. 4293–4310.

[32] M. Büscher, P. Holst Mogensen und M. Kristensen, "When and How (Not) to Trust It? Supporting Virtual Emergency Teamwork," *International Journal of Information Systems for Crisis Response and Management*, Jg. 1, Nr. 2, S. 1–15, Apr. 2009, ISSN: 1937-9390.

[33] F. Cangialosi, T. Chung, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove und C. Wilson, "Measurement and Analysis of Private Key Sharing in the HTTPS Ecosystem," in *Proceedings of the 2016 ACM SIGSAC*, New York, NY, USA: ACM, Okt. 2016.

[34] F. Cangialosi und A. Narayan, "Reproducible Experiments for Internet Systems," in *Proceedings of the 5th International Workshop on Practical Reproducible Evaluation of Computer Systems*, Ser. P-RECS '22, Minneapolis, MN, USA: ACM, 2022, S. 11–14, ISBN: 9781450393133.

[35] X. de Carné de Carnavalet und P. C. van Oorschot, "A Survey and Analysis of TLS Interception Mechanisms and Motivations: Exploring how end-to-end TLS is made "end-to-me" for web traffic," *ACM Comput. Surv.*, Jg. 55, Nr. 13s, Juli 2023, ISSN: 0360-0300.

[36] M. Castells, "A Network Theory of Power," *International Journal of Communication*, Jg. 5, S. 773–787, 2011, ISSN: 1932-8036.

[37] D. W. Chadwick und A. Basden, "Evaluating trust in a public key certification authority," *Computers and Security*, Jg. 20, Nr. 7, S. 592–611, 2001, ISSN: 0167-4048.

[38] R. Chatterjee, S. Ruj und S. DasBit, "Public Key Infrastructure for Named Data Networks," in *Proceedings of the 21st International Conference on Distributed Computing and Networking*, Ser. ICDCN '20, Kolkata, India: ACM, Jan. 2020, S. 1–10.

[39] J. Chen, M. Arumaithurai, X. Fu und K. K. Ramakrishnan, "CNS: Content-oriented Notification Service for Managing Disasters," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, Ser. ICN '16, New York, NY, USA: ACM, Sep. 2016, S. 122–131.

[40] Q. A. Chen, M. Thomas, E. Osterweil, Y. Cao, J. You und Z. M. Mao, "Client-side Name Collision Vulnerability in the New gTLD Era: A Systematic Study," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, S. 941–956.

[41] J.-H. Cho, K. Chan und S. Adali, "A Survey on Trust Modeling," *ACM Computing Surveys*, Jg. 48, Nr. 2, S. 1–40, Okt. 2015, ISSN: 0360-0300.

[42] T. Chung, Y. Liu, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove und C. Wilson, "Measuring and Applying Invalid SSL Certificates: The Silent Majority," in *Proceedings of the 2016 Internet Measurement Conference*, Ser. IMC '16, Santa Monica, California, USA: ACM, 2016, S. 527–541, ISBN: 9781450345262.

[43] T. Chung, J. Lok, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, J. Rula, N. Sullivan und C. Wilson, "Is the Web Ready for OCSP Must-Staple?" In *Proceedings of the Internet Measurement Conference 2018*, Ser. IMC '18, Boston, MA, USA: ACM, 2018, S. 105–118, ISBN: 9781450356190.

[44] T. Chung, R. van Rijswijk-Deij, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove und C. Wilson, "Understanding the Role of Registrars in DNSSEC Deployment," in *Proceedings of the ACM IMC '17*, New York, NY, USA: ACM, 2017, S. 369–383, ISBN: 9781450351188.

[45] T. Chung, R. Van Rijswijk-Deij, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove und C. Wilson, “A longitudinal, end-to-end view of the DNSSEC ecosystem,” in *Proceedings of the 26th USENIX Security Symposium*, Berkeley, CA, USA: USENIX Association, 2017, S. 1307–1322, ISBN: 9781931971409.

[46] D. E. Comer und L. L. Peterson, “Understanding naming in distributed systems,” *Distributed Computing*, Jg. 3, Nr. 2, S. 51–60, Juni 1989.

[47] A. Compagno, M. Conti und R. Droms, “OnboardICNg: a Secure Protocol for On-boarding IoT Devices in ICN,” in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, Ser. ICN ’16, New York, NY, USA: ACM, Sep. 2016, S. 166–175.

[48] C. L. Corritore, B. Kracher und S. Wiedenbeck, “On-line trust: concepts, evolving themes, a model,” *International Journal of Human-Computer Studies*, Jg. 58, Nr. 6, S. 737–758, Juni 2003.

[49] E. Costante, J. Den Hartog und M. Petkovic, “On-line trust perception: What really matters,” *Proceedings of the 1st Workshop on Socio-Technical Aspects in Security and Trust*, STAST ’11, S. 52–59, 2011.

[50] C. Dannewitz, J. Golic, B. Ohlman und B. Ahlgren, “Secure Naming for a Network of Information,” in *2010 INFOCOM IEEE Conference on Computer Communications Workshops*, IEEE, März 2010, S. 1–6, ISBN: 978-1-4244-6739-6.

[51] C. Dannewitz, D. Kutscher, B. Ohlman, S. Farrell, B. Ahlgren und H. Karl, “Network of Information (NetInf) – An information-centric networking architecture,” *Computer Communications*, Jg. 36, Nr. 7, S. 721–735, Apr. 2013, ISSN: 01403664.

[52] O. Darwich, H. Rimlinger, M. Dreyfus, M. Gouel und K. Vermeulen, “Replication: Towards a Publicly Available Internet Scale IP Geolocation Dataset,” in *Proceedings of the 2023 ACM on Internet Measurement Conference*, Ser. IMC ’23, ACM, Okt. 2023.

[53] A. Delignat-Lavaud, M. Abadi, A. Birrell, I. Mironov, T. Wobber und Y. Xie, “Web PKI: Closing the Gap between Guidelines and Practices,” in *Proceedings of the 2014 NDSS*, Reston, VA, USA: Internet Society, 2014, S. 23–26, ISBN: 1-891562-35-5.

[54] A. Delignat-Lavaud und K. Bhargavan, “Network-based Origin Confusion Attacks against HTTPS Virtual Hosting,” in *Proceedings of the 24th International Conference on World Wide Web*, Ser. WWW ’15, Florence, Italy: International World Wide Web Conferences Steering Committee, 2015, S. 227–237, ISBN: 9781450334693.

[55] N. Demir, M. Große-Kampmann, T. Urban, C. Wressnegger, T. Holz und N. Pohlmann, “Reproducibility and Replicability of Web Measurement Studies,” in *Proceedings of the ACM Web Conference 2022*, Ser. WWW ’22, Virtual Event, Lyon, France: ACM, 2022, S. 533–544, ISBN: 9781450390965.

[56] Der deutsche Bundestag, “Gesetz zur Verbesserung der Rechtsdurchsetzung in sozialen Netzwerken (Netzwerkdurchsetzungsgesetz – NetzDG),” *Bundesgesetzblatt*, Jg. Teil I, Nr. 61, S. 3352–3355, Sep. 2017.

[57] A. Detti, N. Blefari Melazzi, S. Salsano und M. Pomposini, “CONET: A Content Centric Inter-Networking Architecture,” in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, Ser. ICN ’11, New York, NY, USA: ACM, 2011, S. 50–55, ISBN: 9781450308014.

[58] S. DiBenedetto und C. Papadopoulos, “Mitigating poisoned content with forwarding strategy,” in *2016 IEEE Conference on Computer Communications Workshops*, Piscataway, NJ, USA: IEEE, Apr. 2016, S. 164–169.

[59] G. Dietz, “Going Back to the Source: Why Do People Trust Each Other?” *Journal of Trust Research*, Jg. 1, Nr. 2, S. 215–222, 2011.

[60] P. T. Dinesen, “Does Generalized (Dis)Trust Travel? Examining the Impact of Cultural Heritage and Destination-Country Environment on Trust of Immigrants,” *Political Psychology*, Jg. 33, Nr. 4, S. 495–511, Mai 2012.

[61] Dirk Trossen und Alexandros Kostopoulos, “Techno-Economic Aspects of Information-Centric Networking,” *Journal of Information Policy*, Jg. 2, S. 26–50, 2012, ISSN: 21583897.

[62] M. Dodgson, “Learning, Trust, and Technological Collaboration:” *Human Relations*, Jg. 46, Nr. 1, S. 77–95, 2016.

[63] Z. Dong, K. Kane und L. J. Camp, “Detection of Rogue Certificates from Trusted Certificate Authorities Using Deep Neural Networks,” *ACM Transactions on Privacy and Security (TOPS)*, Jg. 19, Nr. 2, Sep. 2016, ISSN: 2471-2566.

[64] A. Drichel, V. Drury, J. von Brandt und U. Meyer, “Finding Phish in a Haystack: A Pipeline for Phishing Classification on Certificate Transparency Logs,” in *Proceedings of the ARES*, Ser. ARES ’21, New York, NY, USA: ACM, Aug. 2021.

[65] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey und J. A. Halderman, “A Search Engine Backed by Internet-Wide Scanning,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, Ser. CCS ’15, ACM, Okt. 2015.

[66] Z. Durumeric, D. Adrian, P. Stephens, E. Wustrow und J. A. Halderman, “Ten Years of ZMap,” in *Proceedings of the 2024 ACM on Internet Measurement Conference*, Ser. IMC ’24, Madrid, Spain: ACM, 2024, S. 139–148, ISBN: 9798400705922.

[67] Z. Durumeric, J. Kasten, M. Bailey und J. A. Halderman, “Analysis of the HTTPS certificate ecosystem,” in *Proceedings of the 2013 conference on Internet measurement conference*, Ser. IMC ’13, New York, NY, USA: ACM, 2013, S. 291–304.

[68] Z. Durumeric, M. Payer, V. Paxson, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann und J. Beekman, "The Matter of Heartbleed," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, Ser. IMC '14, New York, NY, USA: ACM, 2014, S. 475–488.

[69] Z. Durumeric, E. Wustrow und J. A. Halderman, "ZMap: Fast Internet-Wide Scanning and Its Security Applications," in *Proceedings of the 22nd USENIX Conference on Security*, Ser. SEC '13, Washington, D.C.: USENIX Association, 2013, S. 605–620.

[70] T. B. Eikeland und T. Saevi, "Beyond Rational Order: Shifting the Meaning of Trust in Organizational Research," *Human Studies*, Jg. 40, Nr. 4, S. 603–636, 2017.

[71] T. Fadai, S. Schrittwieser, P. Kieseberg und M. Mulazzani, "Trust Me, I'm a Root CA! Analyzing SSL Root CAs in Modern Browsers and Operating Systems," in *Proceedings of the 2015 10th ARES*, Piscataway, NJ, USA: IEEE, 2015, S. 174–179, ISBN: 9781467365901.

[72] D. Faems, M. Janssens, A. Madhok und B. V. Looy, "Toward an Integrative Perspective on Alliance Governance: Connecting Contract Design, Trust Dynamics, and Contract Application," *The Academy of Management Journal*, Jg. 51, Nr. 6, S. 1053–1078, 2008.

[73] S. M. Farhan und T. Chung, "Exploring the evolution of TLS certificates," in *Passive and Active Measurement*, A. Brunstrom, M. Flores und M. Fiore, Hrsg., Cham, Switzerland: Springer Nature Switzerland, 2023, S. 71–84, ISBN: 978-3-031-28486-1.

[74] E. Fasllija, H. F. Enişer und B. Prünster, "Phish-Hook: Detecting Phishing Certificates Using Certificate Transparency Logs," in *Security and Privacy in Communication Networks*, Cham, Switzerland: Springer International Publishing, Dez. 2019, S. 320–334.

[75] A. P. Felt, R. W. Reeder, A. Ainslie, H. Harris, M. Walker, C. Thompson, M. E. Acer, E. Morant und S. Consolvo, "Rethinking connection security indicators," in *Proceedings of 15th SOUPS*, USENIX Association, 2019, S. 1–14, ISBN: 9781931971317.

[76] A. Filastò und J. Appelbaum, "OONI: Open Observatory of Network Interference," in *2nd USENIX Workshop on Free and Open Communications on the Internet (FOCI 12)*, Bellevue, WA: USENIX Association, Aug. 2012.

[77] M. Flittner, M. N. Mahfoudi, D. Saucez, M. Wählisch, L. Iannone, V. Bajpai und A. Afanasyev, "A Survey on Artifacts from CoNEXT, ICN, IMC, and SIGCOMM Conferences in 2017," *ACM SIGCOMM Computer Communication Review*, Jg. 48, Nr. 1, S. 75–80, Apr. 2018, ISSN: 0146-4833.

[78] B. J. Fogg, P. Swani, M. Treinen, J. Marshall, O. Laraki, A. Osipovich, C. Varma, N. Fang, J. Paul, A. Rangnekar und J. Shon, "What makes Web sites credible?" In *Proceedings of the SIGCHI conference on Human factors in computing systems*, Ser. CHI '01, ACM, 2001.

[79] P. Fotouhi Tehrani, R. Hiesgen, T. Lübeck, T. C. Schmidt und M. Wählisch, “Do CAA, CT, and DANE Interlink in Certificate Deployments? A Web PKI Measurement Study,” in *Proc. of Network Traffic Measurement and Analysis Conference (TMA)*, Dresden, Germany: IEEE, Mai 2024, S. 1–11.

[80] P. Fotouhi Tehrani, N. von Kalckreuth und S. Lamprecht, “Toward an Integrative Model of Trust for Digital Emergency Communication,” in *ISCRAM 2020 Conference Proceedings–17th International Conference on Information Systems for Crisis Response and Management*, Ser. ISCRAM ’20, Mai 2020, S. 1012–1021.

[81] P. Fotouhi Tehrani, L. Keidel, E. Osterweil, J. H. Schiller, T. C. Schmidt und M. Wählisch, “NDNSSEC: Namespace Management in NDN with DNSSEC,” in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, Ser. ICN ’19, Macau, China: ACM, Sep. 2019.

[82] P. Fotouhi Tehrani, E. Osterweil, J. H. Schiller, T. C. Schmidt und M. Wahlisch, “Authenticated Communication in Crises: Toward an Infrastructureless Trust Model for Challenged Networks,” in *2019 International Conference on Information and Communication Technologies for Disaster Management*, Ser. ICT-DM ’19, Paris, France: IEEE, Dez. 2019.

[83] P. Fotouhi Tehrani, E. Osterweil, J. H. Schiller, T. C. Schmidt und M. Wählisch, “The Missing Piece: On Namespace Management in NDN and How DNSSEC Might Help,” in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, Ser. ICN ’19, Macau, China: ACM, Sep. 2019.

[84] P. Fotouhi Tehrani, E. Osterweil, J. H. Schiller, T. C. Schmidt und M. Wählisch, “Security of Alerting Authorities in the WWW: Measuring Namespaces, DNSSEC, and Web PKI,” in *Proceedings of the Web Conference 2021*, Ser. WWW ’21, Virtual Event, Ljubljana, Slovenia: ACM, Apr. 2021.

[85] P. Fotouhi Tehrani, E. Osterweil, T. C. Schmidt und M. Wählisch, “SoK: Public key and namespace management in NDN,” in *Proceedings of the 9th ACM Conference on Information-Centric Networking*, Ser. ICN ’22, New York, NY, USA: ACM, Sep. 2022.

[86] P. Fotouhi Tehrani, E. Osterweil, T. C. Schmidt und M. Wählisch, “A Security Model for Web-Based Communication,” *Communication of the ACM*, 2024.

[87] P. Fotouhi Tehrani, J. H. Schiller, T. C. Schmidt und M. Wählisch, “On economic, societal, and political aspects in ICN,” in *Proceedings of the 7th ACM Conference on Information-Centric Networking*, Ser. ICN ’20, ACM, Sep. 2020.

[88] Z. Gao und A. Venkataramani, “Measuring Update Performance and Consistency Anomalies in Managed DNS Services,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Piscataway, NJ, USA, 2019, S. 2206–2214.

[89] O. Gasser, B. Hof, M. Helm, M. Korczynski, R. Holz und G. Carle, "In Log We Trust: Revealing Poor Security Practices with Certificate Transparency Logs and Internet Measurements," in *Passive and Active Measurement*, Cham, Switzerland: Springer International Publishing, März 2018, S. 173–185.

[90] P. Gasti, G. Tsudik, E. Uzun und L. Zhang, "DoS and DDoS in Named Data Networking," in *2013 22nd International Conference on Computer Communication and Networks*, Ser. ICCCN '13, Piscataway, NJ, USA: IEEE, Juli 2013, S. 1–7, ISBN: 978-1-4673-5775-3.

[91] A. Gebhard und D. Perouli, "Poster: Measuring Adoption of SPF, DMARC, and CAA DNS Records with Nonfederal Governments in the United States," in *Proceedings of the IEEE ICNP*, Piscataway, NJ, USA, 2023.

[92] C. Ghali, G. Tsudik und E. Uzun, "Network-Layer Trust in Named-Data Networking," *ACM SIGCOMM Computer Communication Review*, Jg. 44, Nr. 5, S. 12–19, 13. Feb. 2014, ISSN: 0146-4833.

[93] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti und S. Shenker, "Naming in content-oriented architectures," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, Ser. ICN '11, New York, NY, USA: ACM, 2011, S. 1–6, ISBN: 9781450308014.

[94] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan und J. Wilcox, "Information-centric networking," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, Ser. HotNets '11, New York, NY, USA: ACM, 2011, S. 1–6, ISBN: 9781450310598.

[95] A. W. Gouldner, "The Norm of Reciprocity: A Preliminary Statement," *American Sociological Review*, Jg. 25, Nr. 2, 1960.

[96] M. S. Granovetter, "The Strength of Weak Ties," *American Journal of Sociology*, Jg. 78, Nr. 6, S. 1360–1380, Mai 1973, ISSN: 0002-9602.

[97] G. Guette, "Automating trusted key rollover in DNSSEC," *Journal of Computer Security*, Jg. 17, Nr. 6, S. 839–854, Dez. 2009, ISSN: 0926-227X.

[98] G. Guette, B. Cousin und D. Fort, "Algorithm for DNSSEC Trusted Key Rollover," in *Information Networking. Convergence in Broadband and Mobile Networking*, C. Kim, Hrsg., Berlin, Heidelberg, Germany: Springer Berlin Heidelberg, 2005, S. 679–688, ISBN: 978-3-540-30582-8.

[99] C. Gündoğan, C. Amsüss, T. C. Schmidt und M. Wählisch, "Reliable Firmware Updates for the Information-Centric Internet of Things," in *Proceedings of the 8th ACM Conference on Information-Centric Networking*, Ser. ICN '21, New York, NY, USA: ACM, Sep. 2021, S. 59–70.

[100] C. Gündoğan, J. Pfender, M. Frey, T. C. Schmidt, F. Shzu-Juraschek und M. Wählisch, "Gain More for Less: The Surprising Benefits of QoS Management in Constrained NDN Networks," in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, Ser. ICN '19, New York, NY, USA: ACM, Sep. 2019, S. 141–152.

[101] P. Gusev und J. Burke, “NDN-RTC: Real-Time Videoconferencing over Named Data Networking,” in *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, Ser. ICN ’15, San Francisco, CA, USA: ACM, Sep. 2015, S. 117–126.

[102] J. Gustafsson, G. Overier, M. Arlitt und N. Carlsson, “A first look at the CT landscape: Certificate transparency logs in practice,” in *Passive and Active Measurement*, Cham, Switzerland, 2017, S. 87–99, ISBN: 978-3-319-54328-4.

[103] A. van Haasteren, F. Gille, M. Fadda und E. Vayena, “Development of the mHealth App Trustworthiness checklist,” *DIGITAL HEALTH*, Jg. 5, S. 1–21, Jan. 2019.

[104] G. R. Hagen, “Sovereign domains and property claims,” *International Journal of Law and Information Technology*, Jg. 11, Nr. 1, S. 1–39, 2003, ISSN: 14643693.

[105] B. Hamdane, R. Boussada, M. E. Elhdhili und S. G. E. Fatmi, “Hierarchical Identity Based Cryptography for Security and Trust in Named Data Networking,” in *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*, Ser. WETICE ’17, Piscataway, NJ, USA: IEEE, Juni 2017, S. 226–231.

[106] Hao Yang, E. Osterweil, D. Massey, Songwu Lu und Lixia Zhang, “Deploying Cryptography in Internet-Scale Systems: A Case Study on DNSSEC,” *IEEE Transactions on Dependable and Secure Computing*, Jg. 8, Nr. 5, S. 656–669, Sep. 2011, ISSN: 1545-5971.

[107] S. N. Hebrok, S. Nachtigall, M. Maehren, N. Erinola, R. Merget, J. Somorovsky und J. Schwenk, “We Really Need to Talk About Session Tickets: A Large-Scale Analysis of Cryptographic Dangers with TLS Session Tickets,” in *32st USENIX Security Symposium*, Berkeley, CA, USA: USENIX, 2023.

[108] M. P. Heinl, A. Giehl, N. Wiedermann, S. Plaga und F. Kargl, “MERCAT: A Metric for the Evaluation and Reconsideration of Certificate Authority Trustworthiness,” in *Proceedings of the 2019 ACM SIGSAC CCSW*, New York, NY, USA: ACM, 2019, S. 1–15, ISBN: 9781450368261.

[109] E. Hemmati und J. Garcia-Luna-Aceves, “A New Approach to Name-Based Link-State Routing for Information-Centric Networks,” in *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, Ser. ICN ’15, San Francisco, CA, USA: ACM, Sep. 2015, S. 29–38.

[110] R. Holz, J. Amann, O. Mehani, M. Wachs und M. Ali Kaafar, “TLS in the Wild: An Internet-wide Analysis of TLS-based Protocols for Electronic Communication,” in *Proceedings 2016 Network and Distributed System Security Symposium*, Reston, VA, USA: Internet Society, 2016, ISBN: 1-891562-41-X.

[111] R. Holz, L. Braun, N. Kammenhuber und G. Carle, “The SSL landscape – A Thorough Analysis of the X.509 PKI Using Active and Passive Measurement,” in *Proceedings of the ACM IMC ’11*, New York, NY, USA: ACM, 2011, S. 427–444, ISBN: 9781450310130.

[112] A. K. M. M. Hoque, S. O. Amin, A. Alyyan, B. Zhang, L. Zhang und L. Wang, “NLSR: Named-Data Link State Routing Protocol,” in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking*, Ser. ICN ’13, Hong Kong, China: ACM, 2013, S. 15–20, ISBN: 9781450321792.

[113] S. Hussain, S. S. Ullah, A. Gumaei, M. Al-Rakhami, I. Ahmad und S. M. Arif, “A Novel Efficient Certificateless Signature Scheme for the Prevention of Content Poisoning Attack in Named Data Networking-Based Internet of Things,” *IEEE Access*, Jg. 9, S. 40 198–40 215, 2021.

[114] L. Izhikevich, R. Teixeira und Z. Durumeric, “LZR: Identifying Unexpected Internet Services,” in *30th USENIX Security Symposium (USENIX Security 21)*, Berkeley, CA, USA: USENIX Association, Aug. 2021, S. 3111–3128.

[115] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs und R. L. Braynard, “Networking named content,” in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, Ser. CoNEXT ’09, New York, NY, USA: ACM, Jan. 2009, S. 1–12, ISBN: 9781605586366.

[116] Y. Jin, M. Tomoishi und S. Matsuura, “A Detection Method Against DNS Cache Poisoning Attacks Using Machine Learning Techniques: Work in Progress,” in *2019 IEEE 18th International Symposium on Network Computing and Applications*, Ser. NCA ’19, Piscataway, NJ, USA, 2019, S. 1–3.

[117] W. John, S. Tafvelin und T. Olovsson, “Passive internet measurement: Overview and guidelines based on experiences,” *Computer Communications*, Jg. 33, Nr. 5, S. 533–550, März 2010, ISSN: 0140-3664.

[118] K. V. Katsaros, L. Saino, I. Psaras und G. Pavlou, “On information exposure through named content,” in *10th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, Piscataway, NJ, USA: IEEE, Aug. 2014, S. 152–157, ISBN: 9781631900259.

[119] M. T. Khan, X. Huo, Z. Li und C. Kanich, “Every second counts: Quantifying the negative externalities of cybercrime via typosquatting,” in *2015 IEEE Symposium on Security and Privacy*, IEEE, Piscataway, NJ, USA, 2015, S. 135–150.

[120] S. Khan, F. Luo, Z. Zhang, F. Ullah, F. Amin, S. F. Qadri, M. B. B. Heyat, R. Ruby, L. Wang, S. Ullah, M. Li, V. C. M. Leung und K. Wu, “A Survey on X.509 Public-Key Infrastructure, Certificate Revocation, and Their Modern Implementation on Blockchain and Ledger Technologies,” *Commun. Surveys Tuts.*, Jg. 25, Nr. 4, S. 2529–2568, Okt. 2023, ISSN: 1553-877X.

[121] R. Khare, “What’s in a Name? Trust. Internet-Scale Namespaces, Part II,” *IEEE Internet Computing*, Jg. 3, Nr. 6, S. 80–84, 1999.

[122] S. Khoussi, D. Pesavento, L. Benmohamed und A. Battou, “NDN-Trace: A Path Tracing Utility for Named Data Networking,” in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, Ser. ICN ’17, New York, NY, USA: ACM, Sep. 2017, S. 116–122.

[123] D. Kim, H. Cho, Y. Kwon, A. Doupé, S. Son, G.-J. Ahn und T. Dumitras, “Security Analysis on Practices of Certificate Authorities in the HTTPS Phishing Ecosystem,” in *Proceedings of the 2021 ACM ASIA-CCS*, ACM, Mai 2021.

[124] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, N. Nikiforakis und M. Antonakakis, “Hiding in Plain Sight: A Longitudinal Study of Combosquatting Abuse,” in *Proceedings of the 2017 ACM SIGSAC CCS*, New York, NY, USA: ACM, 2017, S. 569–586, ISBN: 9781450349468.

[125] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker und I. Stoica, “A data-oriented (and beyond) network architecture,” *ACM SIGCOMM Computer Communication Review*, Jg. 37, Nr. 4, S. 181, Okt. 2007, ISSN: 01464833.

[126] M. Korczynski, M. Wullink, S. Tajalizadehkhoob, G. C. M. Moura, A. Noroozian, D. Bagley und C. Hesselman, “Cybercrime After the Sunrise: A Statistical Analysis of DNS Abuse in New gTLDs,” in *Proceedings of the 2018 ACM ASIACCS*, New York, NY, USA: ACM, 2018, S. 609–623, ISBN: 9781450355766.

[127] N. Korzhitskii und N. Carlsson, “Characterizing the root landscape of certificate transparency logs,” in *2020 IFIP Networking Conference*, IEEE, Piscataway, NJ, USA: IEEE, 2020, S. 190–198.

[128] P. Kotzias, A. Razaghpanah, J. Amann, K. G. Paterson, N. Vallina-Rodriguez und J. Caballero, “Coming of Age: A Longitudinal Study of TLS Deployment,” in *Proceedings of the Internet Measurement Conference 2018*, Ser. IMC ’18, Boston, MA, USA: ACM, 2018, S. 415–428, ISBN: 978-978-1-4503-5619-0.

[129] S. Kramer, R. Goré und E. Okamoto, “Formal definitions and complexity results for trust relations and trust domains fit for TTPs, the web of trust, PKIs, and ID-based cryptography,” *ACM SIGACT News*, Jg. 41, Nr. 1, S. 75–98, März 2010.

[130] M. Król, K. Habak, D. Oran, D. Kutscher und I. Psaras, “RICE: Remote Method Invocation in ICN,” in *Proceedings of the 5th ACM Conference on Information-Centric Networking*, Ser. ICN ’18, New York, NY, USA: ACM, Sep. 2018, S. 1–11.

[131] M. Król, S. Mastorakis, D. Oran und D. Kutscher, “Compute First Networking: Distributed Computing meets ICN,” in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, Ser. ICN ’19, New York, NY, USA: ACM, Sep. 2019, S. 67–77.

[132] M. Król und I. Psaras, “NFaaS: Named Function as a Service,” in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, Ser. ICN ’17, New York, NY, USA: ACM, Sep. 2017, S. 134–144.

[133] K. Krombholz, W. Mayer, M. Schmiedecker und E. Weippl, “‘I have no idea what I’m doing’ – on the usability of deploying HTTPS,” in *Proceedings of the 26th USENIX Security Symposium*, Berkley, CA, USA: USENIX Association, 2017, S. 1339–1356, ISBN: 9781931971409.

[134] J. Kurihara, K. Yokota und A. Tagami, “A Consumer-Driven Access Control Approach to Censorship Circumvention in Content-Centric Networking,” in *Proceedings of the 2016 conference on 3rd ACM Conference on Information-Centric Networking*, Ser. ICN ’16, New York, NY, USA: ACM, 2016, S. 186–194.

[135] D. Kutscher, L. A. Wardani und T. M. R. Gias, “Vision: information-centric dataflow,” in *Proceedings of the 8th ACM Conference on Information-Centric Networking*, Ser. ICN ’21, New York, NY, USA: ACM, 2021, S. 52–58.

[136] T. Lauinger, N. Laoutaris, P. Rodriguez, T. Strufe, E. Biersack und E. Kirda, “Privacy risks in named data networking: what is the cost of performance?” *SIGCOMM Comput. Commun. Rev.*, Jg. 42, Nr. 5, S. 54–57, Sep. 2012, ISSN: 0146-4833.

[137] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński und W. Joosen, “Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation,” in *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, Ser. NDSS ’19, The Internet Society, Feb. 2019.

[138] M. Leaffer, “Domain Names, Globalization, and Internet Governance,” *Indiana Journal of Global Legal Studies*, Jg. 6, Nr. 1, S. 139–165, 1998.

[139] H. Lee, I. Ashiq, M. Müller, R. van Rijswijk-Deji, T. Kwon und T. Chung, “Under the Hood of DANE Mismanagement in SMTP,” in *USENIX Security 22*, Boston, MA: USENIX Association, Aug. 2022.

[140] H. Lee, A. Gireesh, R. van Rijswijk-Deij, T. Kwon und T. Chung, “A Longitudinal and Comprehensive Study of the DANE Ecosystem in Email,” in *Proc. of the USENIX Security Symposium*, Berkeley, CA, USA: USENIX Association, Aug. 2020, S. 613–630. URL: `https://www.usenix.org/conference/usenixsecurity20/presentation/lee-hyeonmin`.

[141] H. Lee, D. Kim und Y. Kwon, “TLS 1.3 in Practice: How TLS 1.3 Contributes to the Internet,” in *Proceedings of the Web Conference 2021*, Ser. WWW ’21, Virtual Event, Ljubljana, Slovenia: ACM, 2021, S. 70–79, ISBN: 9781450383127.

[142] J. Lee, H. Lee, J. Jeong, D. Kim und T. T. Kwon, “Analyzing Spatial Differences in the TLS Security of Delegated Web Services,” in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, New York, NY, USA: ACM, Mai 2021, S. 475–487.

[143] K. C. Lee und N. Chung, “Understanding factors affecting trust in and satisfaction with mobile banking in Korea: A modified DeLone and McLean’s model perspective,” *Interacting with Computers*, Jg. 21, Nr. 5-6, S. 385–392, Dez. 2009.

[144] J. D. Lewis und A. Weigert, “Trust as a Social Reality,” *Social Forces*, Jg. 63, Nr. 4, S. 967–985, 1985, ISSN: 0037-7732.

[145] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot und H. Pfister, “UpSet: Visualization of Intersecting Sets,” *IEEE Transactions on Visualization and Computer Graphics*, Jg. 20, Nr. 12, S. 1983–1992, 2014.

[146] W. Lian, E. Rescorla, H. Shacham und S. Savage, “Measuring the practical impact of DNSSEC deployment,” in *Proceedings of the 22nd USENIX Security Symposium*, Berkley, CA, USA: USENIX Association, 2013, S. 573–587, ISBN: 9781931971034.

[147] J. Liang, J. Jiang, H. Duan, K. Li, T. Wan und J. Wu, “When HTTPS Meets CDN: A Case of Authentication in Delegated Service,” in *2014 IEEE Symposium on Security and Privacy*, Piscataway, NJ, USA: IEEE, Mai 2014.

[148] T. Liang, Z. Xia, G. Tang, Y. Zhang und B. Zhang, “NDN in Large LEO Satellite Constellations: A Case of Consumer Mobility Support,” in *Proceedings of the 8th ACM Conference on Information-Centric Networking*, Ser. ICN ’21, Paris, France: ACM, 2021, S. 1–12.

[149] Y. Liu, W. Tome, L. Zhang, D. Choffnes, D. Levin, B. Maggs, A. Mislove, A. Schulman und C. Wilson, “An End-to-End Measurement of Certificate Revocation in the Web’s PKI,” in *Proceedings of the 2015 Internet Measurement Conference*, Ser. IMC ’15, New York, NY, USA: ACM Press, 2015, S. 183–196, ISBN: 9781450338486.

[150] P. H. Longstaff und S.-U. Yang, “Communication Management and Trust: Their Role in Building Resilience to SSurprisesSSuch As Natural Disasters, Pandemic Flu, and Terrorism,” *Ecology and Society*, Jg. 13, Nr. 1, S. 1–14, 2008.

[151] Z. Luo, Z. He, J. Wang, M. Dong, J. Huang, M. Chen und B. Zheng, “AutoSmart: An Efficient and Automatic Machine Learning Framework for Temporal Relational Data,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Ser. KDD ’21, Virtual Event, Singapore: ACM, 2021, S. 3976–3984, ISBN: 9781450383325.

[152] Z. Ma, J. Austgen, J. Mason, Z. Durumeric und M. Bailey, “Tracing your roots: exploring the TLS trust anchor ecosystem,” in *Proceedings of the 21st ACM Internet Measurement Conference*, New York, NY, USA: ACM, Nov. 2021, S. 179–194.

[153] Z. Ma, A. Faulkenberry, T. Papastergiou, Z. Durumeric, M. D. Bailey, A. D. Keromytis, F. Monrose und M. Antonakakis, “Stale TLS Certificates: Investigating Precarious Third-Party Access to Valid TLS Keys,” in *Proceedings of the 2023 ACM on Internet Measurement Conference*, Ser. IMC ’23, Montreal QC, Canada: ACM, 2023, S. 222–235, ISBN: 9798400703829.

[154] Z. Ma, J. Mason, M. Antonakakis, Z. Durumeric und M. Bailey, “What’s in a Name? Exploring CA Certificate Control,” in *30th USENIX Security Symposium (USENIX Security 21)*, Berkeley, CA, USA: USENIX Association, Aug. 2021, S. 4383–4400.

[155] P. Mahadevan, E. Uzun, S. Sevilla und J. Garcia-Luna-Aceves, “CCN-KRS: A Key Resolution Service for CCN,” in *Proceedings of the 1st International Conference on Information-centric Networking*, Ser. ICN ’14, New York, New York, USA: ACM, 2014, S. 97–106.

[156] K. Man, Z. Qian, Z. Wang, X. Zheng, Y. Huang und H. Duan, "DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, Ser. CCS '20, Virtual Event, USA: ACM, 2020, S. 1337–1350, ISBN: 9781450370899. URL: `https://doi.org/10.1145/3372297.3417280`.

[157] K. Man, X. Zhou und Z. Qian, "DNS Cache Poisoning Attack: Resurrections with Side Channels," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, Ser. CCS '21, Virtual Event, Republic of Korea: ACM, 2021, S. 3400–3414, ISBN: 9781450384544.

[158] K. Manandhar, B. Adcock und X. Cao, "Preserving the Anonymity in MobilityFirst networks," in *2014 23rd International Conference on Computer Communication and Networks*, Ser. ICCCN '14, Piscataway, NJ, USA: IEEE, Aug. 2014, S. 1–6, ISBN: 9781479935727.

[159] E. Mannes und C. Maziero, "Naming content on the network layer: A security analysis of the information-centric network model," *ACM Computing Surveys*, Jg. 52, Nr. 3, S. 1–28, 2019, ISSN: 1557-7341.

[160] B. Manoj und A. H. Baker, "Communication challenges in emergency response," *Communications of the ACM*, Jg. 50, Nr. 3, S. 51, März 2007.

[161] D. Mansour und C. Tschudin, "Towards a Monitoring Protocol Over Information-Centric Networks," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, Ser. ICN '17, New York, NY, USA: ACM, Sep. 2016, S. 60–64.

[162] A. M. Mehta, A. Bruns und J. Newton, "Trust, but verify: social media models for disaster management," *Disasters*, Jg. 41, Nr. 3, S. 549–565, Juli 2017.

[163] S. Meyer und P. R. Ward, "Reworking the Sociology of Trust: Making a Semantic Distinction between Trust and Dependence," in *Proceedings of the Australian Sociological Association Conference*, Australian National University, Canberra: The Australian Sociological Association, 2009, S. 1–16.

[164] S. M. Miran, C. Ling, J. J. James, A. Gerard und L. Rothfusz, "User perception and interpretation of tornado probabilistic hazard information: Comparison of four graphical designs," *Applied Ergonomics*, Jg. 65, S. 277–285, Nov. 2017.

[165] J. Mirkovic und P. Reiher, "Building accountability into the future Internet," in *2008 4th Workshop on Secure Network Protocols*, Piscataway, NJ, USA: IEEE, Okt. 2008, S. 45–51.

[166] M. A. Mishari, E. De Cristofaro, K. E. Defrawy und G. Tsudik, "Harvesting SSL Certificate Data to Identify Web-Fraud," *International Journal of Network Security*, Jg. 14, Nr. 6, S. 324–338, Sep. 2009, ISSN: 1816-353X.

[167] P. Mockapetris und K. J. Dunlap, "Development of the domain name system," in *Proceedings of ACM SIGCOMM*, New York, NY, USA: ACM, 1988, S. 123–133.

[168] I. Moiseenko und D. Oran, “TCP/ICN: Carrying TCP over Content Centric and Named Data Networks,” in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, Ser. ICN ’17, New York, NY, USA: ACM, Sep. 2016, S. 112–121.

[169] P. Moll, S. Theuermann, N. Rauscher, H. Hellwagner und J. Burke, “Inter-Server Game State Synchronization Using Named Data Networking,” in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, Ser. ICN ’19, Macao, China: ACM, 2019, S. 12–18.

[170] G. Möllering, “The Trust/Control Duality: An Integrative Perspective on Positive Expectations of Others,” *International Sociology*, Jg. 20, Nr. 3, S. 283–305, 2005.

[171] E. Monaghesh und A. Hajizadeh, “The role of telehealth during COVID-19 outbreak: a systematic review based on current evidence,” *BMC Public Health*, Jg. 20, Nr. 1, Aug. 2020.

[172] L. Mui, M. Mohtashemi und A. Halberstadt, “A computational model of trust and reputation,” in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, Jan. 2002, S. 2431–2439.

[173] M. Müller, T. Chung, A. Mislove und R. van Rijswijk-Deij, “Rolling with Confidence: Managing the Complexity of DNSSEC Operations,” *IEEE Transactions on Network and Service Management*, Jg. 16, Nr. 3, S. 1199–1211, 2019.

[174] M. Müller, M. Thomas, D. Wessels, W. Hardaker, T. Chung, W. Toorop und R. v. Rijswijk-Deij, “Roll, Roll, Roll Your Root: A Comprehensive Analysis of the First Ever DNSSEC Root KSK Rollover,” in *Proceedings of the Internet Measurement Conference*, Ser. IMC ’19, Amsterdam, Netherlands: ACM, 2019, S. 1–14, ISBN: 978-1-4503-6948-0.

[175] M. Müller, W. Toorop, T. Chung, J. Jansen und R. van Rijswijk-Deij, “The Reality of Algorithm Agility: Studying the DNSSEC Algorithm Life-Cycle,” in *Proceedings of the ACM Internet Measurement Conference*, Ser. IMC ’20, Virtual Event, USA: ACM, 2020, S. 295–308, ISBN: 9781450381383.

[176] X. Mwangi und K. Sollins, “MNDN: Scalable Mobility Support in Named Data Networking,” in *Proceedings of the 5th ACM Conference on Information-Centric Networking*, Ser. ICN ’18, New York, NY, USA: ACM, Sep. 2018, S. 117–124.

[177] M. Nawrocki, M. Jonker, T. C. Schmidt und M. Wählisch, “The far side of DNS amplification: tracing the DDoS attack ecosystem from the internet core,” Ser. IMC ’21, Virtual Event: ACM, 2021, S. 419–434, ISBN: 9781450391290.

[178] M. Nawrocki, P. F. Tehrani, R. Hiesgen, J. Mücke, T. C. Schmidt und M. Wählisch, “On the interplay between TLS certificates and QUIC performance,” in *Proceedings of the 18th International Conference on Emerging Networking EXperiments and Technologies*, Ser. CoNEXT ’22, Roma, Italy: ACM, 2022, S. 204–213, ISBN: 9781450395083.

[179] K. Nichols, "Lessons Learned Building a Secure Network Measurement Framework Using Basic NDN," in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, Ser. ICN '19, Macao, China: ACM, 2019, S. 112–122.

[180] H. Nissenbaum, "Securing trust online: Wisdom or oxymoron?" *Boston University Law Review*, Jg. 81, Nr. 3, S. 635–664, Juni 2001.

[181] H. Nissenbaum, "Computing and accountability," *Communications of the ACM*, Jg. 37, Nr. 1, S. 72–80, Jan. 1994.

[182] H. Nyquist, "Certain topics in telegraph transmission theory," *Transactions of the American Institute of Electrical Engineers*, Jg. 47, Nr. 2, S. 617–644, 1928.

[183] G. O'Reilly, A. Jrad, R. Nagarajan, T. Brown und S. Conrad, "Critical Infrastructure Analysis of Telecom for Natural Disasters," in *Networks 2006. 12th International Telecommunications Network Strategy and Planning Symposium*, Piscataway, NJ, USA: IEEE, 2006, S. 1–6, ISBN: 1-4244-0952-7.

[184] E. Oakes, J. Kline, A. Cahn, K. Funkhouser und P. Barford, "A Residential Client-side Perspective on SSL Certificates," in *2019 Network Traffic Measurement and Analysis Conference*, Ser. TMA '19, Piscataway, NJ, USA: IEEE, Juni 2019, S. 185–192, ISBN: 978-3-903176-17-1.

[185] B. Ohlman, "From ID/locator split to ICN," in *2015 12th Annual IEEE Consumer Communications and Networking Conference*, Ser. CCNC '12, IEEE Press, Jan. 2015, S. 256–261, ISBN: 9781479963904.

[186] E. Osterweil, P. Fotouhi Tehrani, T. C. Schmidt und M. Wahlisch, "From the Beginning: Key Transitions in the First 15 Years of DNSSEC," *IEEE Transactions on Network and Service Management*, Jg. 19, Nr. 4, S. 5265–5283, Dez. 2022, ISSN: 2373-7379.

[187] E. Osterweil, D. Massey und L. Zhang, "Deploying and Monitoring DNS Security (DNSSEC)," in *2009 Annual Computer Security Applications Conference*, 2009, S. 429–438.

[188] E. Osterweil, M. Ryan, D. Massey und L. Zhang, "Quantifying the operational status of the DNSSEC deployment," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, Ser. IMC '08, Vouliagmeni, Greece: ACM, 2008, S. 231–242, ISBN: 9781605583341.

[189] G. Ouvrier, M. Laterman, M. Arlitt und N. Carlsson, "Characterizing the HTTPS Trust Landscape: A Passive View from the Edge," *IEEE Communications Magazine*, Jg. 55, Nr. 7, S. 36–42, 2017.

[190] C. Partridge und M. Allman, "Ethical considerations in network measurement papers," *Communications of the ACM*, Jg. 59, Nr. 10, S. 58–64, Sep. 2016.

[191] E. Pauley und P. McDaniel, "Understanding the Ethical Frameworks of Internet Measurement Studies," in *Proceedings of the 2nd International Workshop on Binary Analysis Research*, Reston, VA, USA: Internet Society, 2023.

[192] S. N. Penn, “Copyright Law: Intellectual Property Protection in Cyberspace Case Comment,” *Journal of Technology Law & Policy*, Jg. 7, S. 263–274, 2002.

[193] D. Perino und M. Varvello, “A reality check for content centric networking,” in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, Ser. ICN ’11, New York, NY, USA: ACM, 2011, S. 44–49, ISBN: 9781450308014.

[194] S. Pletinckx, T.-D. Nguyen, T. Fiebig, C. Kruegel und G. Vigna, “Certifiably Vulnerable: Using Certificate Transparency Logs for Target Reconnaissance,” in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*, 2023, S. 817–831.

[195] J. Purushothaman, E. Thompson und A. Abdou, “Position Paper: Certificate Root Stores—An Area of Unity or Disparity?” In *Cyber Security Experimentation and Test Workshop*, New York, NY, USA: ACM, Aug. 2022, S. 105–110.

[196] R. Radu und M. Hausding, “Consolidation in the DNS resolver market–how much, how fast, how dangerous?” *Journal of Cyber Policy*, Jg. 5, Nr. 1, S. 46–64, 2020.

[197] R. S. Raman, L. Evdokimov, E. Wurstrow, J. A. Halderman und R. Ensafi, “Investigating Large Scale HTTPS Interception in Kazakhstan,” in *Proceedings of the ACM Internet Measurement Conference*, Ser. IMC ’20, Virtual Event, USA: ACM, 2020, S. 125–132, ISBN: 9781450381383.

[198] R. S. Raman, A. Stoll, J. Dalek, R. Ramesh, W. Scott und R. Ensafi, “Measuring the Deployment of Network Censorship Filters at Global Scale,” in *Proceedings 2020 Network and Distributed System Security Symposium*, Reston, VA, USA: Internet Society, 2020.

[199] S. K. Ramani, R. Tourani, G. Torres, S. Misra und A. Afanasyev, “NDN-ABS: Attribute-Based Signature Scheme for Named Data Networking,” in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, Ser. ICN ’19, New York, NY, USA: ACM, Sep. 2019, S. 123–133.

[200] B. Reynolds und M. W. Seeger, “Crisis and emergency risk communication as an integrative model,” *Journal of Health Communication*, Jg. 10, Nr. 1, S. 43–55, 2005, ISSN: 1081-0730.

[201] J. Reynolds, D. Kumar, Z. Ma, R. Subramanian, M. Wu, M. Shelton, J. Mason, E. Stark und M. Bailey, “Measuring Identity Confusion with Uniform Resource Locators,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, Ser. CHI ’20, Honolulu, HI, USA: ACM, 2020, S. 1–12, ISBN: 9781450367080.

[202] D. Rezende, C. Maziero und E. Mannes, “A distributed online certificate status protocol for named data networks,” in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, New York, NY, USA: ACM, Apr. 2018, S. 2102–2108.

[203] R. van Rijswijk-Deij, T. Chung, D. Choffnes, A. Mislove und W. Toorop, “The Root Canary: Monitoring and Measuring the DNSSEC Root Key Rollover,” in *Proceedings of the SIGCOMM Posters and Demos*, Ser. SIGCOMM Posters and Demos ’17, Los Angeles, CA, USA: ACM, 2017, S. 63–64, ISBN: 9781450350570.

[204] RIPE NCC Staff, “RIPE atlas: A global internet measurement network,” *Internet Protocol Journal*, Jg. 18, Nr. 3, S. 2–26, 2015.

[205] R. Roberts, Y. Goldschlag, R. Walter, T. Chung, A. Mislove und D. Levin, “You Are Who You Appear to Be: A Longitudinal Study of Domain Impersonation in TLS Certificates,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, Ser. CCS ’19, London, United Kingdom: ACM, 2019, S. 2489–2504, ISBN: 9781450367479.

[206] R. Roberts, D. Lulli, A. Raut, K. Fulton und D. L. Levin, “Mental Models of Domain Names and URLs,” in *Proceedings of 16th SOUPS*, Berkeley, CA, USA: USENIX Association, Aug. 2020, S. 5.

[207] S. B. Roosa und S. Schultze, “Trust Darknet: Control and Compromise in the Internet’s Certificate Authority Model,” *IEEE Internet Computing*, Jg. 17, Nr. 3, S. 18–25, Mai 2013, ISSN: 1089-7801.

[208] S. Rose, “Progress of DNS Security Deployment in the Federal Government,” in *Proceedings of the 26th LISA*, Berkeley, CA, USA: USENIX Association, 2012, S. 223–228.

[209] D. M. Rousseau, S. B. Sitkin, R. S. Burt und C. Camerer, “Not So Different After All: A Cross-Discipline View Of Trust,” *Academy of Management Review*, Jg. 23, Nr. 3, S. 393–404, Juli 1998.

[210] K. Ruth, A. Fass, J. Azose, M. Pearson, E. Thomas, C. Sadowski und Z. Durumeric, “A world wide view of browsing the world wide web,” in *Proceedings of the 22nd ACM Internet Measurement Conference*, Ser. IMC ’22, Nice, France: ACM, 2022, S. 317–336, ISBN: 9781450392594.

[211] K. Ruth, D. Kumar, B. Wang, L. Valenta und Z. Durumeric, “Toppling top lists: evaluating the accuracy of popular website lists,” in *Proceedings of the 22nd ACM Internet Measurement Conference*, Ser. IMC ’22, Nice, France: ACM, 2022, S. 374–387, ISBN: 9781450392594.

[212] C.-A. Sarros, S. Diamantopoulos, S. Rene, I. Psaras, A. Lertsinsrubtavee, C. Molina-Jimenez, P. Mendes, R. Sofia, A. Sathiaseelan, G. Pavlou, J. Crowcroft und V. Tsaoussidis, “Connecting the Edges: A Universal, Mobile-Centric, and Opportunistic Communications Architecture,” *IEEE Communications Magazine*, Jg. 56, Nr. 2, S. 136–143, Feb. 2018, ISSN: 0163-6804.

[213] Q. Scheitle, T. Chung, J. Hiller, O. Gasser, J. Naab, R. van Rijswijk-Deij, O. Hohlfeld, R. Holz, D. Choffnes, A. Mislove und G. Carle, “A First Look at Certification Authority Authorization (CAA),” *SIGCOMM Comput. Commun. Rev.*, Jg. 48, Nr. 2, S. 10–23, Mai 2018, ISSN: 0146-4833.

[214] Q. Scheitle, O. Gasser, T. Nolte, J. Amann, L. Brent, G. Carle, R. Holz, T. C. Schmidt und M. Wählisch, “The Rise of Certificate Transparency and Its Implications on the Internet Ecosystem,” in *Proceedings of the Internet Measurement Conference 2018*, Ser. IMC ’18, Boston, MA, USA: ACM, 2018, S. 343–349, ISBN: 9781450356190.

[215] Q. Scheitle, M. Wählisch, O. Gasser, T. C. Schmidt und G. Carle, “Towards an Ecosystem for Reproducible Research in Computer Networking,” in *Proceedings of the Reproducibility Workshop*, Ser. Reproducibility ’17, Los Angeles, CA, USA: ACM, 2017, S. 5–8, ISBN: 9781450350600.

[216] C. D. Schultz, “A trust framework model for situational contexts,” in *Proceedings of the 2006 International Conference on Privacy, Security and Trust Bridge the Gap Between PST Technologies and Business Services - PST ’06*, ACM, 2006.

[217] L. Schwittmann, M. Wander und T. Weis, “Domain Impersonation is Feasible: A Study of CA Domain Validation Vulnerabilities,” in *2019 IEEE EuroS&P*, Piscataway, NJ, USA: IEEE, 2019, S. 544–559.

[218] M. Seckler, S. Heinz, S. Forde, A. N. Tuch und K. Opwis, “Trust and distrust on the web: User experiences and website characteristics,” *Computers in Human Behavior*, Jg. 45, S. 39–50, 2015, ISSN: 0747-5632.

[219] J. Seedorf, D. Kutscher und F. Schneider, “Decentralised binding of self-certifying names to real-world identities for assessment of third-party messages in fragmented mobile networks,” in *2014 IEEE Conference on Computer Communications Workshops*, Piscataway, NJ, USA: IEEE, Apr. 2014, S. 416–421.

[220] J. Seedorf, A. Tagami, M. Arumaithurai, Y. Koizumi, N. B. Melazzi, D. Kutscher, K. Sugiyama, T. Hasegawa, T. Asami, K. K. Ramakrishnan, T. Yagyu und I. Psaras, “The Benefit of Information Centric Networking for Enabling Communications in Disaster Scenarios,” in *2015 IEEE Globecom Workshops*, Piscataway, NJ, USA: IEEE, Dez. 2015, S. 1–7, ISBN: 978-1-4673-9526-7.

[221] C. E. Shannon, “Communication in the presence of noise,” *Proceedings of the IRE*, Jg. 37, Nr. 1, S. 10–21, Jan. 1949.

[222] A. Sharma, X. Tie, H. Uppal, A. Venkataramani, D. Westbrook und A. Yadav, “A global name service for a highly mobile internetwork,” *ACM SIGCOMM Computer Communication Review*, Jg. 44, Nr. 4, S. 247–258, Aug. 2014, ISSN: 01464833.

[223] K. Shilton, J. A. Burke, K. Claffy und L. Zhang, “Anticipating policy and social implications of named data networking,” *Communications of the ACM*, Jg. 59, Nr. 12, S. 92–101, Dez. 2016.

[224] B. Shobiri, M. Mannan und A. Youssef, “CDNs’ Dark Side: Security Problems in CDN-to-Origin Connections,” *Digital Threats: Research and Practice*, Jg. 4, Nr. 1, S. 1–22, März 2023.

[225] P. Slovic, “Trust, emotion, sex, politics, and science: Surveying the risk-assessment battlefield,” *Risk Analysis*, Jg. 19, Nr. 4, S. 689–701, 1999, ISSN: 1539-6924.

[226] C. Soghoian und S. Stamm, “Certified Lies: Detecting and Defeating Government Interception Attacks Against SSL,” *SSRN Electronic Journal*, Jg. 7035 LNCS, S. 250–259, 2010, ISSN: 1556-5068.

[227] M. Sosnowski, F. Wiedner, E. Hauser, L. Steger, D. Schoinianakis, S. Gallenmüller und G. Carle, “The Performance of Post-Quantum TLS 1.3,” in *Companion of the 19th International Conference on Emerging Networking EXperiments and Technologies*, Ser. CoNEXT ’23, Paris, France: ACM, 2023, S. 19–27, ISBN: 9798400704079.

[228] D. Springall, Z. Durumeric und J. A. Halderman, "Measuring the Security Harm of TLS Crypto Shortcuts," in *Proceedings of the 2016 Internet Measurement Conference*, Ser. IMC '16, Santa Monica, California, USA: ACM, 2016, S. 33–47, ISBN: 9781450345262.

[229] I. Stoica, D. Adkins, S. Zhuang, S. Shenker und S. Surana, "Internet indirection infrastructure," *ACM SIGCOMM Computer Communication Review*, Jg. 32, Nr. 4, S. 73, Okt. 2002, ISSN: 0146-4833.

[230] M. Stojmenović und R. Biddle, "Who are They? Website Authentication: Certificates and Identity," in *Proceedings of 14th SOUPS*, Berkeley, CA, USA: USENIX Association, 2018.

[231] D. Stolle und L. Nishikawa, "Trusting Others - How Parents Shape the Generalized Trust of Their Children," *Comparative Sociology*, Jg. 10, Nr. 2, S. 281–314, 2011.

[232] H. Suzuki, D. Chiba, Y. Yoneya, T. Mori und S. Goto, "ShamFinder: An Automated Framework for Detecting IDN Homographs," in *Proceedings of the Internet Measurement Conference*, Ser. IMC '19, Amsterdam, Netherlands: ACM, 2019, S. 449–462, ISBN: 9781450369480.

[233] E. Sy, C. Burkert, H. Federrath und M. Fischer, "Tracking Users across the Web via TLS Session Resumption," in *Proceedings of the 34th Annual Computer Security Applications Conference*, New York, NY, USA: ACM, Dez. 2018, S. 289–299.

[234] E. Sy, M. Moennich, T. Mueller, H. Federrath und M. Fischer, "Enhanced performance for the encrypted web through TLS resumption across hostnames," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, New York, NY, USA: ACM, 2020, S. 1–10.

[235] A. Tagami und M. Arumaithurai, "GreenICN Project: Architecture and Applications of Green Information Centric Networking," *IEICE Transactions on Communications*, Jg. E99.B, Nr. 12, S. 2470–2476, 2016, ISSN: 0916-8516.

[236] A. Tagami, T. Yagyu, K. Sugiyama, M. Arumaithurai, K. Nakamura, T. Hasegawa, T. Asami und K. K. Ramakrishnan, "Name-based push/pull message dissemination for disaster message board," *IEEE Workshop on Local and Metropolitan Area Networks*, S. 1–6, 2016, ISSN: 19440375.

[237] L. Tamilina und N. Tamilina, "Trust as a Skill," *Psychology and Developing Societies*, Jg. 30, Nr. 1, S. 44–80, Feb. 2018.

[238] O. Tange, "GNU Parallel: The Command-Line Power Tool," *;login: The USENIX Magazine*, Jg. 36, Nr. 1, S. 42–47, Feb. 2011.

[239] A. H. Tapia, K. Bajpai, J. Jansen, J. Yen und L. Giles, "Seeking the Trustworthy Tweet: Can Microblogged Data Fit the Information Needs of Disaster Response and Humanitarian Relief Organizations," in *Proceedings of the 8th International ISCRAM Conference*, Ser. ISCRAM '11, 2011.

[240] J. Thompson, P. Gusev und J. Burke, “NDN-CNL: A Hierarchical Namespace API for Named Data Networking,” in *Proceedings of the 6th ACM Conference on Information-Centric Networking*, Ser. ICN ’19, Macao, China: ACM, 2019, S. 30–36.

[241] K. Tian, S. T. K. Jan, H. Hu, D. Yao und G. Wang, “Needle in a Haystack: Tracking Down Elite Phishing Domains in the Wild,” in *Proceedings of the Internet Measurement Conference 2018*, Ser. IMC ’18, Boston, MA, USA: ACM, 2018, S. 429–442, ISBN: 9781450356190.

[242] R. Tourani, S. Misra, J. Kliewer, S. Ortegel und T. Mick, “Catch Me If You Can: A Practical Framework to Evade Censorship in Information-Centric Networks,” in *Proceedings of the 2nd International Conference on Information-Centric Networking*, Ser. ICN ’15, New York, NY, USA: ACM, 2015, S. 167–176.

[243] R. Tourani, S. Misra, T. Mick und G. Panwar, “Security, Privacy, and Access Control in Information-Centric Networking: A Survey,” *IEEE Communications Surveys & Tutorials*, Jg. 20, Nr. 1, S. 566–600, 2018, ISSN: 1553-877X.

[244] D. Trossen, M. Sarela und K. Sollins, “Arguments for an information-centric internetworking architecture,” *ACM SIGCOMM Computer Communication Review*, Jg. 40, Nr. 2, S. 26, Apr. 2010, ISSN: 01464833.

[245] G. Tyson, E. Bodanese, J. Bigham und A. Mauthe, “Beyond content delivery: can ICNs help emergency scenarios?” *IEEE Network*, Jg. 28, Nr. 3, S. 44–49, Mai 2014, ISSN: 0890-8044.

[246] S. Vallentin und N. Thygesen, “Trust and Control in Public Sector Reform: Complementarity and Beyond,” *Journal of Trust Research*, Jg. 7, Nr. 2, S. 150–169, 2017.

[247] B. VanderSloot, J. Amann, M. Bernhard, Z. Durumeric, M. Bailey und J. A. Halderman, “Towards a Complete View of the Certificate Ecosystem,” in *Proceedings of the 2016 Internet Measurement Conference*, Ser. IMC ’16, Santa Monica, California, USA: ACM, 2016, S. 543–549, ISBN: 9781450345262.

[248] B. S. Vanneste, “From Interpersonal to Interorganisational Trust: The Role of Indirect Reciprocity,” *Journal of Trust Research*, Jg. 6, Nr. 1, S. 7–36, 2016.

[249] A. Venkataramani, J. F. Kurose, D. Raychaudhuri, K. Nagaraja, M. Mao und S. Banerjee, “MobilityFirst: a mobility-centric and trustworthy internet architecture,” *SIGCOMM Comput. Commun. Rev.*, Jg. 44, Nr. 3, S. 74–80, Juli 2014, ISSN: 0146-4833.

[250] M. Wählisch, T. C. Schmidt und M. Vahlenkamp, “Backscatter from the Data Plane – Threats to Stability and Security in Information-Centric Network Infrastructure,” *Computer Networks*, Jg. 57, Nr. 16, S. 3192–3206, Nov. 2013.

[251] J. B. Walther, Z. Wang und T. Loh, “The effect of top-level domains and advertisements on health web-site credibility,” *Journal of Medical Internet Research*, Jg. 6, Nr. 3, S. 1–11, 2004, ISSN: 1438-8871.

[252] G. Wan, L. Izhikevich, D. Adrian, K. Yoshioka, R. Holz, C. Rossow und Z. Durumeric, “On the Origin of Scanning: The Impact of Location on Internet-Wide Scans,” in *Proceedings of the ACM Internet Measurement Conference*, Ser. IMC ’20, Virtual Event, USA: ACM, 2020, S. 662–679, ISBN: 9781450381383.

[253] M. Wander und T. Weis, “Measuring Occurrence of DNSSEC Validation,” in *Passive and Active Measurement*, M. Roughan und R. Chang, Hrsg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, S. 125–134, ISBN: 978-3-642-36516-4.

[254] Y. D. Wang und H. H. Emurian, “An overview of online trust: Concepts, elements, and implications,” *Computers in Human Behavior*, Jg. 21, Nr. 1, S. 105–125, Jan. 2005.

[255] Z. Wang, “A revisit of DNS Kaminsky cache poisoning attacks,” in *2015 IEEE GLOBECOM*, IEEE, 2015, S. 1–6.

[256] Z. Wang und L. Xiao, “Emergency Key Rollover in DNSSEC,” in *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, Piscataway, NJ, USA: IEEE, 2014, S. 598–604.

[257] M. R. Welch, R. E. N. Rivera, B. P. Conway, J. Yonkoski, P. M. Lupton und R. Giancola, “Determinants and Consequences of Social Trust*,” *Sociological Inquiry*, Jg. 75, Nr. 4, S. 453–473, 2005.

[258] W. Wong und P. Nikander, “Secure naming in information-centric networks,” in *Proceedings of the Re-Architecting the Internet Workshop*, Ser. ReARCH ’10, New York, New York, USA: ACM, 2010, ISBN: 9781450304696.

[259] Z. Yan, R. Kantola und P. Zhang, “A Research Model for Human-Computer Trust Interaction,” in *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, Piscataway, NJ, USA: IEEE, Nov. 2011.

[260] W. Yang, F. Wu und K. Tian, “High performance adaptive video streaming using NDN WLAN multicast,” in *Proceedings of the 8th ACM Conference on Information-Centric Networking*, Ser. ICN ’21, New York, NY, USA: ACM, Sep. 2021, S. 42–51.

[261] T. Yu, H. Xie, S. Liu, X. Ma, X. Jia und L. Zhang, “CertRevoke: a certificate revocation framework for named data networking,” in *Proceedings of the 9th ACM Conference on Information-Centric Networking*, Ser. ICN ’22, Osaka, Japan: ACM, 2022, S. 80–90, ISBN: 9781450392570.

[262] Y. Yu, A. Afanasyev, D. Clark, k. claffy kc, V. Jacobson und L. Zhang, “Schematizing Trust in Named Data Networking,” in *Proceedings of the 2nd International Conference on Information-Centric Networking*, Ser. ICN ’15, San Francisco, California, USA: ACM, 2015, S. 177–186.

[263] Y. Yu, A. Afanasyev, J. Seedorf, Z. Zhang und L. Zhang, “NDN DeLorean: An Authentication System for Data Archives in Named Data Networking,” in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, Ser. ICN ’17, Berlin, Germany: ACM, 2017, S. 11–21.

[264] H. Zhang, Z. Wang, C. Scherb, C. Marxer, J. Burke, L. Zhang und C. Tschudin, “Sharing MHealth Data via Named Data Networking,” in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, Ser. ICN ’16, Kyoto, Japan: ACM, 2016, S. 142–147.

[265] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang und B. Zhang, “Named data networking,” *ACM SIGCOMM Computer Communication Review*, Jg. 44, Nr. 3, S. 66–73, Juli 2014, ISSN: 01464833.

[266] M. Zhang, X. Zheng, K. Shen, Z. Kong, C. Lu, Y. Wang, H. Duan, S. Hao, B. Liu und M. Yang, “Talking with Familiar Strangers: An Empirical Study on HTTPS Context Confusion Attacks,” in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, Ser. CCS ’20, Virtual Event, USA: ACM, 2020, S. 1939–1952, ISBN: 9781450370899.

[267] M. Zhang, V. Lehman und L. Wang, “Scalable name-based data synchronization for named data networking,” in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, Piscataway, NJ, USA: IEEE, Mai 2017.

[268] X. Zhang, K. Chang, H. Xiong, Y. Wen, G. Shi und G. Wang, “Towards name-based trust and security for content-centric network,” in *2011 19th IEEE International Conference on Network Protocols*, Piscataway, NJ, USA: IEEE, Okt. 2011, S. 1–6.

[269] Y. Zhang, B. Liu, C. Lu, Z. Li, H. Duan, J. Li und Z. Zhang, “Rusted Anchors: A National Client-Side View of Hidden Root CAs in the Web PKI Ecosystem,” in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA: ACM, Nov. 2021, S. 1373–1387.

[270] Y. Zhang, Z. Xia, S. Mastorakis und L. Zhang, “KITE: Producer Mobility Support in Named Data Networking,” in *Proceedings of the 5th ACM Conference on Information-Centric Networking*, Ser. ICN ’18, New York, NY, USA: ACM, Sep. 2018, S. 125–136.

[271] Z. Zhang, A. Afanasyev und L. Zhang, “NDNCERT: Universal usable trust management for NDN,” in *Proceedings of the 4th ACM Conference on Information Centric Networking*, Ser. ICN ’17, New York, NY, USA: ACM, 2017, S. 178–179, ISBN: 9781450351225.

[272] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev und L. Zhang, “An Overview of Security Support in Named Data Networking,” *IEEE Communications Magazine*, Jg. 56, Nr. 11, S. 62–68, Nov. 2018, ISSN: 0163-6804.

[273] Z. Zhu und A. Afanasyev, “Let’s ChronoSync: Decentralized Dataset State Synchronization in Named Data Networking,” in *2013 21st IEEE International Conference on Network Protocols*, Ser. ICNP ’13, Piscataway, NJ, USA: IEEE, Okt. 2013, S. 1–10.

[274] J. Zirngibl, P. Buschmann, P. Sattler, B. Jaeger, J. Aulbach und G. Carle, “It’s over 9000: analyzing early QUIC deployments with the standardization on the horizon,” in *Proceedings of the 21st ACM Internet Measurement Conference*, Ser. IMC ’21, Virtual Event: ACM, 2021, S. 261–275, ISBN: 9781450391290.

[275] L. G. Zucker, “Production of Trust: Institutional Sources of Economic Structure, 1840–1920,” *Research in organizational behavior : an annual series of analytical essays and critical reviews*, Jg. 8, S. 53–111, 1986.

Other References

[276] 104th U.S. Congress, *Telecommunications Act of 1996*, Feb. 1996. URL: `https://www.congress.gov/bill/104th-congress/senate-bill/652` (besucht am 03. 07. 2020).

[277] M. I. Abdelhamid, *Die Ökonomisierung des Vertrauens: Eine Kritik gegenwärtiger Vertrauensbegriffe [The Economization of Trust: A Critique of Current Concepts of Trust]*. Bielefeld: transcript, 2018, ISBN: 978-3-8376-4205-6.

[278] ACM. “Artifact Review and Badging.” Version 1.1, ACM. (24. Aug. 2020), URL: `https://www.acm.org/publications/policies/artifact-review-and-badging-current` (besucht am 28. 07. 2023).

[279] A. Afanasyev, “Addressing Operational Challenges in Named Data Networking Through NDNS Distributed Database,” Diss., University of California Los Angeles, 2013.

[280] Akamai. “What Is Bot Detection?” Akamai. (15. Mai 2023), URL: `https://www.akamai.com/glossary/what-is-bot-detection` (besucht am 07. 06. 2023).

[281] J. Althouse. “Easily Identify Malicious Servers on the Internet with JARM,” salesforce. (17. Nov. 2020), URL: `https://engineering.salesforce.com/easily-identify-malicious-servers-on-the-internet-with-jarm-e095edac525a/` (besucht am 02. 11. 2023).

[282] Amazon. “Alexa Top Sites.” (), URL: `https://www.alexa.com/`.

[283] J. Andress, *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice* (Syngress basics series). Elsevier Science, 2014, ISBN: 9780128008126.

[284] Apple. “Apple’s Certificate Transparency policy,” Apple. (5. März 2021), URL: `https://support.apple.com/en-us/HT205280` (besucht am 23. 05. 2023).

[285] Apple. “Apple Root Certificate Program,” Apple Inc. (Feb. 2022), URL: `https://www.apple.com/certificateauthority/ca_program.html` (besucht am 19. 05. 2022).

[286] Apple. “List of certificates with additional constraints,” Apple. (2. Dez. 2022), URL: `https://support.apple.com/en-us/HT212865` (besucht am 23. 05. 2023).

[287] M. Arumaithurai, A. Detti, T. Hasegawa, J. Seedorf und S. S. Adhatarao, “GreenICN - Deliverable D1.1.3c: Final GreenICN Architecture,” GreenICN Consortium, Techn. Ber., Apr. 2016, S. 1–55. URL: `http://www.greenicn.org/deliverables/deliverables/`.

[288] D. Atkins und R. Austein, *Threat Analysis of the Domain Name System (DNS)*, RFC 3833, Aug. 2004. URL: `https://www.rfc-editor.org/info/rfc3833`.

[289] T. Backes, W. Jaschensky, K. Langhans, H. Munzinger, B. Witzenberger und V. Wormer. “Timeline der Panik.” (1. Okt. 2016), URL: `https://sz.de/panik`.

[290] S. Bailey, J. Mogul, D. A. Romanow und T. Talpey, *Remote Direct Memory Access (RDMA) over IP Problem Statement*, RFC 4297, Dez. 2005. URL: `https://www.rfc-editor.org/info/rfc4297`.

[291] R. Barnes, *Use Cases and Requirements for DNS-Based Authentication of Named Entities (DANE)*, RFC 6394, Okt. 2011. URL: `https://www.rfc-editor.org/info/rfc6394`.

[292] R. Barnes, J. Hoffman-Andrews, D. McCarney und J. Kasten, *Automatic Certificate Management Environment (ACME)*, RFC 8555, März 2019. URL: `https://www.rfc-editor.org/info/rfc8555`.

[293] M. Belshe, R. Peon und M. Thomson, *Hypertext Transfer Protocol Version 2 (HTTP/2)*, RFC 7540, Mai 2015. URL: `https://www.rfc-editor.org/info/rfc7540`.

[294] S. Bingler. “CRLSets,” Google. (7. Juli 2023), URL: `https://chromium.googlesource.com/website/+/refs/heads/main/site/Home/chromium-security/crlsets/index.md` (besucht am 13. 07. 2023).

[295] M. Bishop, *HTTP/3*, RFC 9114, Juni 2022. URL: `https://www.rfc-editor.org/info/rfc9114`.

[296] B. W. Blanchard, *Guide to Emergency Management and related Terms, Definitions, Concepts, Acronyms, Organizations, Programs, Guidance, Executive Orders & Legislation*, 27. Okt. 2008. URL: `https://web.archive.org/web/20210507022425/https://training.fema.gov/hiedu/docs/terms%20and%20definitions/Terms%20and%20Definitions.pdf` (besucht am 14. 01. 2020).

[297] S. Boeyen, S. Santesson, T. Polk, R. Housley, S. Farrell und D. Cooper, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC 5280, Mai 2008. URL: `https://www.rfc-editor.org/info/rfc5280`.

[298] J. S. Brennen, F. Simon, P. N. Howard und R. K. Nielsen, *Types, sources, and claims of COVID-19 misinformation*, 7. Apr. 2020. URL: `https://reutersinstitute.politics.ox.ac.uk/types-sources-and-claims-covid-19-misinformation`.

[299] CA/Browser Forum, *Information for the Public*, 30. Aug. 2013. URL: `https://cabforum.org/info-for-consumers/`.

[300] CA/Browser Forum. “Ballot 187 – Make CAA Checking Mandatory,” CA/Browser Forum. (8. März 2017), URL: `https://cabforum.org/2017/03/08/ballot-187-make-caa-checking-mandatory/` (besucht am 30. 05. 2024).

[301] CA/Browser Forum. “Ballot 195 – CAA Fixup.” (17. Apr. 2017), URL: `https://cabforum.org/2017/04/17/ballot-195-caa-fixup/` (besucht am 25. 06. 2024).

[302] CA/Browser Forum. “Ballot 214 – CAA Discovery CNAME Errata.” (27. Sep. 2017), URL: `https://cabforum.org/2017/09/27/ballot-214-caa-discovery-cname-errata/` (besucht am 25. 06. 2024).

[303] CA/Browser Forum. “Ballot SC13: CAA Contact Property and Associated E-mail Validation Methods.” (24. Dez. 2018), URL: `https://cabforum.org/2018/12/18/ballot-sc13-caa-contact-property-and-associated-e-mail-validation-methods/` (besucht am 26. 06. 2024).

[304] CA/Browser Forum. “Ballot SC18: Phone Contact with DNS CAA Phone Contact.” (23. Apr. 2018), URL: `https://cabforum.org/2019/05/01/ballot-sc18-phone-contact-with-dns-caa-phone-contact/` (besucht am 30. 05. 2024).

[305] CA/Browser Forum, *Guidelines for The Issuance And Management of Extended Validation Certificates*, 2021.

[306] CA/Browser Forum, *Guidelines for the Issuance and Management of Extended Validation Certificates*, Nov. 2022.

[307] CA/Browser Forum, *Baseline Requirements for the Issuance and Management of Publicly-Trusted TLS Server Certificates*, Online, Jan. 2024. URL: `https://cabforum.org/uploads/CA-Browser-Forum-TLS-BRs-v2.0.2.pdf`.

[308] CA/Browser Forum. “Object Registry of the CA/Browser Forum,” CA / Browser Forum. (), URL: `https://cabforum.org/object-registry/` (besucht am 24. 02. 2024).

[309] C. Castelfranchi und R. Falcone, *Trust Theory: A Socio-Cognitive and Computational Model* (Wiley Series in Agent Technology). Chichester: Wiley, 2010, ISBN: 9780470519844.

[310] W. K. Chai, I. Psaras, M. Charalambides, D. Tuncer, G. Pavlou, D. Flórez, F. J. H. Romero, A. Beben, J. M. Batalla, J. Sliwinski, P. Wisniewski, W. Burakowski, N. Wang, G. Kamel, S. Spirou und M. Georgiades, “COMET - Deliverable D3.2: Final Specification of Mechanisms, Protocols and Algorithms for the Content Mediation System,” COMET Consortium, Techn. Ber., Juni 2012. URL: `http://www.comet-project.org/deliverables.html`.

[311] D. R. Cheriton und M. Gritter, “TRIAD: A New Next-Generation Internet Architecture,” Distributed Systems Group, Stanford University, Techn. Ber., 2000.

[312] Cisco. “Cisco Umbrella 1 Million.” (Juli 2020), URL: `https://umbrella.cisco.com/blog/cisco-umbrella-1-million`.

[313] N. Consortium. “ndn-cxx: NDN C++ library with eXperimental eXtensions 0.6.5.” (2019), URL: `https://web.archive.org/web/20190325105322/http://named-data.net/doc/ndn-cxx/0.6.5/release-notes/release-notes-0.6.5.html` (besucht am 12. 05. 2019).

[314] A. Cooper und J. Postel, *The US Domain*, RFC 1386, Dez. 1992. URL: `https://www.rfc-editor.org/info/rfc1386`.

[315] A. Cooper und J. Postel, *The US Domain*, RFC 1480, Juni 1993. URL: `https://www.rfc-editor.org/info/rfc1480`.

[316] W. E. D. Creed und R. E. Miles, “Trust in Organizations: A Conceptual Framework Linking Organizational Forms, Managerial Philosophies, and the Opportunity Costs of Controls,” in *Trust in Organizations: Frontiers of Theory and Research*, Thousand Oaks, CA, US: Sage Publications, Inc, 1996, S. 16–38.

[317] J. Day, *Patterns in Network Architecture: A Return to Fundamentals*. Pearson Education, 2007, ISBN: 9780132704564.

[318] W. E. Deming, *Out of the Crisis*. MIT Press, 1982.

[319] S. DiBenedetto, P. Gasti, G. Tsudik und E. Uzun, *ANDaNA: Anonymous Named Data Networking Application*, Dez. 2011.

[320] DigiCert. “Certification Practices Statement,” DigiCert. (Jan. 2024), URL: `https://www.digicert.com/content/dam/digicert/pdfs/legal/digicert-cps-v6-05.pdf`.

[321] digicert + QuoVadis, *QuoVadis Root CA 2 / QuoVadis Root CA 2 G3 Certification Policy / Certification Practice Statement*, Version 2.20, März 2023.

[322] DigiCert, Inc., *Certificate Policy*, Version 5.16, Mai 2023.

[323] P. T. Dinesen und R. Bekkers, “The Foundations of Individuals’ Generalized Social Trust,” in *Trust in Social Dilemmas*, P. A. V. Lange, B. Rockenbach und T. Yamagishi, Hrsg., Oxford University Press, Juli 2017.

[324] DomainTools. “Domain Count Statistics for TLDs.” [Accessed 2023-05-23], DomainTools. (23. Mai 2023), URL: `https://research.domaintools.com/statistics/tld-counts/`.

[325] Domo. “Media usage in an internet minute as of April 2022.,” Statista. (21. Sep. 2022), URL: `https://www.statista.com/statistics/195140/new-user-generated-content-uploaded-by-users-per-minute/` (besucht am 22. 06. 2023).

[326] DotGov. “Domain requirements.” (2020), URL: `https://www.about.us/faqs` (besucht am 05. 07. 2020).

[327] V. Dukhovni, S. Huque, W. Toorop, P. Wouters und M. Shore, *TLS DNSSEC Chain Extension*, RFC 9102, Aug. 2021. URL: `https://www.rfc-editor.org/info/rfc9102`.

[328] E. Durkheim, *The Division of Labor in Society*. New York, London, Toronto, Sydney, New Delhi: Free Press, 2014, ISBN: 978-1-4391-1824-5.

[329] D. E. Eastlake, *Transport Layer Security (TLS) Extensions: Extension Definitions*, RFC 6066, Jan. 2011. URL: `https://www.rfc-editor.org/info/rfc6066`.

[330] R. Elz und R. Bush, *Clarifications to the DNS Specification*, RFC 2181, Juli 1997. URL: `https://www.rfc-editor.org/info/rfc2181`.

[331] E. H. Erikson, *Childhood and Society*. New York: W. W. Norton, 1993, ISBN: 9780393347388.

[332] European Parliament, Council of the European Union, *Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC*, Aug. 2014. URL: `http://data.europa.eu/eli/reg/2014/910/oj`.

[333] S. Farrell, D. Kutscher, C. Dannewitz, B. Ohlman, A. Keränen und P. Hallam-Baker, *Naming Things with Hashes*, RFC 6920, Apr. 2013. URL: `https://www.rfc-editor.org/info/rfc6920`.

[334] FEMA, *Alerting Authorities*, 2021. URL: `https://www.fema.gov/emergency-managers/practitioners/integrated-public-alert-warning-system/public-safety-officials/alerting-authorities`.

[335] FEMA, *Integrated Public Alert & Warning System*, 2. Feb. 2022. URL: `https://www.fema.gov/emergency-managers/practitioners/integrated-public-alert-warning-system`.

[336] R. T. Fielding, M. Nottingham und J. Reschke, *HTTP Semantics*, RFC 9110, Juni 2022. URL: `https://www.rfc-editor.org/info/rfc9110`.

[337] S. Fitzpatrick. “We are changing how Majestic reports backlink counts,” Majestic. (11. Aug. 2021), URL: `https://blog.majestic.com/company/we-are-changing-how-majestic-reports-backlink-counts/` (besucht am 22. 05. 2023).

[338] D. W. S. Ford, D. S. Chokhani, S. S. Wu, R. V. Sabett und C. R. Merrill, *Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework*, RFC 3647, Nov. 2003. URL: `https://www.rfc-editor.org/info/rfc3647`.

[339] C. Forum. “Ballot 219 – Clarify handling of CAA Record Sets with no “issue”/“issuewild” property tag,” CA/Browser Forum. (10. Apr. 2018), URL: `https://cabforum.org/2018/04/10/ballot-219-clarify-handling-of-caa-record-sets-with-no-issue/issuewild-property-tag/` (besucht am 28. 05. 2024).

[340] N. Fotiou, S. Arianfar, M. Särelä und G. C. Polyzos, “A Framework for Privacy Analysis of ICN Architectures,” in *Privacy Technologies and Policy*, Ser. Lecture Notes in Computer Science, B. Preneel und D. Ikonomou, Hrsg., Bd. 8450, Cham, Switzerland: Springer International Publishing, 2014, S. 117–132, ISBN: 9783319067483.

[341] P. Fotouhi Tehrani, R. Hiesgen, T. C. Schmidt und M. Wählisch, *A Call to Reconsider Certification Authority Authorization (CAA)*, 2024.

[342] P. Fotouhi Tehrani, E. Osterweil, T. C. Schmidt und M. Wählisch, *How to Measure TLS, X.509 Certificates, and Web PKI: A Tutorial and Brief Survey*, 2024.

[343] A. Frik und L. Mittone, “Factors Influencing the Perceived Websites’ Privacy Trustworthiness and Users’ Purchase Intentions,” Cognitive und Experimental Economics Laboratory, Department of Economics, University of Trento, Italia, CEEL Working Papers 1609, 2016.

[344] F. Fukuyama, *Trust: The Social Virtues and the Creation of Prosperity*. New York, NY, USA: Free Press, 1995, ISBN: 978-0-684-82525-0.

[345] S. Galperin, D. C. Adams, M. Myers, R. Ankney und A. N. Malpani, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*, RFC 2560, Juni 1999. URL: `https://www.rfc-editor.org/info/rfc2560`.

[346] D. Gambetta, "Can we trust trust?" In *Trust: Making and Breaking Cooperative Relations*, Cambridge: B. Blackwell, 1988, S. 213–239, ISBN: 978-0-631-15506-5.

[347] A. Gavrichenkov. "Breaking HTTPS with BGP Hijacking." Black Hat. Briefings. (2015), URL: `https://www.blackhat.com/us-15/briefings.html`.

[348] A. W. Geiger, *Key findings about the online news landscape in America*, 11. Sep. 2019. URL: `https://pewrsr.ch/34CNdu3`.

[349] General Services Administration. "Federal Public Key Infrastructure Guide Introduction – Trust Stores." (Mai 2022), URL: `https://github.com/GSA/ficam-playbooks/blob/staging/_fpki/3_fpki_truststores.md` (besucht am 19. 05. 2022).

[350] R. (Gieben und O. Kolkman, *DNSSEC Operational Practices*, RFC 4641, Sep. 2006. URL: `https://www.rfc-editor.org/info/rfc4641`.

[351] Global Commission on Internet Governance, "Who Runs the Internet?: The Global Multi–stakeholder Model of Internet Governance," Centre for International Governance Innovation, Techn. Ber., 2017.

[352] M. Goodwin. "Revoking Intermediate Certificates: Introducing OneCRL," Mozilla. (3. März 2015), URL: `https://blog.mozilla.org/security/2015/03/03/revoking-intermediate-certificates-introducing-onecrl/` (besucht am 13. 07. 2023).

[353] Google. "Chrome Certificate Transparency Policy." (März 2022), URL: `https://github.com/GoogleChrome/CertificateTransparency/blob/main/ct_policy.md` (besucht am 21. 02. 2024).

[354] Google. "Certificate Transparency: Go Code." (2024), URL: `https://github.com/google/certificate-transparency-go`.

[355] Google. "Certificate Transparency: Known Logs." (2024), URL: `https://certificate.transparency.dev/google/`.

[356] P. Hallam-Baker und R. Stradling, *DNS Certification Authority Authorization (CAA) Resource Record*, RFC 6844, Jan. 2013. URL: `https://www.rfc-editor.org/info/rfc6844`.

[357] P. Hallam-Baker, R. Stradling und J. Hoffman-Andrews, *DNS Certification Authority Authorization (CAA) Resource Record*, RFC 8659, Nov. 2019. URL: `https://www.rfc-editor.org/info/rfc8659`.

[358] W. Hardaker und W. Kumari, "Security Considerations for RFC5011 Publishers," IETF, Internet-Draft – work in progress 13, Juli 2018.

[359] M. Hartmann, *Die Praxis des Vertrauens [The practice of Trust]*, Deutsch. Berlin: Suhrkamp Verlag, 2011, ISBN: 978-3-518-29594-6.

[360] J. Hodges, C. Jackson und A. Barth, *HTTP Strict Transport Security (HSTS)*, RFC 6797, Nov. 2012. URL: `https://www.rfc-editor.org/info/rfc6797`.

[361] P. E. Hoffman, *SMTP Service Extension for Secure SMTP over Transport Layer Security*, RFC 3207, Feb. 2002. URL: `https://www.rfc-editor.org/info/rfc3207`.

[362] P. E. Hoffman, *DNS Security Extensions (DNSSEC)*, RFC 9364, Feb. 2023. URL: `https://www.rfc-editor.org/info/rfc9364`.

[363] P. E. Hoffman und J. Schlyter, *The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA*, RFC 6698, Aug. 2012. URL: `https://www.rfc-editor.org/info/rfc6698`.

[364] P. E. Hoffman, A. Sullivan und K. Fujiwara, *DNS Terminology*, RFC 8499, Jan. 2019. URL: `https://www.rfc-editor.org/info/rfc8499`.

[365] R. Holz, J. Amann, A. Razaghpanah und N. Vallina-Rodriguez, *The Era of TLS 1.3: Measuring Deployment and Use with Active and Passive Methods*, Juli 2019.

[366] A. D. Householder, G. Wassermann, A. Manion und C. King. “The CERT® Guide to Coordinated Vulnerability Disclosure,” Carnegie Mellon University. (12. Dez. 2019), URL: `https://vuls.cert.org/confluence/display/CVD/The+CERT+Guide+to+Coordinated+Vulnerability+Disclosure` (besucht am 27. 06. 2023).

[367] S. Huque, *dane. Go library for DANE authentication*, GitHub. URL: `https://github.com/shuque/dane`.

[368] S. Huque, P. Aras, J. Dickinson, J. Včelák und D. Blacka, *Multi-Signer DNSSEC Models*, RFC 8901, Sep. 2020. URL: `https://www.rfc-editor.org/info/rfc8901`.

[369] ICANN Research, *TLD DNSSEC Report*, 21. Jan. 2022. URL: `http://stats.research.icann.org/dns/tld_report/`.

[370] IETF. “Web PKI OPS (wpkops).” (), URL: `https://datatracker.ietf.org/wg/wpkops/charter/`.

[371] Innovation Cell of the Department of Political and Peacebuilding Affairs, United Nations, *Diplomatic Pulse*, 2022. URL: `https://diplomaticpulse.org/`.

[372] International Organization for Standardization, “Information technology – Open Systems Interconnection – The Directory – Part 1: Overview of concepts, models and services,” International Organization for Standardization, Geneva, CH, Standard ISO/IEC 9594-1:2020, Nov. 2020.

[373] International Organization for Standardization, “The Directory – Part 8: Public-key and attribute certificate frameworks,” International Organization for Standardization, Geneva, Switzerland, Standard ISO/IEC 9594-8:2020, Dez. 2020.

[374] Internet Security Research Group (ISRG). “Combined Certificate Policy and Certification Practice Statement.” Version 5.1, ISRG. (16. Mai 2023), URL: `https://github.com/letsencrypt/cp-cps/blob/ad99251dde65bc7009f1fd10929373d75385c7ac/CP-CPS.md` (besucht am 08. 06. 2023).

[375] M. Just, “PKI Trust Models,” in *Encyclopedia of Cryptography and Security.* Boston, MA: Springer US, 2011, S. 936–939.

[376] B. S. Kaliski Jr, E. Osterweil und G. Wiley, *Extending DNSSEC trust chains to objects outside the DNS*, US Patent 9,705,851, Juli 2017.

[377] D. Kaminsky, “Black ops 2008: It’s the end of the cache as we know it,” Black Hat USA, Conference presentation, 2008.

[378] Kazakhtelecom JSC. “Kazakhtelecom JSC notifies on introduction of National security certificate from 1 January 2016,” Kazakhtelecom JSC. (30. Nov. 2015), URL: `https://web.archive.org/web/20151202203337/http://telecom.kz/en/news/view/18729` (besucht am 12. 07. 2023).

[379] O. Kolkman, “DNSSEC HOWTO, a tutorial in disguise,” NLnet Labs, Techn. Ber., Apr. 2010.

[380] O. Kolkman, “DNSSEC, DANE, and Diginotar,” APNIC 35 Conference, Conference presentation, 2013. URL: `https://conference.apnic.net/__data/assets/pdf_file/0005/58901/dnssec-diginotar-dane_1361864377.pdf`.

[381] O. Kolkman, M. Mekking und R. Gieben, *DNSSEC Operational Practices, Version 2*, RFC 6781, Dez. 2012. URL: `https://www.rfc-editor.org/info/rfc6781`.

[382] A. Kostopoulos, I. Papafili, C. Kalogiros, T. Levä, N. Zhang und D. Trossen, “A Tussle Analysis for Information-Centric Networking Architectures,” in *The Future Internet*, Ser. Lecture Notes in Computer Science, F. Álvarez, F. Cleary, P. Daras, J. Domingue, A. Galis, A. Garcia, A. Gavras, S. Karnourskos, S. Krco, M.-S. Li, V. Lotz, H. Müller, E. Salvadori, A.-M. Sassen, H. Schaffers, B. Stiller, G. Tselentis, P. Turkama und T. Zahariadis, Hrsg., Bd. 7281, Berlin, Heidelberg: Springer Berlin Heidelberg, Mai 2012, S. 6–17, ISBN: 9783642302411.

[383] E. Kovacs. “Symantec Revokes Wrongly Issued Certificates,” SecurityWeek. (23. Jan. 2017), URL: `https://www.securityweek.com/symantec-revokes-wrongly-issued-certificates/` (besucht am 29. 05. 2024).

[384] D. Kozlov. “Introducing: Backup Certificates,” Cloudflare. (März 2022), URL: `https://blog.cloudflare.com/introducing-backup-certificates/` (besucht am 23. 05. 2023).

[385] D. Kutscher, B. Kauffmann, J.-F. Peltier, P. Truong, B. Ahlgren, M. D. Ambrosio, E. Davies, A. E. Eriksson, S. Farrell, H. Flinck, C. Imbrenda, B. Kauffmann, G. Kunzmann, A. Lindgren, I. Marsh, L. Muscariello, H. Negrette, B. Ohlman, K.-A. Persson, P. Pöyhönen, J. Rajahalme, M. Shehada, M. Sosa, D. Staehle, O. Strandberg, J. Tuononen und V. Vercellone, “SAIL - Deliverable D.B.3:

Final NetInf Architecture," SAIL Consortium, Techn. Ber., Jan. 2013, S. 1–127. URL: https://sail-project.eu/wp-content/uploads/2013/01/SAIL-DB3-v1.1-final-public.pdf.

[386] D. Lagutin, K. Visala und S. Tarkoma, "Publish/subscribe for internet: PSIRP perspective," in *Towards the Future Internet: Emerging Trends from European Research*, Amsterdam, The Netherlands: IOS Press, 2010, S. 75–84, ISBN: 9781607505389.

[387] A. Langley. "Enhancing digital certificate security." (2013), URL: https://security.googleblog.com/2013/01/enhancing-digital-certificate-security.html.

[388] B. Laurie, A. Langley und E. Kasper, *Certificate Transparency*, RFC 6962, Juni 2013. URL: https://www.rfc-editor.org/info/rfc6962.

[389] B. Laurie, A. Langley, E. Kasper, E. Messeri und R. Stradling, *Certificate Transparency Version 2.0*, RFC 9162, Dez. 2021. URL: https://www.rfc-editor.org/info/rfc9162.

[390] B. Leiba, *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*, RFC 8174, Mai 2017. URL: https://www.rfc-editor.org/info/rfc8174.

[391] M. Lepinski und S. Kent, *An Infrastructure to Support Secure Internet Routing*, RFC 6480, Feb. 2012. URL: https://www.rfc-editor.org/info/rfc6480.

[392] LG Ansbach, *2 O 99/97 – Verletzung des Namensrechts durch Internet-Adresse – ansbach.de*, März 1997.

[393] LG Köln, *3 O 507-96 – Verwendung von Domain-Namen – Pulheim.de*, Dez. 1996.

[394] S. Lloyd und C. Adams, "Key Management," in *Encyclopedia of Cryptography and Security*. Boston, MA: Springer US, 2011, S. 683–688.

[395] N. Luhmann, *Trust and Power*. Chichester: Wiley, 1979, ISBN: 978-0-471-99758-0.

[396] N. Luhmann, *Vertrauen*. Konstanz: UTB, 2014, ISBN: 3-8252-4004-5.

[397] Majestic. "The Majestic Million." (Jan. 2021), URL: https://majestic.com/reports/majestic-million.

[398] Majestic SEO. "Welcome to Majestic Million." [Online; accessed 2023-05-22], Majestic. (28. Mai 2015), URL: https://blog.majestic.com/welcome-to-majestic-million/.

[399] D. Margolis, M. Risher, B. Ramakrishnan, A. Brotman und J. Jones, *SMTP MTA Strict Transport Security (MTA-STS)*, RFC 8461, Sep. 2018. URL: https://www.rfc-editor.org/info/rfc8461.

[400] S. P. Marsh, "Formalising Trust as a Computational Concept," Diss., Apr. 1994.

[401] MDN contributers. "Certificate Transparency," Mozilla. (26. Feb. 2023), URL: https://developer.mozilla.org/en-US/docs/Web/Security/Certificate_Transparency (besucht am 23. 05. 2023).

[402] MDN contributers. "eTLD," Mozilla. (16. März 2023), URL: https://developer.mozilla.org/en-US/docs/Glossary/eTLD (besucht am 23. 05. 2023).

[403] U. Meissen, S. Pfennigschmidt, M. Hardt und D. Faust, “KATWARN – A Microservice-Based Architecture for Distributed, Flexible and Robust Warning Systems,” in *Progress in IS*, Springer International Publishing, 2018, S. 213–225.

[404] P. Mends, S. Diamantopoulos, C.-A. Sarros, V. Tsaoussidis, I. Psaras, S. Rene, J. Crowcroft, A. Lertsinsrubtavee, C. Molina-Jimenez, S. Perez Sanchez, I. Sedano und S. Rute, “UMOBILE - Deliverable D3.4: UMOBILE ICN layer abstraction final specification,” UMOBILE Consortium, Techn. Ber., Juli 2017, S. 1–67.

[405] J. Menn. “Mysterious company with government ties plays key internet role,” The Washington Post. (8. Nov. 2022), URL: `https://www.washingtonpost.com/technology/2022/11/08/trustcor-internet-addresses-government-connections/` (besucht am 29. 05. 2024).

[406] S. A. Meyerowitz, “Trust Management,” in *Intrusion Detection Networks*, 5, Bd. 125, Auerbach Publications, Nov. 2013, S. 51–72.

[407] Microsoft. “Configure trusted roots and disallowed certificates in Windows,” Microsoft. (4. Mai 2023), URL: `https://learn.microsoft.com/en-us/windows-server/identity/ad-cs/configure-trusted-roots-disallowed-certificates` (besucht am 13. 07. 2023).

[408] D. L. Mills, *Internet name domains*, RFC 799, Sep. 1981. URL: `https://www.rfc-editor.org/info/rfc799`.

[409] P. Mockapetris, *Domain names: Concepts and facilities*, RFC 882, Nov. 1983. URL: `https://www.rfc-editor.org/info/rfc882`.

[410] P. Mockapetris, *Domain names - concepts and facilities*, RFC 1034, Nov. 1987. URL: `https://www.rfc-editor.org/info/rfc1034`.

[411] T. Mononen, T. Kause, S. Farrell und D. C. Adams, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*, RFC 4210, Sep. 2005. URL: `https://www.rfc-editor.org/info/rfc4210`.

[412] K. Moriarty und S. Farrell, *Deprecating TLS 1.0 and TLS 1.1*, RFC 8996, März 2021. URL: `https://www.rfc-editor.org/info/rfc8996`.

[413] S. Morris, J. Stenstam, J. Dickinson und M. Mekking, *DNSSEC Key Rollover Timing Considerations*, RFC 7583, Okt. 2015. URL: `https://www.rfc-editor.org/info/rfc7583`.

[414] Mozilla Foundation. “Mozilla PKI Policy.” [Accessed 2023-05-25], Mozilla. (1. Feb. 2023), URL: `https://github.com/mozilla/pkipolicy`.

[415] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” Bitcoin, Techn. Ber., 2019.

[416] Named Data Networking Project. “NDN Certificate Format Version 2.0,” NDN Project Team. (Okt. 2021), URL: `https://github.com/named-data/ndn-cxx/blob/4999b2eddd7da10bc934a27f2f151e37f2095765/docs/specs/certificate.rst` (besucht am 03. 05. 2022).

[417] Named Data Networking Project. "NDN Packet Format Specification 0.3 – KeyLocator." Online; accessed 2022-05-03, NDN Project Team. (Dez. 2021), URL: `https://github.com/named-data/NDN-packet-spec/blob/acaccafd710a306df405e4b40302080d729b4ade/signature.rst#keylocator` (besucht am 03. 05. 2022).

[418] Named Data Networking Project. "NDN Packet Format Specification 0.3 – NDN Name Format," NDN Project Team. (Dez. 2021), URL: `https://github.com/named-data/NDN-packet-spec/blob/23e340cb138afa95ee228dffb8c50870eea26633/name.rst#ndn-name-format` (besucht am 03. 05. 2022).

[419] National Telecommunications and Information Administration, *Improvement of Technical Management of Internet Names and Addresses*, Feb. 1998.

[420] NDN Consortium. "Named Data Networking: Motivation & Details." (2019), URL: `https://web.archive.org/web/20190408131300/https://named-data.net/project/archoverview/` (besucht am 08. 04. 2019).

[421] NDN Consortium, *Named Data Networking: Motivation & Details*. URL: `https://named-data.net/project/archoverview/` (besucht am 22. 06. 2020).

[422] NDN Project Team. "NFD – Control Command." (28. Juni 2020), URL: `https://redmine.named-data.net/projects/nfd/wiki/ControlCommand/52` (besucht am 17. 05. 2022).

[423] NDN Project Team, "NDN Technical Memo: Naming Conventions," Techn. Ber. NDN-0022, Okt. 2021.

[424] Neustar. "Frequently Asked Questions about the .US Domain." (2020), URL: `https://www.about.us/faqs` (besucht am 13. 04. 2020).

[425] NIST, *Estimating USG IPv6 & DNSSEC External Service Deployment Status*, 2. Feb. 2022. URL: `https://fedv6-deployment.antd.nist.gov/cgi-bin/generate-gov`.

[426] Nonprofit Tech for Good, *2018 Global Trends in Giving Report*, 2018. URL: `https://givingreport.ngo/`.

[427] M. Nottingham, *Well-Known Uniform Resource Identifiers (URIs)*, RFC 8615, Mai 2019. URL: `https://www.rfc-editor.org/info/rfc8615`.

[428] E. Osterweil. "SecSpider." (), URL: `https://secspider.net`.

[429] W. Palant. "Weakening TLS protection, South Korean style." (26. Feb. 2023), URL: `https://palant.info/2023/02/06/weakening-tls-protection-south-korean-style/` (besucht am 06. 06. 2023).

[430] J. M. Perkel. "Why Jupyter is data scientists' computational notebook of choice." (30. Okt. 2018), URL: `https://www.nature.com/articles/d41586-018-07196-1` (besucht am 23. 10. 2024).

[431] H. Perl, S. Fahl und M. Smith, "You Won't Be Needing These Any More: On Removing Unused Certificates from Trust Stores," in *Financial Cryptography and Data Security*, Berlin/Heidelberg, Germany: Springer Berlin Heidelberg, 2014, S. 307–315.

[432] J. Postel und J. Reynolds, *Domain requirements*, RFC 920, Okt. 1984. URL: https://www.rfc-editor.org/info/rfc920.

[433] W. W. Powell, "Trust-Based Forms of Governance," in *Trust in Organizations: Frontiers of Theory and Research*, Thousand Oaks: SAGE Publications, Inc., 1996, S. 51–67. URL: https://sk.sagepub.com/books/trust-in-organizations/n4.xml.

[434] M. Prince. "How the Consumer Product Safety Commission is (Inadvertently) Behind the Internet's Largest DDoS Attacks." (2016), URL: https://blog.cloudflare.com/how-the-consumer-product-safety-commission-is-inadvertently-behind-the-internets-largest-ddos-attacks/.

[435] R. D. Putnam, "Bowling Alone: America's Declining Social Capital," in *Culture and Politics: A Reader*, L. Crothers und C. Lockhart, Hrsg., New York: Palgrave Macmillan US, 2000, S. 223–234, ISBN: 978-1-349-62965-7.

[436] Radio Farda. "Iran Says Its Intranet Almost Ready To Shield Country From 'Harmful' Internet." (Mai 2019), (besucht am 24. 06. 2020).

[437] Rapid7. "An Introduction to Project Sonar," Rapid7. (2023), URL: https://www.rapid7.com/research/project-sonar/ (besucht am 03. 07. 2023).

[438] E. Rescorla, *HTTP Over TLS*, RFC 2818, Mai 2000. URL: https://www.rfc-editor.org/info/rfc2818.

[439] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, RFC 8446, Aug. 2018. URL: https://www.rfc-editor.org/info/rfc8446.

[440] E. Rescorla und T. Dierks, *The Transport Layer Security (TLS) Protocol Version 1.2*, RFC 5246, Aug. 2008. URL: https://www.rfc-editor.org/info/rfc5246.

[441] E. Rescorla, H. Tschofenig und N. Modadugu, *The Datagram Transport Layer Security (DTLS) Protocol Version 1.3*, RFC 9147, Apr. 2022. URL: https://www.rfc-editor.org/info/rfc9147.

[442] J. Riegelsberger, "Trust in Mediated Interactions," Diss., University of London, Juni 2005.

[443] R. L. Rivest und B. Lampson, *SDSI – A Simple Distributed Security Infrastructure*, Working Document, Okt. 1996.

[444] S. Rose, M. Larson, D. Massey, R. Austein und R. Arends, *DNS Security Introduction and Requirements*, RFC 4033, März 2005. URL: https://www.rfc-editor.org/info/rfc4033.

[445] S. Rose, M. Larson, D. Massey, R. Austein und R. Arends, *Protocol Modifications for the DNS Security Extensions*, RFC 4035, März 2005. URL: https://www.rfc-editor.org/info/rfc4035.

[446] S. Rose, M. Larson, D. Massey, R. Austein und R. Arends, *Resource Records for the DNS Security Extensions*, RFC 4034, März 2005. URL: https://www.rfc-editor.org/info/rfc4034.

[447] J. H. Saltzer, "Naming and Binding of Objects," in *Operating Systems - An Advanced Course*, Berlin/Heidelberg, Germany: Springer, 1978, S. 99–208.

[448] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin und D. C. Adams, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*, RFC 6960, Juni 2013. URL: `https://www.rfc-editor.org/info/rfc6960`.

[449] J. Seedorf, M. Arumaithurai, A. Tagami, K. Ramakrishnan und N. Blefari-Melazzi, "Research Directions for Using ICN in Disaster Scenarios," Internet Engineering Task Force, Internet-Draft draft-irtf-icnrg-disaster-07, Juni 2019, Work in Progress, 19 S.

[450] G. Selander, J. P. Mattsson, F. Palombini und L. Seitz, *Object Security for Constrained RESTful Environments (OSCORE)*, RFC 8613, Juli 2019. URL: `https://www.rfc-editor.org/info/rfc8613`.

[451] J. Sermersheim, *Tentative proposal for a Unified User Level Protocol*, RFC 4511, Feb. 1973. URL: `https://www.rfc-editor.org/info/rfc451`.

[452] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," in *Advances in Cryptology*, G. R. Blakley und D. Chaum, Hrsg., Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, S. 47–53.

[453] Z. Shelby, K. Hartke und C. Bormann, *The Constrained Application Protocol (CoAP)*, RFC 7252, Juni 2014. URL: `https://www.rfc-editor.org/info/rfc7252`.

[454] M. Shore, R. Barnes, S. Huque und W. Toorop, "A DANE Record and DNSSEC Authentication Chain Extension for TLS," Internet Engineering Task Force, Internet-Draft draft-ietf-tls-dnssec-chain-extension-07, März 2018, Work in Progress, 24 S.

[455] C. Shu und J. Schieber, *Facebook, Reddit, Google, LinkedIn, Microsoft, Twitter and YouTube issue joint statement on misinformation*, 17. März 2020. URL: `https://tcrn.ch/2xJXrg8`.

[456] M. Shuler. "ca-certificates README.debian," The Debian Project. (15. Dez. 2011), URL: `https://salsa.debian.org/debian/ca-certificates/-/blob/6b03bfc73df7677270fd98893e089ad785400948/debian/README.Debian` (besucht am 13. 07. 2023).

[457] G. Simmel, *The Sociology of Georg Simmel*. Glencoe: Free Press, 1950, ISBN: 978-0-02-928920-4.

[458] K. R. Sollins, "Recursively invoking Linnaeus: A Taxonomy for Naming Systems," MIT, Techn. Ber. MIT-CSAIL-TR-2008-064, Okt. 2008.

[459] R. C. Solomon und F. Flores, *Building Trust: In Business, Politics, Relationships, and Life*. Oxford: Oxford University Press, 2003, ISBN: 9780198029243.

[460] A. Souppouris. "Turkish citizens use Google to fight Twitter ban." (März 2014), URL: `https://www.theverge.com/2014/3/21/5532522/turkey-twitter-ban-google-dns-workaround` (besucht am 29. 06. 2020).

[461] Spamhaus Project, *The World's Most Abused TLDs*, 2. Feb. 2022. URL: `https://www.spamhaus.org/statistics/tlds/`.

[462] StatCounter. “Market share of leading internet browsers in the United States and worldwide as of August 2022,” Statista. (Aug. 2022), URL: https://www.statista.com/statistics/276738/worldwide-and-us-market-share-of-leading-internet-browsers/ (besucht am 23. 05. 2023).

[463] Statista, *Nachrichten*, 2021. URL: https://de.statista.com/statistik/studie/id/7061/dokument/nachrichten-statista-dossier/.

[464] M. StJohns, *Automated Updates of DNS Security (DNSSEC) Trust Anchors*, RFC 5011, Sep. 2007. URL: https://www.rfc-editor.org/info/rfc5011.

[465] J. Styler. “The top 25 most expensive domain names.” (Juni 2019), (besucht am 06. 07. 2020).

[466] Z.-S. Su, *Distributed system for Internet name service*, RFC 830, Okt. 1982. URL: https://www.rfc-editor.org/info/rfc830.

[467] Z.-S. Su und J. Postel, *The Domain Naming Convention for Internet User Applications*, RFC 819, Aug. 1982. URL: https://www.rfc-editor.org/info/rfc819.

[468] S. Tarkoma, M. Ain und K. Visala, “The publish/subscribe internet routing paradigm (PSIRP): Designing the future internet architecture,” in *Towards the Future Internet: A European Research Perspective*, Amsterdam, The Netherlands: IOS Press, 2009, S. 102–111, ISBN: 9781607500070.

[469] D. Tatang, R. Flume und T. Holz, “Extended Abstract: A First Large-Scale Analysis on Usage of MTA-STS,” in *Detection of Intrusions and Malware, and Vulnerability Assessment*, Cham, Switzerland: Springer International Publishing, 2021, S. 361–370.

[470] Telecommunication Standardaziation of ITU, “The Directory: Public-key and attribute certificate frameworks,” International Telecommunication Union, Techn. Ber., Okt. 2021.

[471] The Linux Foundation. “Common CA Database.” (2024), URL: https://www.ccadb.org/resources.

[472] M. Thomson und S. Turner, *Using TLS to Secure QUIC*, RFC 9001, Mai 2021. URL: https://www.rfc-editor.org/info/rfc9001.

[473] H. C. A. van Tilborg und S. Jajodia, *Encyclopedia of Cryptography and Security* (Encyclopedia of Cryptography and Security). Springer US, 2011, ISBN: 9781441959065.

[474] S. Trapp, M. Wählisch und J. Schiller, *Bridge the Gap: Measuring and Analyzing Technical Data for Social Trust between Smartphones*, Mai 2012.

[475] N. Trenaman. “Lessons Learned on Improving RPKI.” (2020), URL: https://labs.ripe.net/Members/nathalie_nathalie/lessons-learned-on-improving-rpki.

[476] D. Trossen, G. Parisis, K. Visala, B. Gajic, J. Riihijarvi, P. Flegkas, P. Sarolahti, P. Jokela, X. Vasilakos, C. Tsilopoulos und S. Arainfar, “PURSUIT - Deliverable D2.2: Conceptual Architecture: Principles, patterns and sub-components descriptions,” PURSUIT Consortium, Techn. Ber., Mai 2011, S. 1–50. URL: http://www.fp7-pursuit.eu/PursuitWeb/?page_id=158.

[477] D. Trossen, B. Tagger, G. Parisis, M. Skjegstad, N. Fotiou, X. Vasilakos, C. Stais, Y. Thomas, M. Al-Naday, M. Reed, V. Vassilakis, A. Bontozoglou, B. Gajic, P. Sarolahti, P. Flegkas und V. Sourlas, “PURSUIT - Deliverable D2.5: Final Updated Architecture,” PURSUIT Consortium, Techn. Ber., März 2013. URL: `http://www.fp7-pursuit.eu/PursuitWeb/?page_id=158`.

[478] US CERT Vulnerability Notes Database. “Vulnerability Note VU#800113: Multiple DNS implementations vulnerable to cache poisoning.” (Juli 2008).

[479] S. Weiler und J. Stenstam, *Minimally Covering NSEC Records and DNSSEC On-line Signing*, RFC 4470, Apr. 2006.

[480] J. Westfall, *Common Alerting Protocol Version 1.2*, Juli 2010.

[481] M. E. Whitman und H. J. Mattord, *Principles of Information Security*. Boston: Course Technology, 2009, ISBN: 9781423901778.

[482] WHO, *Novel Coronavirus (2019-nCoV) Situation Report - 13*, 2. Feb. 2020. URL: `https://apps.who.int/iris/handle/10665/330778`.

[483] M. Williams, M. Tüxen und R. Seggelmann, *Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension*, RFC 6520, Feb. 2012. URL: `https://www.rfc-editor.org/info/rfc6520`.

[484] B. Wissingh, C. A. Wood, A. Afanasyev, L. Zhang, D. R. Oran und C. Tschudin, “Information-Centric Networking (ICN): CCN and NDN Terminology,” Internet Engineering Task Force, Internet-Draft draft-irtf-icnrg-terminology-04, Juni 2019, Work in Progress, 18 S.

[485] C. Wright. “Understanding Kaminsky’s DNS Bug.” (Juli 2008), URL: `https://www.linuxjournal.com/content/understanding-kaminskys-dns-bug` (besucht am 23. 06. 2024).

[486] Y. Yu, “Public Key Management in Named Data Networking,” UCLA, Techn. Ber. NDN-0029, Apr. 2015.

[487] “ZCrypto. Liberal Go TLS + X.509 Library for Research.” (), URL: `https://github.com/zmap/zcrypto`.

[488] M. Zey, *Rational Choice Theory and Organizational Theory: A Critique*. London: SAGE Publications, 1997, ISBN: 978-1-4522-5496-8.

[489] H. Zhang, “NDNFit: An Open mHealth Application Built on Named Data Networking,” Diss., University of California Los Angeles, 2018.

[490] L. Zhang, D. Estrin, B. Jeffrey, V. Jacobso, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, k. claffy kc, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley und E. Yeh, “Named Data Networking (NDN) Project,” Techn. Ber. NDN-0001, Okt. 2010.

[491] L. Zhu, D. Wessels, A. Mankin und J. Heidemann, “Measuring DANE TLSA Deployment,” in *Traffic Monitoring and Analysis*, Springer International Publishing, 2015, S. 219–232.

[492] P. R. Zimmermann, *The Official PGP User's Guide*. Cambridge, MA, USA: MIT Press, 1995, ISBN: 0-262-74017-6.