sensors

MDPI

Article

A Hybrid Secure Scheme for Wireless Sensor Networks against Timing Attacks Using Continuous-Time Markov Chain and Queueing Model

Tianhui Meng [1,‡], Xiaofan Li [2,†,‡], Sha Zhang [2,*,†,‡] and Yubin Zhao [3]

1 Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin 14195, Germany; tianhui.meng@fu-berlin.de
2 Shenzhen Institute of Radio Testing & Tech., Shenzhen 518000, China; lixiaofan@srtc.org.cn
3 Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China; zhaoyb@siat.ac.cn
* Correspondence: zhangsha@srtc.org.cn; Tel.: +86-755-8639-2359
† Current address: The State Monitoring Center and Testing Center, Beijing 100876, China
‡ These authors contributed equally to this work.

Academic Editor: Rongxing Lu
Received: 14 April 2016; Accepted: 5 September 2016; Published: 28 September 2016

Abstract: Wireless sensor networks (WSNs) have recently gained popularity for a wide spectrum of applications. Monitoring tasks can be performed in various environments. This may be beneficial in many scenarios, but it certainly exhibits new challenges in terms of security due to increased data transmission over the wireless channel with potentially unknown threats. Among possible security issues are timing attacks, which are not prevented by traditional cryptographic security. Moreover, the limited energy and memory resources prohibit the use of complex security mechanisms in such systems. Therefore, balancing between security and the associated energy consumption becomes a crucial challenge. This paper proposes a secure scheme for WSNs while maintaining the requirement of the security-performance tradeoff. In order to proceed to a quantitative treatment of this problem, a hybrid continuous-time Markov chain (CTMC) and queueing model are put forward, and the tradeoff analysis of the security and performance attributes is carried out. By extending and transforming this model, the mean time to security attributes failure is evaluated. Through tradeoff analysis, we show that our scheme can enhance the security of WSNs, and the optimal rekeying rate of the performance and security tradeoff can be obtained.

Keywords: Markov chain; queueing model; side-channel attacks; random padding

1. Introduction

Wireless sensor networks (WSNs), which are enabled by the developments of wireless communications, micro-electro-mechanical systems (MEMS) technology and digital electronics, have found their way into a wide variety of applications and systems with vastly varying requirements and characteristics. In accordance with the vision of [1], WSNs are slowly becoming an integral part of our daily lives. The sensing technology combined with processing power and wireless communication makes it lucrative for being exploited in abundance in the future [2]. Recently, considerable amounts of research efforts have enabled the actual implementation and deployment of sensor networks tailored to the unique requirements of certain sensing and monitoring applications [3–5].

Since these networks are usually deployed in remote areas and left unattended, they should be defended by security mechanisms against attacks, such as physical tampering, node capture,

eavesdropping, denial of service, etc. Unfortunately, traditional security mechanisms with high overhead are not feasible for the resource-constrained sensor nodes [6]. An important issue of WSNs is how to preserve sensitive measurements taken from everyday life where data privacy is an essential aspect. In many scenarios, the confidentiality of propagated data can be considered critical. For instance, in a healthcare system, data from sensors might measure patients' health information, such as heartbeat and blood pressure details. Data privacy can be simply defined as a process in which private data can be overheard and decrypted by adversaries or other trusted participating sensor nodes [7], but it can still provide a mechanism that prevents them from recovering sensitive information, i.e., control disclosure of any information about the data.

Securing cryptographic implementations in WSNs against side-channel attacks is one of the most important challenges [8], which is not covered by traditional notions of cryptographic security [9]. Among side-channel attacks, timing attacks, whose remote feasibility has been proven by Brumley and Boneh [10,11], make a practical threat against sensor network systems. Energy is an essential resource of WSNs, so the implementations of cryptographic algorithms in such systems often perform computations in non-constant time due to performance optimization. The timing attack involves algorithms that have non-constant execution time, and this can potentially leak secret information. As represented in [12], the operating system running on the sensor nodes is event-driven and extremely optimized in terms of memory consumption, which suggests that the timing side-channel is practical in WSNs.

In this paper, we propose a secure scheme to defend WSNs from timing attacks. The rekeying process with an optimal rate is introduced into the WSNs, and random paddings are interposed in the processing time to mitigate the information leakage of the timing side-channel. Our contributions are now summarized as follows:

- In order to proceed to a quantitative treatment of the performance-security tradeoff of WSNs, we propose a hybrid continuous-time Markov chain (CTMC) and queueing model for the system under the specific threat of timing attacks.
- We have shown the measures' formulation, including both security and performance attributes, and the optimal tradeoff between the two.
- Experimental evaluations demonstrate the effectiveness of the random padding countermeasure against timing attacks and the tradeoff improvement one can obtain from the proposed scheme.

The remainder of this paper is structured as follows. In Section 2, we overview the current state-of-art related works. Section 3 shows the considered multi-gateway clustered topology and rekeying management method as the system architecture. A hybrid model that employs the hybrid CTMC and queueing model for a WSN under timing attacks is proposed in Section 4. In this section, we describe the state-transition model and the scheduling method. The system metrics on which the evaluation is based are addressed in Section 5. Section 6 gives the theoretical model analysis of the security and performance attributes using quantitative assessment methods. Experimental evaluation of the general WSN system is presented in Section 7. Finally, the paper is concluded in Section 8.

2. Related Work

A WSN is a collection of nodes organized into a cooperative network [13]. Applications of WSNs are growing rapidly, which range from indoor deployment scenarios in offices to outdoor deployments in adversarial territory. However, due to their deployment environments and resource limitations, these networks are vulnerable to many security threats that can deteriorate the performance [14].

These attacks are often classified as external attacks and internal attacks [15]. In an external attack, the attacker node is not an authorized participant of the sensor network. Adversaries can severely limit the usefulness of a WSN through denial-of-service (DoS) attacks [16,17]. An attacker may simply disrupt the network operation by broadcasting a high-energy signal. If the transmission is powerful enough, the entire system communication could be jammed. The standard defense against jamming

involves various forms of spread-spectrum or frequency hopping communication [18]. However, cryptographically-secure spread-spectrum radios are not commercially available. In addition, this defense is not sufficiently secure against node capture attacks and cryptographic key extracting [14].

Node compromise is the major problem in sensor networks that leads to internal attacks. In contrast to disabled nodes, compromised nodes actively seek to disrupt or paralyze the network [19]. Routing and data forwarding are essential services for enabling communication in sensor networks. Unfortunately, many routing protocols suffer from security vulnerabilities [20]. The simplest attacks involve injecting malicious routing information into the network, resulting in routing inconsistencies.

Another area that has received a great deal of attention in WSN security is key management. A taxonomy of key management protocols in WSNs is described in [21].

Non-constant execution time can be caused by conditional branching and various optimization techniques. If such operations involve secret parameters, these timing variations can leak some information, and a careful statistical analysis could even lead to the total recovery of these secret keys. Because timing attacks gain secret information from the response times and rather than brute force attacks or theoretical weaknesses in the algorithms, they are a real threat to WSNs as an internal attack. However, this threat is not covered by traditional notions of cryptographic security [8]. It was commonly believed that timing attacks can be directed only towards smart cards or affect the inter-process locally, but more recent research reveals that remote timing attacks are also possible and should be taken into consideration [10,11].

The interposition of random delays in the cryptographic algorithm execution flow is a simple, but rather effective countermeasure against side-channel and fault attacks by mitigating the information leakage. Random delays are easily deployed even if the source code of the application is not at hand. They are widely used for the protection of cryptographic implementations in embedded devices [22]. To date, based on random delay insertion, a processor architecture resistant to side-channel attacks is proposed in [23] using a combination of randomized scheduling, randomized instruction insertion and randomized pipeline-delay. The researchers in [24] present a design and hardware implementation of asynchronous AES with random noise injection for improved side-channel attack resistance.

3. Regarded System Architecture

We adopt the sensor network model in Figure 1, which is proposed by Younis et al. [25]. In this model, the network consists of a large number of sensor nodes distributed over an area of interest. There is at least one command node in charge of the system. The model also introduces some super-nodes, called gateway nodes, in addition to the sensor nodes. The gateway nodes have more computational, energy, memory and communication resources. They act as a gateway between sensor nodes and the end user, as they typically forward data from the WSN on to a command node [26]. As depicted in Figure 1, the gateways partition the sensors into different distinct clusters. Each cluster is composed of a gateway node and a set of sensor nodes. The operations within a cluster are independent of each other. The cluster sensor nodes transmit the gathered and integrated data to the gateway node of their cluster.

It is assumed that the gateway nodes are deployed in safe places and cannot be captured by the adversary; while the other nodes in the cluster may be captured and compromised. For the key management, we adopt the scheme proposed by Eschenauer and Gligor [27]. The management scheme triggers the rekeying phase regularly to handle the timing attacks from inside or outside attackers, and our goal is to indicate the optimal rekeying rate. In our approach, the sensor nodes use a symmetric key mechanism. A cluster key is used for the encryption and decryption operations of all data between the sensors and the gateway node. Before being deployed, the cluster key is programmed into the memory of the sensors. Sensor and gateway keys are stored in the flash RAM and, hence, can be renewed in the rekeying phase. We assume that the time to perform a rekeying operation is measured based on a generic Diffie–Hellman protocol (GDH) [28,29]. At the same time, random delays are

padded for each gateway node response message to mitigate the information leakage of the timing side-channel.

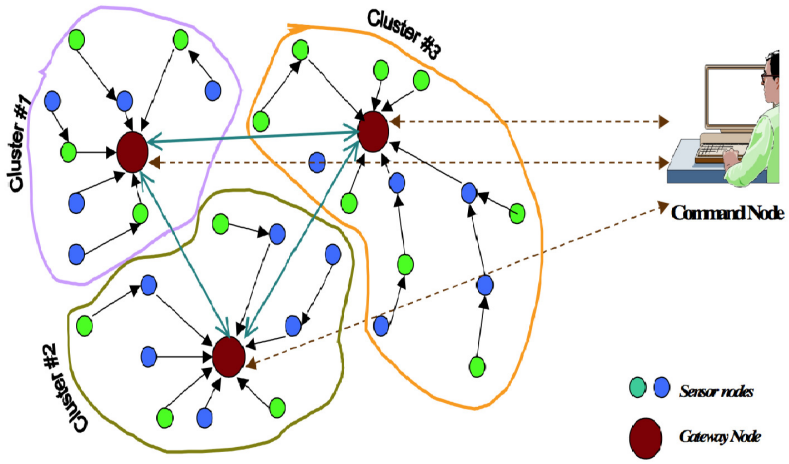

Figure 1. Architecture of a multi-gateway clustered sensor network.

4. The System Model

In order to analyze the performance and security attributes of WSNs under the threat of timing attacks quantitatively, we need to combine the behaviors of an attacker who is trying to gain the sensitive information in conjunction with the countermeasures taken by the system. Therefore, we developed a hybrid CTMC and queueing model, shown in Figure 2, that takes into account the behavior of both sides. The goal is to identify optimal system settings to maximize the security and performance tradeoff metric described in Section 5.3.

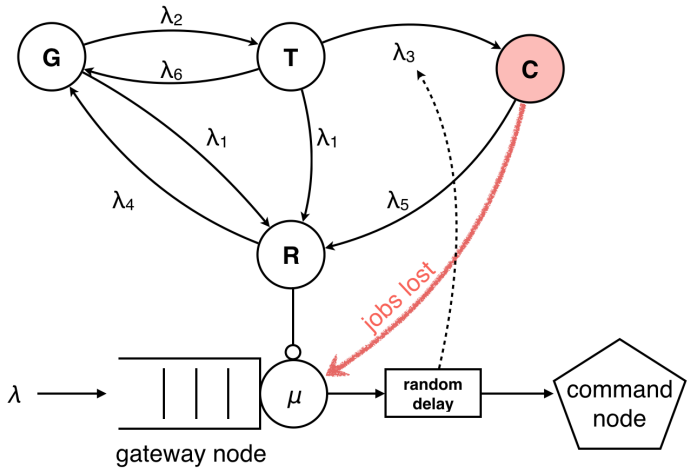

Figure 2. State transition diagram for a WSN under timing attacks.

The state transition model represents the system behavior under a specific attack and given system configuration that depends on the actual security requirements; whereas, the queueing model is proposed to exhibit the data transmission between sensors and gateway nodes. A coming job joins a queue before it is processed and forwarded to a command node. The states and parameters of the CTMC state transition model are summarized here:

- G Good state in which the WSN works properly

- T Timing attack state
- C Compromised state after the attacker knows the secret of the system
- R Rekeying state in which the system renews its cluster key
- λ_1 rate at which the system launches the rekeying process in state G and state T
- λ_2 rate at which an attacker triggers a timing attack on the system
- λ_3 rate at which a timing attack succeeds to break the cluster key
- λ_4 rate at which the WSN is brought back to the good state by the rekeying process
- λ_5 rate at which the system launches the rekeying process in the compromised state C
- λ_6 rate at which the attacker successfully breaks the key, while failing to access legitimately-authorized information.

4.1. Attack Model

The WSN is vulnerable to timing attacks in which the attacker in the worst case will eventually obtain the secret cluster key. In order to analyze the security properties of the proposed scheme, it is necessary to consider the actions undertaken by an adversary, as well as the system's response to the adversary.

In timing attacks on a WSN, an attacker first compromises a sensor node or mimics a sensor node to send data to the gateway node. In addition, the attacker records each response time for a certain query and tries to find clues to the cluster key by comparing differences in the response times. On the condition that the attacker breaks the cluster key from the timing results, he or she can attack the secrecy and authenticity of communication channels. Obviously, this requires the attacker to spend effort, where we use time to represent the attacking effort. This time is modeled as a random variable, which may follow one distribution function depending on the nature of the attack. Deterministic, exponential, hypo-exponential, hyper-exponential, gamma, Weibull and log-logistic, etc., are some of the distribution functions that make sense in the context of security analysis [30]; while we use an exponential distribution to model the attacker arrival time and the time a timing attack takes.

4.2. Security and Performance Model

In the presence of inside and outside attacks, our secure scheme for a WSN is concerned primarily with the evaluation of the three attributes: confidentiality (sensitive information is not disclosed to unauthorized member), integrity (data and programs are modified only in a authorized manner) and availability (readiness for usage).

The upper part of Figure 2 shows a CTMC model representing the states of the system. After initialization, the system starts to operate properly in the good state G. The system is under the specific threat of timing attacks conducted by random attackers. We describe the events that trigger transitions among states in terms of transition rates. It is assumed that there is only one attacker in the system at a time. If an attack happens, the system is brought to the timing attack state T at rate λ_2. In this state, the attacker tries to break the cluster key by taking time observations. Therefore, while the system is in state T, the attacker is not yet able to access confidential information.

It takes a certain time to perform the timing attack after which the attacker may know the cluster key, and the system moves to the compromised state C at rate λ_3. Consequently, λ_3^{-1} is the mean time a timing attack takes. There is a possibility indicated by the arc λ_6 that the attacker successfully breaks the key, while he or she fails at accessing the data or just fails to conduct a successful timing attack. If the attacker succeeds in determining the cluster key through time measurements, confidential data will be disclosed, which is assumed to incur a high cost. This can only happen if the system is in the compromised state C, and we call the incident of entering the compromised state a security failure. In this state, all of the data transited between sensors and gateway nodes are not secure any more; therefore, they must be repeated and do not contribute to the throughput. The lost jobs are represented by the red arc in Figure 2.

Renewing the cluster encryption key can prevent or interrupt a timing attack. The arcs from other states to state R represent these operations. The rekeying rate is the parameter one can tune at the command node. It indicates how often the system launches the rekeying process. The rate λ_1 is the rekeying rate when the system is in the good state G or in the timing attack state T. The considered WSN has intrusion detection mechanisms running that can find indicators of compromised behavior, in which case the system will trigger the rekeying process more frequently. The intrusion detection mechanism does not trigger the rekeying immediately because of the rekeying cost to the system. Therefore, in the compromised state C, we assume that the rekeying process is triggered at a different rate, $\lambda_5 = n\lambda_1$. The parameter n is called the coefficient of rekeying in the compromised state because it represent the relationship between the rekeying rate in the good state and the rekeying rate in the compromised state. All of these three paths transfer the WSN cluster to the rekeying state R from which it will finally return to the initial state G. The challenge is to find an optimal value for the rekeying interval. The rekeying should in the optimal case happen before or soon after the system enters the compromised state.

In the rekeying state, the sensors do not transit any data to the gateway, and some jobs will be lost in this state. As a result, the system throughput is degraded. The rekeying process will bring the system back to the initial state G at rate λ_4. The mean time to perform the rekeying process is λ_4^{-1}.

The lower part of Figure 2 shows the queueing model proposed to exhibit the performance attribute of the WSN. The data collected and preprocessed by sensors are transmitted to the gateway node before forwarded to a command node, which is represented by a queue. The parameter λ indicates the arrival rate, and μ presents the service rate of the gateway node, including the data transmission time. When random delays are interposed in the gateway node, the attacker needs more samples to average and successfully guess the cluster key. Therefore, it takes more time for him or her to conduct the timing attack. As a result, the attacking rate λ_3 decreases as this mitigation method is taken, which is represented by the dashed arc in the figure.

5. Metrics

After defining the model and its parameters, we establish the measures we want to investigate in this section. We present security and performance metrics, respectively, which are illustrated in Table 1. Since the sensor nodes carry limited battery resources, WSN applications and communication protocols focus primarily on power conservation. As a result, communication is an important factor, since data transmission mainly influences energy consumption.

Table 1. Table of Notation.

π_i	$\triangleq$	The steady-state probability that the continuous-time Markov process is in state i, $i \in \{R, C\}$
Θ	$\triangleq$	System cost metric
Λ	$\triangleq$	Confidentiality metric
X	$\triangleq$	System throughput
T	$\triangleq$	Tradeoff metric

5.1. Security Metrics

As security measures, we define system cost and confidentiality, which are functions of the steady-state probabilities of the CTMC model. The steady-state probabilities π_i may be interpreted as the proportion of time that the CTMC spends in state i, where $i \in \{R, G, T, C\}$. The WSN suffers from cost in two states, the compromised state C and the rekeying state R. The system loses sensitive information in the compromised state, and cost is also incurred when the clustered sensor network deploys a rekeying process due to the extra communication. The rekeying cost and the data disclosure cost are both interpreted as the proportion of system life time, that is the steady-state probability of the

CTMC. We define a weight w and its complement $1 - w$ for the two kinds of cost. We use normalized weights for simplicity. Mathematically, the system cost is computed with:

$$\begin{aligned} \Theta &= C_{rekeying} + C_{disclosure} \\ &= w\pi_R + (1-w)\pi_C \end{aligned} \tag{1}$$

where $\pi_i, i \in \{R, C\}$ denotes the steady-state probability that the continuous-time Markov process is in state i. $0 \leq w \leq 1$ is the weighting parameter used to share the relative importance between the loss of sensitive information and the power consumption needed to rekey regularly.

If a timing attack on the WSN cluster is successful, the attacker obtains the cluster key and can browse unauthorized data thereafter. The entered states denote the loss of confidentiality. Therefore, the steady-state confidentiality measure can be computed as:

$$\Lambda = 1 - \pi_C \, . \tag{2}$$

For quantifying the reliability of a software system, mean time to failure (MTTF) is a widely-used reliability measure. We also use the mean time to security failure (MTTS) metric as a measure for quantifying the security of WSNs in Section 6.3.

5.2. Performance Metrics

The performance metrics of interest describe the system in terms of throughput, completion time or response time, as defined in queueing theory or networking. Here, we use the throughput as the performance metric for the WSN. By Little's law, the throughput (denoted by X) is defined as:

$$X = \frac{E[N]}{E[R]} \, . \tag{3}$$

For a queue, the throughput equals the average number of jobs in the queueing station ($E[N]$) divided by the average time a job spends in the queueing station ($E[R]$).

5.3. Tradeoff Metric

In order to investigate how the security of WSNs will interact with performance, we define a tradeoff metric. An objective function formed from the product of the security attribute confidentiality and system throughput is created to demonstrate the tradeoff situation:

$$T = \Lambda \times X \, . \tag{4}$$

As a WSN designer, one may look forward to maintaining the confidentiality of sensitive information with higher throughput, as for the tradeoff measure, the larger the better. In the next section, we will evaluate these measures by computing the steady-state probability of the CTMC and solving the queueing model.

6. Model Analysis

In this section, we derive and evaluate the security and performance attributes using methods for the quantitative assessment of dependability, known as the dependability attributes, e.g., reliability, availability and safety, which have been well established quantitatively.

6.1. CTMC Steady-State Probability Computation

For the system security attributes, we have described the system's dynamic behavior by a CTMC model with the state space $X_s = \{R, G, T, C\}$ and the transitions between these states. In order to carry out the security quantification analysis, we need to determine the stationary distribution of the CTMC model.

The steady-state probabilities $\{\pi_i, i \in X_s\}$ of the CTMC can be computed by solving the system of linear equations [31]:

$$\boldsymbol{\pi}\mathbf{Q} = 0, \tag{5}$$

where $\boldsymbol{\pi} = [\pi_R, \pi_G, \pi_T, \pi_C]$ and $\mathbf{Q}$ is the infinitesimal generator (or transition-rate matrix), which can be written as:

$$\mathbf{Q} = \begin{array}{c} \\ R \\ G \\ T \\ C \end{array} \begin{array}{c} \begin{array}{cccc} R & G & T & C \end{array} \\ \begin{pmatrix} -\lambda_4 & \lambda_4 & 0 & 0 \\ \lambda_1 & -\lambda_1 - \lambda_2 & \lambda_2 & 0 \\ \lambda_1 & \lambda_6 & -\lambda_1 - \lambda_3 - \lambda_6 & \lambda_3 \\ \lambda_5 & 0 & 0 & -\lambda_5 \end{pmatrix} \end{array} \tag{6}$$

In addition, we have the total probability relationship:

$$\sum_i \pi_i = 1 \quad i \in X_s. \tag{7}$$

The transition-rate matrix $\mathbf{Q}$ describes the dynamic behavior of the security model, as shown in Figure 2. The first step towards quantitatively evaluating security attributes is to find the steady-state probability vector π of the CTMC states by solving Equations (5) and (7). We obtain:

$$\begin{aligned} \pi_R &= \frac{[(\lambda_1 + \lambda_2)(\lambda_1 + \lambda_3) + \lambda_1\lambda_6]\lambda_5}{\phi}, \\ \pi_G &= \frac{(\lambda_1 + \lambda_3 + \lambda_6)\lambda_4\lambda_5}{\phi}, \\ \pi_T &= \frac{\lambda_2\lambda_4\lambda_5}{\phi}, \\ \pi_C &= \frac{\lambda_2\lambda_3\lambda_4}{\phi}, \end{aligned} \tag{8}$$

for the sake of brevity, where $\phi = (\lambda_1 + \lambda_4)(\lambda_1 + \lambda_3 + \lambda_6)\lambda_5 + [(\lambda_1 + \lambda_4)\lambda_5 + (\lambda_4 + \lambda_5)\lambda_3]\lambda_2$.

Given the steady-state probabilities of CTMC model, the confidentiality measure Λ and Θ measures can be computed via Equations (1) and (2):

$$\Theta = w\frac{[(\lambda_1 + \lambda_2)(\lambda_1 + \lambda_3) + \lambda_1\lambda_6]\lambda_5}{\phi} + (1 - w)\frac{\lambda_2\lambda_3\lambda_4}{\phi}, \tag{9}$$

$$\Lambda = 1 - \frac{\lambda_2\lambda_3\lambda_4}{\phi}. \tag{10}$$

6.2. Throughput Analysis

Let the total system life time be T. The steady-state probabilities π_i can be interpreted as the proportion of time that the CTMC spends in state i. In the good state G and timing attack state T, the number of jobs precessed is $\lambda(\pi_G + \pi_T)T$, given that the queues are stable $(\lambda < \mu)$. While in the rekeying state R, the sensors do not transmit any data to the gateway node. In the compromised state C, all of the data dispatched to the gateway node are not secure, so they do not contribute to the throughput. Therefore, the system throughput is:

$$X = \frac{\lambda(\pi_G + \pi_T)T}{T} = (\pi_G + \pi_T)\lambda \ . \tag{11}$$

6.3. CTMC with Absorbing State: MTTSF Analysis

Mean time to failure (MTTF) provides the mean time it takes for the system to reach one of the designated failure states, given that the system starts in a good state. We use the mean time to security failure (MTTSF) as a measure for quantifying the security of the WSN. MTTF or MTTSF can be evaluated by making the compromised state of the CTMC an absorbing state, as shown in Figure 3. Once the system reaches the absorbing state (the compromised state *C*), the probability of moving out of this state is zero, i.e., there are no outgoing arcs from such states.

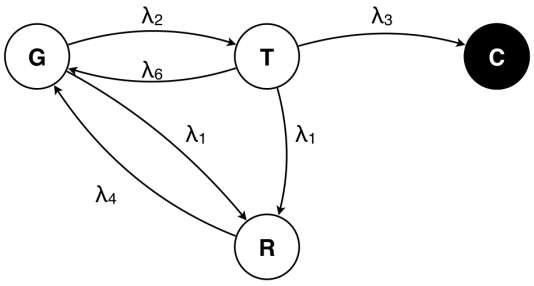

Figure 3. CTMC model with an absorbing state.

Given a CTMC with one absorbing state, we may enter this chain at some state i with probability α_i. For each state that is visited, the time before going to the next state follows an exponential distribution. Thus, the time required to reach the absorbing state from an initial state i is a sum of samples from exponential distributions. The mean time to absorption can be solved easily by introducing the phase-type distribution (PH distribution). For a given CTMC, a PH distribution is defined as the distribution of the time to absorption that can be observed along the paths in a CTMC with one absorbing state [32]. The first moment of a PH distribution exactly expresses the mean time to absorption in an absorbing CTMC:

$$E[X] = -\underline{\alpha}\mathbf{T}^{-1}\underline{1} \ , \tag{12}$$

where $\underline{\alpha} = \{\alpha_i, i \in X_s\}$ is the row vector of initial probabilities of the states and $\underline{1}$ is the column vector of ones. Substituted into our model parameters, we get:

$$MTTSF = \frac{(\lambda_1 + \lambda_4)(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_6)}{\lambda_2\lambda_3\lambda_4} \ . \tag{13}$$

When a software lifetime is given as T_l, we assume the supremum to be denoted by $sup(T_l)$. In order to ensure system security, the MTTSF should be larger than the $sup(T_l)$. That is:

$$\frac{(\lambda_1 + \lambda_4)(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_6)}{\lambda_2\lambda_3\lambda_4} > sup(T_l) \ . \tag{14}$$

Given the system parameters, the required rekeying rate λ_1 can be obtained by solving the inequality Equation (14). It has two roots, and we only take the positive root as:

$$\lambda_1 > \frac{1}{2}\left(-\lambda_2 - \lambda_3 - \lambda_4 - \lambda_6 + \sqrt{\sigma^2}\right) \ , \tag{15}$$

in which:

$$\sigma^2 = \lambda_2^2 + (\lambda_3 - \lambda_4 + \lambda_6)^2 + 2\lambda_2 (\lambda_3 - \lambda_4 + \lambda_6 + 2\lambda_3\lambda_4 sup(T_l)) . \quad (16)$$

Then, the system rekeying rate can be determined for a certain software lifetime. After determining the security and performance measures for the model, we conduct the analysis of the tradeoff between security and performance of WSNs in the following section.

7. Evaluation

Next, we evaluate our secure scheme for WSNs against timing attacks. Experiments have been conducted to investigate:

1. the effectiveness of the random padding countermeasure against timing attacks. We study how the parameter sets of the random distribution affect the mitigation effectiveness on timing attacks; and
2. the improvement of the performance and security tradeoff. From our approach, we investigate the optimal rekey interval for the WSN system under the threat of timing attacks.

7.1. Experiment Setup

Our experiments are developed using the OMNeT++ [33] tool based on the MiXiM [34] and the INET framework [35]. The connection between sensors and gateway nodes is enabled by the IEEE 802.15.4 standard. All tests were run under Mac OS X 10.10 on a 2.6-GHz Intel Core i5 processor with 8 GB of 1600-MHz DDR3 RAM.

Simply, a timing attack is on in which the attacker repeatedly send guesses about a secret value to the target machine, which rejects them as incorrect. However, if his or her first byte of the guess is correct, it takes a very slightly longer time to return the error. With enough measurements and proper filtering, the attacker can distinguish this difference and guess the secret correctly. The key idea of conducting a timing attack is to find the processing time differences. In order to simplify the implementation, we mimic a timing attack by recording and analyzing the gateway node response time to compare two values bit by bit.

7.2. Timing Attack Resilience

In this section, we investigate the timing attack resilience of WSNs with a random delay countermeasure. We evaluate the mitigation effectiveness of the countermeasure in terms of the number of additional measurements an attacker needs to make to achieve a certain percentage of successful guesses about the cluster key.

The attacker sends to the gateway node two messages with a certain bit equal to zero and one, respectively (as the two messages g and $g_h i$ in [11]), while the rest of bits are the same. After the gateway node processes each received message, random delays are padded before sending the responses. Different numbers of timing samples are taken from the attacker's results. When the attacker can tell the time difference accurately from statistical analysis of the samples, we call it a successful guess, and we use the percentage of successful guesses to represent the mitigation influence upon timing attacks exercised by the random padding countermeasure.

We choose Weibull distributed delay as the mitigation against timing attacks because it is easy to adjust the mean and the variance of Weibull distribution [36] by tuning the parameters, and the Weibull distribution is widely used in reliability engineering and failure analysis. We perform an experiment with different parameter sets for the Weibull distributed delays (as shown in Table 2). The experiment is conducted by changing the Weibull distribution shape parameter $k \in \{0.4, 0.37, 0.35, 0.34\}$ while keeping the scale parameter $\eta = 0.05$ and making the situation when no random paddings are added as a bench mark. The result is depicted in Figure 4. One can see that when the shape parameter k is

becoming larger, the attacking client needs more samples to guess the cluster key. As a consequence, the effectiveness of the mitigating countermeasure is becoming greater. We assume that 90 percent right guesses are adequate for a successful attack. It shows that the attacker only needs about 375 samples to get 90 percent successful guesses when no random delay is padded. While when Weibull distributed delays have the scale parameter as 0.34, 1750 samples are needed for the attacker to conduct a successful timing attack. We believe that this is due to the variance of the Weibull random delay growing with the shape parameter k as:

$$var(X) = \eta^2 \left[\Gamma\left(1 + \frac{2}{k}\right) - \left(\Gamma\left(1 + \frac{1}{k}\right)\right)^2 \right]. \quad (17)$$

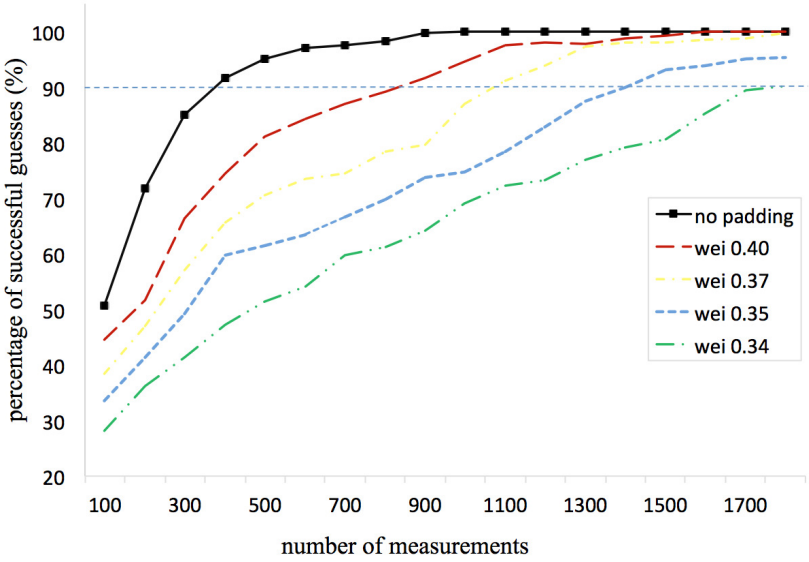

Figure 4. The mitigation effectiveness of different random delay paddings.

Table 2. Optimum rekeying rate for the WSN with different parameter sets of the Weibull distribution.

Shape Para	Scale Para	Variance	n (Sample)	Optimal Rate
no padding			375	0.4348
wei 0.40	0.05	0.2725	830	0.3000
wei 0.37	0.05	0.5642	1070	0.2653
wei 0.35	0.05	0.9980	1400	0.2326
wei 0.34	0.05	1.6151	1750	0.2084

7.3. Performance and Security Tradeoff

In this section, we evaluate the different measures proposed in Section 5 and conduct the tradeoff analysis using the model analysis methods. The parameters for the model are taken from the implementation, and we use normalization. It is assumed that the attack rate on the system is $\lambda_2 = 1$. We also assume it takes an average of 0.5 time units for the gateway node to carry out a rekeying process, and the rate at which the system is brought back to the good state by the rekeying process is $\lambda_4 = 2$. The rate at which the attacker successfully breaks the key, while failing at accessing the data or he or she just fails to conduct a successful timing attack, is assumed to be $\lambda_6 = 1$.

Firstly, the effect of the rekeying rate λ_1 and the weighting parameter w on the system cost measure Θ is analyzed in Figure 5. The marginal values are investigated firstly. It is shown from the figure that when $w = 0$, where only the costs of losing sensitive information in the compromised system are considered, the system cost decreases monotonically with the rekeying rate λ_1. Intuitively,

in this case, when we trigger the rekeying process more often, the security cost will decrease because the WSN is more likely to be brought back to the good working state. When we put all weight on the rekeying effort ($w = 1$), the cost increases with the rekeying rate. The light color in the middle of the figure shows the optimum rekeying rate. For the middle values of the weighting parameter w, the optimum rekeying rate for the lowest cost decreases when we put more weight on rekeying effort cost. For each specific rekeying rate λ_1, the system cost is a straight line weighting the two kinds of cost. In this figure, we get the largest rekeying effort cost and lowest security cost at rate $\lambda_1 = 2.0$.

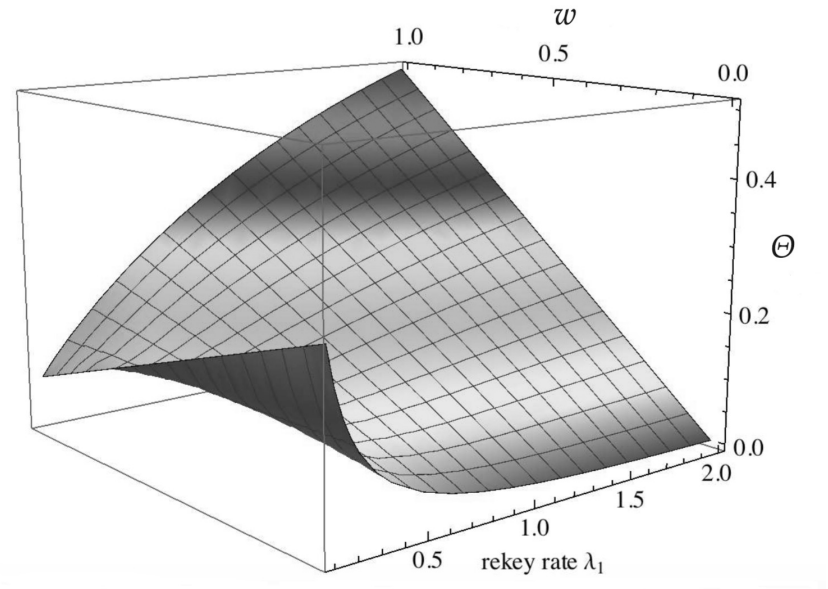

Figure 5. Θ over rekeying rate λ_1 and weighting parameter w.

Figure 6 shows the system performance and security measures, i.e., system confidentiality Λ (defined in Equation (2)) and throughput X (defined in Equation (3)), changing with the rekeying rate λ_1. It can be seen that the confidentiality measure monotonically increases with growing rekeying rate λ_1, because the security improves when the system launches the rekeying process more frequently, as the system is more likely to be brought back to the good state from the timing attack state and the compromised state. While for the throughput measure, at small values of the rekeying rate, it is pretty low, because more time is spent in the compromised state when the throughput is not contributing. After obtaining the maximum throughput, the more often the gateway node triggers the rekeying act, the more often the gateway node denies sensor requests. As a result, the system throughput decreases with the rekeying rate. For a certain value of the rekeying rate λ_1, the performance and security of WSNs are both improved when random paddings are interposed.

At last, we present the security and performance tradeoff analysis for WSNs in Figure 7. The tradeoff measure T, defined in Equation (4), increases with the rekeying rate λ_1 at its low values, as the WSN security improves quickly. The rekeying rate λ_1 indicates how often the system launches the rekeying process. We find the optimum rekeying rate for the best security and performance tradeoff at $\lambda_1 = 0.4348$, when no random delays are added. However, after reaching the optimum value, the tradeoff measure decreases much slower as the rekeying rate is getting larger. When the rekeying rate has a large value, the system tradeoff metric decreases because of the degrading system throughput at large rekeying rates. As Weibull random delays are padded, the tradeoff metric is greater than that with no random delay padding. This shows that the random delay countermeasure can improve the security and performance tradeoff effectively. For a specific value of the rekeying rate λ_1, the trade measure always improves as random delays are padded. From Table 2, it can also be seen that the

optimal rekeying rate decreases when the variance of the Weibull distributed delay is growing. As a consequence, the system needs to make less effort to launch the rekeying process, which leads to less system cost. The command node can adjust the rekeying rate for the gateway node to obtain the minimum cost or maximum security performance tradeoff of the WSN.

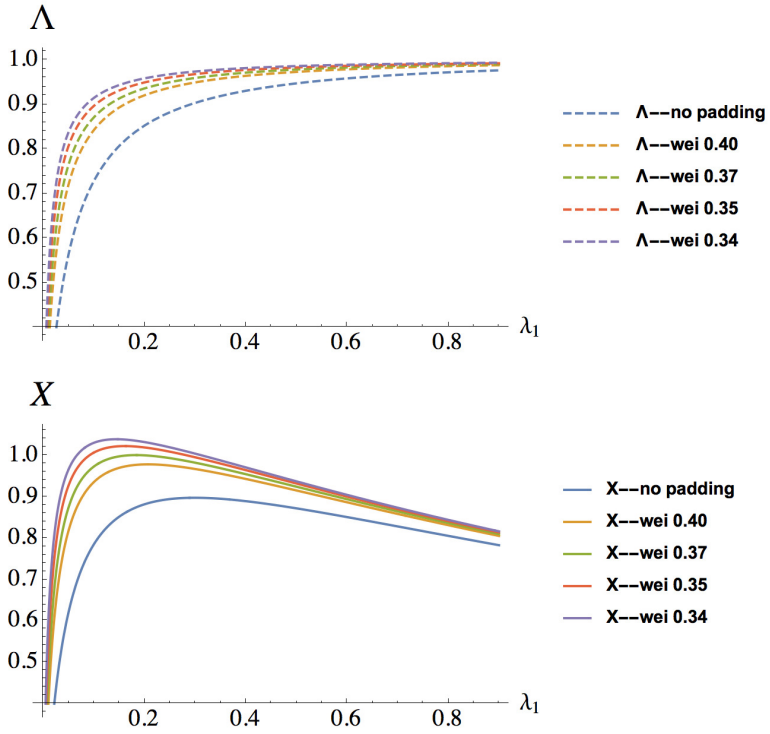

Figure 6. Confidentiality and throughput changing with rekeying rate λ_1.

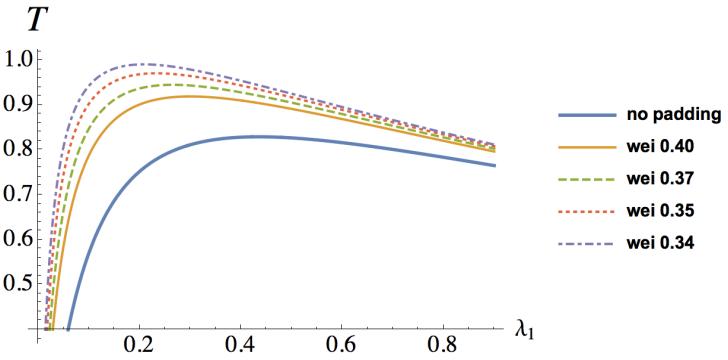

Figure 7. Security and performance tradeoff changing with rekeying rate λ_1.

8. Conclusions

A secure scheme consists of a rekeying procedure, and random delay padding is proposed for WSNs under the specific threat of timing attacks. In order to proceed to a quantitative treatment of the security-performance tradeoff, we have proposed a hybrid CTMC and queueing model for a WSN. We have shown how to formulate measures that include both security and performance attributes and that optimize the tradeoff between the two. System metrics are also proposed that take into account both the rekeying effort a system makes and the sensitive information loss. We obtained the optimal rekeying rate for the tradeoff measure.

Acknowledgments: This work is partially supported by the Major Project of the Chinese National Programs for Fundamental Research and Development (973 Program, No. 2015CB352400), the National Nature Science Foundation of China (Grant Nos. 61501443, U1401258), the Science and Technology Planning Project of Guangdong Province (2015B010129011), the Shenzhen Fundamental Research Project (Grant Nos. JCYJ20150630151329298, JCY20151117161854942), Shenzhen Peacock Plan (Grant No. KQTD2015071715073798), Public Welfare Research and Capacity Building Project of Guangdong Province (No. 2015A030401016) and the Leading Talents of Guangdong Province Program (Grant No. 00201510).

Author Contributions: Tianhui Meng proposed the CTMC and queueing model, then further designed the security scheme. Xiaofan Li constructed the experiments, and Sha Zhang analyzed the data results. Yubin Zhao optimized the scheme and the WSN platform.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless sensor networks: A survey. *Comput. Netw.* **2002**, *38*, 393–422.
2. Lewis, F.L.; Cook, D.; Das, S. Wireless sensor networks. *Smart Environments: Technologies, Protocols, and Applications*; Wiley Online Library: New York, NY, USA, 2004; pp. 11–46.
3. Zhu, H.; Gao, L.; Li, H. Secure and Privacy-Preserving Body Sensor Data Collection and Query Scheme. *Sensors* **2016**, *16*, doi:10.3390/s16020179.
4. Guo, L.; Li, Y.; Cai, Z. Minimum-latency aggregation scheduling in wireless sensor network. *J. Comb. Optim.* **2016**, *31*, 279–310.
5. Holthoff, E.L.; Stratis-Cullum, D.N.; Hankus, M.E. A nanosensor for TNT detection based on molecularly imprinted polymers and surface enhanced Raman scattering. *Sensors* **2011**, *11*, 2700–2714.
6. Sen, J. Security in wireless sensor networks. In *Wireless Sensor Networks: Current Status and Future Trends*; CRC Press: Boca Raton, FL, USA, 2012; p. 407.
7. Bista, R.; Chang, J.W. Privacy-preserving data aggregation protocols for wireless sensor networks: A survey. *Sensors* **2010**, *10*, 4577–4601.
8. Köpf, B.; Basin, D. Automatically deriving information-theoretic bounds for adaptive side-channel attacks. *J. Comput. Secur.* **2011**, *19*, 1–31.
9. Cagalj, M.; Perkovic, T.; Bugaric, M. Timing Attacks on Cognitive Authentication Schemes. *IEEE Trans. Inform. Forensics Secur.* **2015**, *10*, 584–596.
10. Brumley, B.B.; Tuveri, N. Remote timing attacks are still practical. In *Computer Security–ESORICS 2011*; Springer: Berlin, Germany, 2011; pp. 355–371.
11. Brumley, D.; Boneh, D. Remote timing attacks are practical. *Comput. Netw.* **2005**, *48*, 701–716.
12. Roosta, T.; Shieh, S.; Sastry, S. Taxonomy of security attacks in sensor networks and countermeasures. In Proceedings of the 1st IEEE International Conference on System Integration and Reliability Improvements, Hanoi, Vietnam, 6–8 December 2006; Volume 25, p. 94.
13. Hill, J.; Szewczyk, R.; Woo, A.; Hollar, S.; Culler, D.; Pister, K. System architecture directions for networked sensors. In *ACM SIGOPS Operating Systems Review*; ACM: New York, NY, USA, 2000; Volume 34, pp. 93–104.
14. Perrig, A.; Stankovic, J.; Wagner, D. Security in wireless sensor networks. *Commun. ACM* **2004**, *47*, 53–57.
15. Shi, E.; Perrig, A. Designing secure sensor networks. *IEEE Wirel. Commun.* **2004**, *11*, 38–43.
16. Wood, A.D.; Stankovic, J.A. Denial of service in sensor networks. *Computer* **2002**, *35*, 54–62.
17. Han, G.; Shen, W.; Duong, T.Q.; Guizani, M.; Hara, T. A proposed security scheme against Denial of Service attacks in cluster-based wireless sensor networks. *Secur. Commun. Netw.* **2014**, *7*, 2542–2554.
18. Adamy, D. *EW 102: A Second Course in Electronic Warfare*; Artech House: Norwood, MA, USA, 2004.
19. Chen, X.; Makki, K.; Yen, K.; Pissinou, N. Sensor network security: A survey. *IEEE Commun. Surv. Tutor.* **2009**, *11*, 52–73.
20. Karlof, C.; Wagner, D. Secure routing in wireless sensor networks: Attacks and countermeasures. *Ad Hoc Netw.* **2003**, *1*, 293–315.
21. Wang, Y.; Attebury, G.; Ramamurthy, B. A survey of security issues in wireless sensor networks. *IEEE Commun. Surv. Tutor.* **2006**, *8*, 2–23.
22. Coron, J.S.; Kizhvatov, I. An efficient method for random delay generation in embedded software. In *Cryptographic Hardware and Embedded Systems-CHES 2009*; Springer: Berlin, Germany, 2009; pp. 156–170.

23. He, Z.; Deng, X.; Yang, B.; Dai, K.; Zou, X. A SCA-resistant processor architecture based on random delay insertion. In Proceedings of the IEEE 2015 International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 26–27 Feburary 2015; pp. 278–281.
24. Kotipalli, S.; Kim, Y.B.; Choi, M. Asynchronous advanced encryption standard hardware with random noise injection for improved side-channel attack resistance. *J. Electr. Comput. Eng.* **2014**, *2014*, 837572.
25. Younis, M.; Youssef, M.; Arisha, K. Energy-aware routing in cluster-based sensor networks. In Proceedings of the 10th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, Fort Worth, TX, USA, 16 Octomber 2002; pp. 129–136.
26. Jolly, G.; Kuşçu, M.C.; Kokate, P.; Younis, M. A low-energy key management protocol for wireless sensor networks. In Proceedings of the 8th IEEE International Symposium on Computers and Communication (ISCC 2003), Kiris-Kemer, Turkey, 30 June–3 July 2003; pp. 335–340.
27. Eschenauer, L.; Gligor, V.D. A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA, 18–22 November 2002; ACM: New York, NY, USA, 2002; pp. 41–47.
28. Steiner, M.; Tsudik, G.; Waidner, M. Diffie-Hellman key distribution extended to group communication. In Proceedings of the 3rd ACM conference on Computer and Communications Security, New Delhi, India, 14–15 March 1996; pp. 31–37.
29. Steiner, M.; Tsudik, G.; Waidner, M. Key agreement in dynamic peer groups. *IEEE Trans. Parallel Distrib. Syst.* **2000**, *11*, 769–780.
30. Trivedi, K.S. *Probability & Statistics With Reliability, Queuing and Computer Science Applications*; John Wiley & Sons: New York, NY, USA, 2008.
31. Stewart, W.J. *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*; Princeton University Press: Princeton, NJ, USA, 2009.
32. Neuts, M.F. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*; Courier Corporation: New York, NY, USA, 1981.
33. Varga, A.; Hornig, R. An overview of the OMNeT++ simulation environment. In Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, Marseille, France, 3–7 March 2008; ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering): Brussels, Belgium, 2008; p. 60.
34. Köpke, A.; Swigulski, M.; Wessel, K.; Willkomm, D.; Haneveld, P.; Parker, T.E.; Visser, O.W.; Lichte, H.S.; Valentin, S. Simulating wireless and mobile networks in OMNeT++ the MiXiM vision. In Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, Marseille, France, 3–7 March 2008; ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering): Brussels, Belgium, 2008; p. 71.
35. INET Framework Home Page. Available online: https://inet.omnetpp.org/ (accessed on 1 August 2016).
36. Papoulis, A.; Pillai, S.U. *Probability, Random Variables, and Stochastic Processes*; Tata McGraw-Hill Education: Noida, India, 2002.