Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 9021650, 13 pages
http://dx.doi.org/10.1155/2016/9021650

Research Article

An Improved μTESLA Protocol Based on Queuing Theory and Benaloh-Leichter SSS in WSNs

Haiping Huang,[1,2] Tianhe Gong,[1,2] Tao Chen,[1,2] Mingliang Xiong,[1,2] Xinxing Pan,[1,2] and Ting Dai[3]

[1]*College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[2]*Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China*
[3]*Department of Computer Science, College of Engineering, North Carolina State University, Raleigh, NC 27695, USA*

Correspondence should be addressed to Haiping Huang; hhp@njupt.edu.cn

Received 15 March 2016; Accepted 5 July 2016

Academic Editor: Iftikhar Ahmad

Broadcast authentication is a fundamental security technology in wireless sensor networks (ab. WSNs). As an authentication protocol, the most widely used in WSN, μTESLA protocol, its publication of key is based on a fixed time interval, which may lead to unsatisfactory performance under the unstable network traffic environment. Furthermore, the frequent network communication will cause the delay authentication for some broadcast packets while the infrequent one will increase the overhead of key computation. To solve these problems, this paper improves the traditional μTESLA by determining the publication of broadcast key based on the network data flow rather than the fixed time interval. Meanwhile, aiming at the finite length of hash chain and the problem of exhaustion, a self-renewal hash chain based on Benaloh-Leichter secret sharing scheme (SRHC-BL SSS) is designed, which can prolong the lifetime of network. Moreover, by introducing the queue theory model, we demonstrate that our scheme has much lower key consumption than μTESLA through simulation evaluations. Finally, we analyze and prove the security and efficiency of the proposed self-renewal hash chain, comparing with other typical schemes.

1. Introduction

We can imagine there will be thousands of sensors deployed in the future space, but how can we ensure the security of these sensors? Aside from confidential communications, authentication is one of the essential services in security protocols of wireless sensor networks (ab. WSNs) system [1]. If the authentication system stays defective or noneffective, attackers may launch threats to the whole network such as the wormhole attack, the man-in-the-middle attack, and the multiple identities attack. Data leakage may occur even in a military area, which can cause serious consequences. Therefore, the study of authentication system especially the broadcast authentication protocol for large-scale WSNs still remains challenging. However, restrained by the finite resources of WSNs, many previous protocols cannot be directly applied to the broadcast authentication of WSNs. For example, most protocols rely on asymmetric mechanism such as the public key cryptography, but this mechanism has heavy communication, computation, and storage overhead, which are impractical for WSNs.

Therefore, designing a protocol that can guarantee the data integrity, confidentiality, and authentication in the broadcast has been a popular research topic in WSNs. One straightforward solution is to let the base station and all other nodes share a common broadcast authentication key, but the key will be disclosed if one of nodes is corrupt. Another solution is to use one-time key for each packet so that the leak of current key will not have a bad influence on the following packets, but the cost of frequently updating keys is unacceptable for WSNs. Perrig et al. proposed a classic broadcast authentication protocol μTESLA [2], which has a great improvement over the original protocol TESLA [3, 4]. The contribution of μTESLA protocol is to implement a broadcast authentication process based on the symmetric key mechanism instead of the asymmetric one, and it overcomes

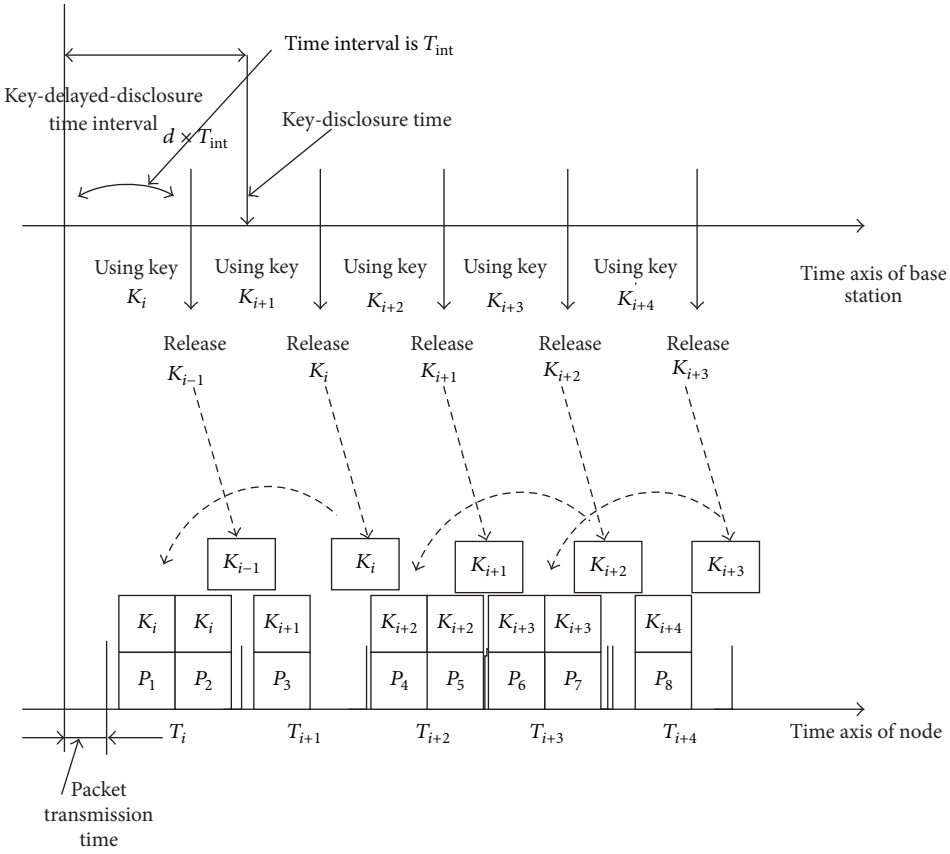

Figure 1: The broadcast authentication process of μTESLA.

the problems in traditional protocols by delaying the publication of one-way hash function key. This protocol decreases the computational complexity for broadcast authentication and improves the authentication efficiency as well. In the following paragraph, we will give a brief overview of μTESLA.

The main idea of μTESLA is to broadcast a packet authenticated by the key K_{mac} at first and then publish K_{mac} so that there is no way to forge the broadcast packets before the publication of the key. In addition, the protocol achieves the secret sharing with the key generation algorithm shared by the entire network. The one-way hash function and the key chain mechanism can ensure the safety of keys and the tolerance of packet loss. Figure 1 illustrates the broadcast authentication process of μTESLA.

μTESLA protocol consists of three phases: (1) securely initializing the configuration of base station, (2) bootstrapping the new receivers, and (3) authenticating the broadcast packets. The base station generates a key pool $(K_{N-1}, K_{N-2}, \ldots, K_1, K_0)$ by one-way hash function in the first phase and determines the synchronization time interval T_{int} and the key-delayed-disclosure time interval $d \times T_{\text{int}}$. The synchronization time interval represents the lifetime of a broadcast key, which means the broadcast packets sent from the base station use the same key K_i in a synchronous period $[i \times T_{\text{int}}, (i+1) \times T_{\text{int}}]$. The value of integer d should make $d \times T_{\text{int}}$ longer than the time of packet-switching between the base station and the farthest node so that all the nodes can be ensured to have received the broadcast packet before the corresponding key is disclosed.

When the new node joins the network, μTESLA distributes the key synchronized parameters and initialized related keys to the new node based on the SNEP protocol [3]. For example, Figure 1 shows the process of node A requesting to join the broadcast network during the time interval $[i \times T_{\text{int}}, (i+1) \times T_{\text{int}}]$. Consider

$$\begin{aligned} A &\longrightarrow S: \left(N_A \parallel D_{\text{req}}\right) \\ S &\longrightarrow A: \left(T_S \parallel K_i \parallel T_i \parallel T_{\text{int}} \parallel d\right), \\ &\quad \text{MAC}\left(K_{AS}, N_A \parallel T_S \parallel K_i \parallel T_i \parallel T_{\text{int}} \parallel d\right), \end{aligned} \tag{1}$$

where N_A is a nonce which is generated by A to achieve a strong freshness authentication; D_{req} is a request data packet; K_{AS} is an authentication key between A and S; T_S is the current time; K_i is an initial key; T_i is the starting time of the current synchronization interval; T_{int} is the synchronization interval; and d is the disclosure delay. The key will be published after $d \times T_{\text{int}}$.

After receiving a broadcast packet from the base station, the receiver will judge the validity of authentication key based on the synchronization time. The node will further verify the key's validity by running the hash calculation on it. Finally, the node will use the key to authenticate the packets that have been stored in the buffer during the time interval.

In μTESLA protocol, the publication of key is dependent on a specific time interval, which is fixed after initialization. However, we notice that the current network traffic is not stable in each time interval, and we divide this unstable traffic into two cases:

(i) The base station broadcasts the packets frequently to the sensor nodes. In this case, the broadcast packets in one time interval will dramatically increase. If the key is still disclosed according to the original time interval, the excessive number of packets is unable to get a timely authentication and the storage space of the sensor nodes will be exhausted inevitably.

(ii) The base station just broadcasts a few packets in a long time. In this case, it is possible that there are few packets during the fixed time interval. Consequently, the release of keys will lead to the increase of communication and computation overhead, which degrades the efficiency of key chain.

To decrease unnecessary consumption as well as to ensure security in the process of broadcast authentication, in this paper, we replace the fixed time interval with network traffic to determine the publication of broadcast key. In other words, the base station will not publish the authentication key unless it has broadcasted a certain number of packets. And our experiment has shown that some drawbacks of μTESLA can be solved based on our mechanism.

Due to the one-way and lightweight characteristics, hash chains have been widely applied to various scenarios such as one-time password system [5], video stream security [6, 7], micropayment protocol [8], key distribution scheme [9], and broadcast authentication [10]. However, there is a trade-off between the length and the efficiency of hash chain. The exhaustion of the current hash chain will inevitably result in producing another new hash chain initialized with the public key cryptography. And this reinitialization will bring about the extra overhead of the network.

Aimed at overcoming the inadequacies of the above schemes, another concern of this paper is to design a novel self-renewal one-way hash chain scheme based on Benaloh-Leichter SSS (SRHC-BL). This scheme can effectively prolong the lifetime of network and increase the tolerance of key loss. Comparing with the typical self-renewal hash chain schemes, our approach has the benefit of higher security and less consumption of communication, computation, and storage.

Therefore, the main contribution of this paper can be summarized as follows:

(1) A novel key distribution method based on data flow instead of fixed time interval is proposed in order to keep network stable in any situations. In addition, some special cases are discussed as the supplement.

(2) A self-renewal one-way hash chain scheme based on Benaloh-Leichter SSS is adapted for both keeping extending life time of network and ensuring the tolerance of key loss.

(3) Simulation experiments and theoretical analysis based on queue model are conducted to compare the storage cost and calculation complex among our schemes and traditional μTESLA protocol. Consequently, the result proves that our design achieves a better performance.

2. Preliminary Knowledge

2.1. Basic Concepts of Queue Theory. Queue theory, also known as random service system theory, is a theoretical basis for the queuing problem. It is one of the interdisciplinary theories of probability, statistics, and operational research. Queuing phenomenon is composed of two aspects: demand service and provide service. Here are four common queuing models as follows: M/D/1/∞ queuing model, M/M/1/∞ queuing model, M/G/1/∞ queuing model, and G/G/1/∞ queuing model.

Queuing system has the following six features, which can be applied to the broadcast authentication in WSNs:

(i) Input process, which characterizes and describes the law of data packets coming to the random service system.

(ii) Service time, namely, the time for the base station to authenticate the data packets.

(iii) Waiter, namely, the base station.

(iv) Size of line determined by the number of customers waiting to be served, which characterizes the number of valid data packets to be processed by the base station.

(v) Customer source, which corresponds to the data packets.

(vi) Queue rule, determined by the detail of queuing model.

2.2. Basic Concepts of Self-Renewal Hash Chain. In this section, we introduce some basic concepts of SSS and the definition of the Benaloh-Leichter SSS.

2.2.1. Concept of SSS. First, we formally define the necessary monotone access structure.

Definition 1. Given a set P, a monotone access structure on P is a family of subsets $Z \subseteq 2^P$ such that

$$\begin{gathered} A \in Z, \\ A \subseteq A' \subseteq P \\ \downarrow \\ A' \in Z. \end{gathered} \tag{2}$$

Let n be an integer, $n \geq 2$, let the set of participants be $P = \{p_1, p_2, \ldots, p_n\}$, and let an access structure Z defined on P be comprised of a collection of subsets of P. Z is a monotone access structure whenever $A \in Z$ and $A \subseteq A' \subseteq P$.

Similarly, Z-SSS is a method of generating $(S, (I_1, \ldots, I_n))$ such that,

(1) for any $A \in Z$, finding the element S, given the set $\{I_i \mid i \in A\}$, is easy,

(2) for any $A \in \overline{Z}$, finding the element S, given the set $\{I_i \mid i \in A\}$, is difficult.

The set Z is the authorized access structure or simply the access structure, S is the secret, and $I_1, \ldots, I_n$ are the shares (or the shadows) of S. The elements of the set Z are the authorized access sets of the scheme.

2.2.2. Benaloh-Leichter SSS

Definition 2. Let P be a set. The set V of variables indexed by P is the set $V = \{v_p : p \in P\}$.

Definition 3. Given a monotone function F on variables indexed by a set P, the access structure defined by F is the set of subsets of A of P for which F is true precisely when the variables indexed by A are set to be true.

It is clear that, for every monotone function F, the access structure defined by F is a monotone access structure.

Definition 4. For a given set P and a monotone access structure Z denoted by $Z_{\min}$ on P, define $F(Z)$ to be the set of monotone function on $|P|$ variables such that, for every formula $F \in F(Z)$, the output of F is true if and only if the true variables in F correspond exactly to a set $A \in Z$.

Note that $F, F' \in F(Z)$ implies F and F' denote the same function. They may, however, use entirely different expressions to express this function.

The formula can be expressed using only $\wedge$ operator and $\vee$ operator, and it is sufficient to indicate how to "split" the secret with these operators.

Definition 5. One can recursively define the share of a secret S with respect to a formula F as follows:

$$F = \begin{cases} (S, i), & \text{if } F = v_i,\ 1 \le i \le n \\ \bigcup_{i=1}^{n} \text{Shares}\,(S, F_i); & \text{if } F = F_1 \vee F_2 \vee \cdots \vee F_n \\ \bigcup_{i=1}^{n} \text{Shares}\,(s_i, F_i); & \text{if } F = F_1 \wedge F_2 \wedge \cdots \wedge F_n, \end{cases} \tag{3}$$

where based on Definitions 1, 2, and 3, selecting the specific integer n and $Z_{\min}$, for the case $F = F_1 \wedge F_2 \wedge \cdots \wedge F_n$, one can use a (k, n)-threshold secret sharing scheme for deriving some shares $s_1, s_2, \ldots, s_k$ corresponding to the secret S, and then every distinct share is assigned to each I_i. Thus one has $I_i = \{s_i \mid (s_i, i) \in \text{Shares}\,(S, F)\}$, for all $1 \le i \le n$, where F is an arbitrary formula in the set F_A.

2.2.3. Definition of Hash Chain

Definition 6. The secure hash function is a publicly known function $f_n : \{0,1\}^* \to \{0,1\}^k$, it takes s as an input, and the output is a bit string $f_n(s)$ of length n. In $f_n(s)$, s is generated randomly from a pseudo-random string generator. One-way hash chain can be visually expressed as follows:

$$s \xrightarrow{h(\cdot)} h(s) \xrightarrow{h(\cdot)} h^2(s) \cdots \xrightarrow{h(\cdot)} h^n(s). \tag{4}$$

3. Our Scheme

3.1. The Key Distribution Algorithm Based on Data Flow. Compared with the traditional μTESLA protocol which releases keys based on the fixed time interval, our approach releases keys according to the data flow based on the queue theory and the renewable hash chain.

3.1.1. Assumptions

(i) μTESLA protocol is as follows:

(1) the packet transmission time between the base station and the farthest node is $T_{\max}$;
(2) the base station releases the key every T_{int} by a fixed time interval;
(3) the delay time of key publication is $\sigma \times T_{\text{int}}$, and it satisfies the condition that $\sigma \times T_{\text{int}} > T_{\max}$;
(4) the verification condition is $\lfloor (T_c + \Delta - T_1)/T_{\text{int}} \rfloor < i + \sigma - 1$, where T_c is the current time, Δ is the maximum clock difference, T_1 is the start time, and i is the ith interval time.

(ii) The improved broadcast authentication protocol based on the queue theory and the renewable hash chain is as follows:

(1) the maximum speed (or frequency) for the base station to send packets is $V_{s_{\max}}$;
(2) the maximum transmission speed (or frequency) in WSNs is $V_{t_{\max}}$;
(3) the communication radius of the base station is R_{bs};
(4) the base station releases the authentication key every N_{int} packets based on data traffic;
(5) the delay of data flow of key publication is $N_{\text{int}} + \theta$, and it satisfies the condition that $(N_{\text{int}} + \theta)/V_{s_{\max}} > R_{\text{bs}}/V_{t_{\max}}$;
(6) the verification condition is $\lfloor (N_c - N_1)/N_{\text{int}} \rfloor < \lfloor i + \theta - 1 \rfloor$, where N_c is the identification number of packets that is currently received, N_1 is the ID number of first packet received, and i is the ith time interval of data flow.

3.1.2. The Process of Key Distribution Based on Data Flow. The process of broadcast authentication based on queue theory and renewable hash chain is shown in Figure 2. Comparing with Figure 1, we can see the difference between μTESLA and

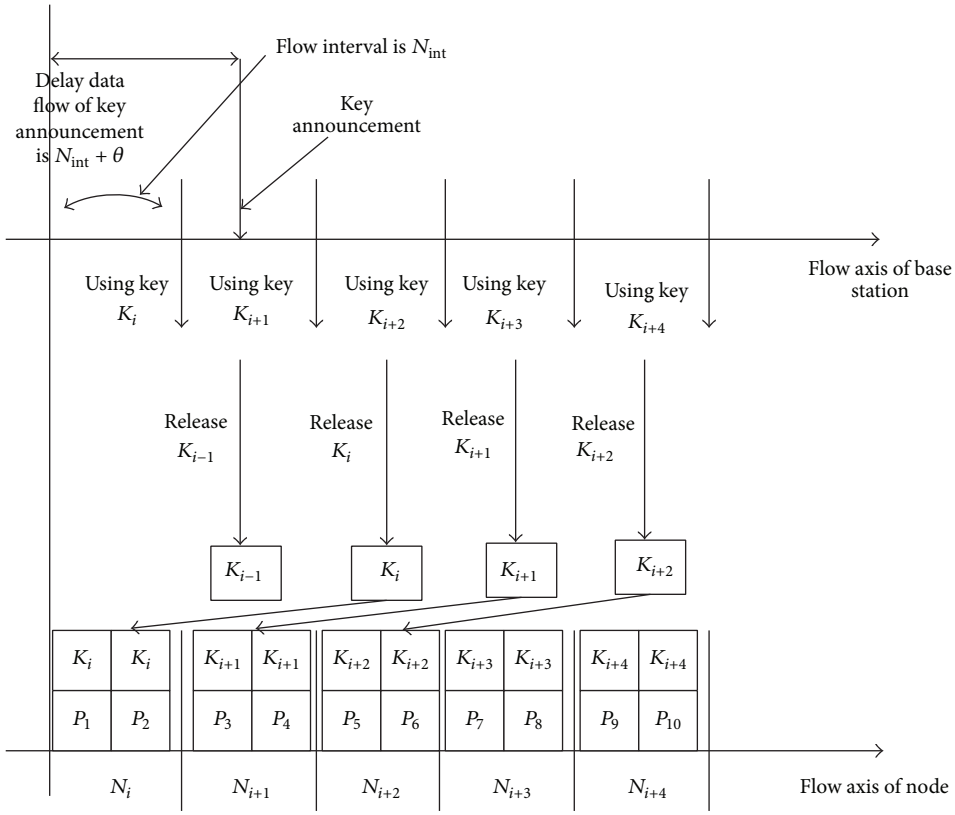

Figure 2: The process of broadcast authentication based on queue theory.

ours; μTESLA maps the key distribution to the time domain, while ours maps the key distribution to the flow domain.

3.1.3. Several Cases to Discuss

Case 1. If the base station has not broadcasted a packet after a long period, and the number of packets broadcasted has not achieved a certain threshold, the base station will not release the key during this long period, which disables the node to authenticate the buffered packets. In this case, we can set a time threshold T (T is the upper bound of broadcast key lifetime). So after time T, the base station is required to release key no matter whether the condition is satisfied.

Case 2. It is very common to have packet loss in WSNs. Consider the following case: the base station will not send packets in a long period and thus the key for the next round will not be released either, but unfortunately, at this time, one node lost the current authentication key, which implies that this node cannot authenticate the remaining packets in the buffer any more. In terms of this case, we set the interval time $2T$ for the node to wait, where T is the upper bound of broadcast key lifetime. If the waiting time exceeds $2T$, the node can send the request message to the base station for the key of current round.

Case 3. Synchronization problem: how do we know which packet should be authenticated by which type of key? We use the counting mechanism to solve this problem. That is, the broadcast packet sent by the base station is counted from 0 to N and authentication key is also numbered from 0 to N so that we can create the relations between the packet and the key by simply mapping.

3.2. A Self-Renewal Hash Chain Based on Benaloh-Leichter SSS. In this section, we propose a novel self-renewed hash chain based on Benaloh-Leichter SSS. This scheme has three phases: the hash chain initial phase, the hash chain usage phase, and the hash chain extension phase. Let C and R denote communication initiator and the recipient, respectively.

3.2.1. Initial Phase. In the initial phase, C and R are synchronized in time, and there is a maximum error time denoted as Δ; R can reject the message which exceeds the time Δ plus the acceptable transmission delay.

(1) The initiator C generates an initial random value s as the seed of the first hash chain, and then C uses the preloaded hash function to compute n hash value of the first hash chain. Consider

$$s \xrightarrow{h(\cdot)} h(s) \xrightarrow{h(\cdot)} h^2(s) \cdots \xrightarrow{h(\cdot)} h^n(s). \quad (5)$$

(2) Then, C selects $Z_{\min}$ based on Benaloh-Leichter SSS and a new random value s' to generate n hash value of the next hash chain. Consider

$$s' \xrightarrow{h(\cdot)} h(s') \xrightarrow{h(\cdot)} h^2(s') \cdots \xrightarrow{h(\cdot)} h^n(s'). \quad (6)$$

(3) Therefore, according to the Benaloh-Leichter SSS, C takes $h^n(s')$ as the secret S, divides it into n parts as the set V, and then defines the set $F(Z)$ as the set of formula on set V. Further, we select an arbitrary formula F in the set F_A. In this case, according to $Z_{\min}$ we can obtain Shares (S, F) of the secret S. Thus, the shares corresponding to the secret S in the access structure Z are distributed as shadows $I_1, I_2, \ldots, I_n$.

3.2.2. Usage Phase

(1) Before the usage phase, C and R have confirmed the initial time T_0, and meanwhile the value $h^n(s)$ and the hash function have been preloaded in R securely, as well as the message authentication code $\mathrm{MAC}_0(h^{n-1}(s) \oplus I_1)$. During the usage phase, the hash value is used from $h^{n-1}(s)$ (firstly) to s (finally) corresponding to the time period $T_0 + i^*\Delta$ $(1 \le i \le n)$.

(2) In the time $T_0 + \Delta$, C releases the Msg_1 and its corresponding message authentication code MAC_1 to R, the formats of Msg_1 and MAC_1 are shown, respectively, as follows:

$$\mathrm{Msg}_1\left(T_0 + \Delta, h^{n-1}(s), I_1, \mathrm{MAC}_1\right), \quad \mathrm{MAC}_1\left(h^{n-2}(s) \oplus I_2\right). \tag{7}$$

So in the time $T_0 + i^*\Delta$ $(1 \le i \le n)$, C will compute and release

$$\mathrm{Msg}_i\left(T_0 + i^*\Delta, h^{n-i}(s), I_i, \mathrm{MAC}_i\right), \quad \mathrm{MAC}_i\left(h^{n-i-1}(s) \oplus I_{i+1}\right), \tag{8}$$

where Msg_i is the content of current message and MAC_i is used to verify MAC_{i-1}.

(3) For the ith authentication, after R receives the Msg_i and MAC_i, R will calculate the difference between the last time of receiving packets and the current time of receiving packets. If the difference has not exceeded Δ, R will carry out the following steps:

(a) Compute and verify whether $h(h^{n-i}(s))$ is equal to $h^{n-i+1}(s)$, where $h^{n-i+1}(s)$ is the valid hash value stored in the last process. If it is equal, R saves it.

(b) Compute and verify whether $I_i \oplus h^{n-i}(s)$ is equal to MAC_{i-1}. If it is, R saves MAC_i and I_i.

On the other hand, if the difference exceeds Δ,

(a) C drops $h^{n-i}(s)$ and I_i and saves MAC_i; then it will wait until the next authentication process, which is assumed as the jth authentication where $i < j$;

(b) compute and verify whether $h^{j-i+1}(h^{n-j}(s))$ is equal to $h^{n-i+1}(s)$, where $h^{n-i+1}(s)$ is the valid hash value stored in the last process; if it is equal, R saves it;

(c) compute and verify whether $h^{n-j}(s) \oplus I_j$ is equal to MAC_{j-1}; if all checks are valid, R verifies C successfully and then stores the shadow I_i.

The hash chain usage phase has a detailed description in $\mu TESLA$. If the hash chain is exhausted, the protocol goes into the hash chain extension phase.

3.2.3. Extension Phase. When one hash chain has been exhausted, R has stored n shadows I_i. One thing we need to notice is that even though the number of shadows that R has stored is less than n (as long as the number is not less than k), we can still recover the final secret S. The detailed description is as follows.

(1) Based on the shadows $I_1, I_2, \ldots, I_n$, we can easily deduce Shares(S, F) corresponding to the secret S with the (k, n)-threshold secret sharing scheme.

(2) With the Shares(S, F), we can simply recover the secret S. In other words, we have obtained the tail of the next hash chain $h^n(s')$. Then, a new hash chain can be applied in the right way, and we can use the same protocol in the next hash chain in order to achieve the purpose of self-renewed one.

Therefore, this protocol provides an on-demand hash chain extension without exhaustion, so the hash chain is able to work smoothly and infinitely.

4. Performance Analysis

4.1. The Key Distribution Algorithm Based on Data Flow. (1) Our algorithm releases the keys based on the data flow instead of the original timeline and takes full account of the uneven distribution of arrival of the packets in the network.

(2) Valid packets simulation in the μTESLA protocol: many simulation techniques in [11, 12] are introduced to wireless sensor networks to help researchers to understand the behavior of the network which is hard to capture in situ. In this paper, we use Matlab to simulate the four queuing models of M/D/1/∞, M/M/1/∞, M/G/1/∞, and GI/G/1/∞, respectively. We take the base station as the waiter and the broadcast packets as the customer source, so the service time obeys the distribution of the packets to be processed and broadcasted by the base station and customer source obeys the distribution of arrival of packets. By considering practical situations, we give an example of packets arriving intensively. The arrival of data packets of M/D/1/∞, M/M/1/∞, and M/G/1/∞ obeys Poisson distribution with the randomly selected parameter $\lambda = 0.5$, while GI/G/1/∞ obeys the general random distribution. We set a fixed time interval T_{int} as 60 s and the numbers of valid packets N_{str} in T_{int} as 20, and the simulation time was half an hour. If the number is over 20, we would consider it as invalid one. There are two reasons for that. First, overly late authentication would cause the large storage overhead caused by the accumulated packets in the node buffer. Second, the message is more likely to be vulnerable to chosen plaintext attacks. It can also be proved that the conclusions of simulation experiments will not change by altering the values of parameters such as λ and T_{int}.

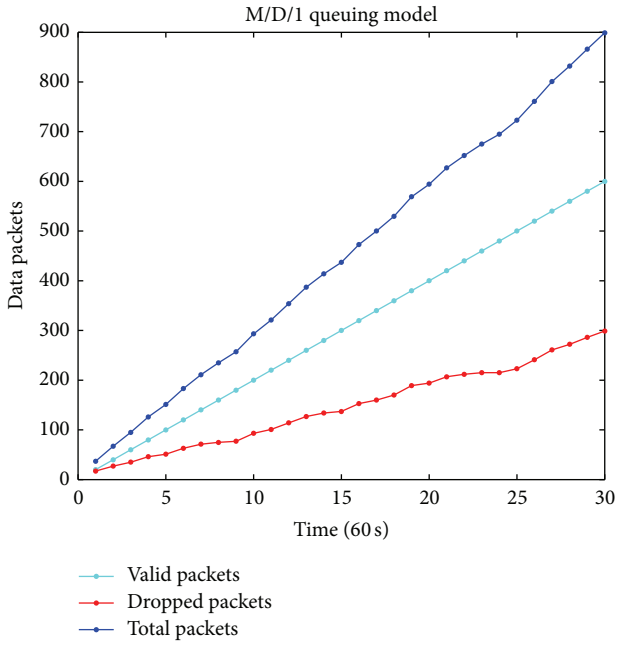

Figure 3: Packets of M/D/1.

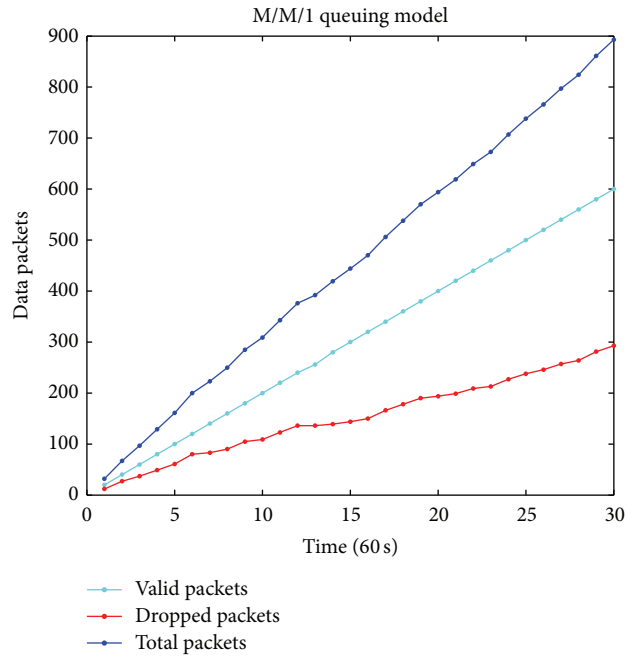

Figure 4: Packets of M/M/1.

(3) Simulation comparison of key packets consumed: we use Matlab to simulate the four queuing models of M/D/1/∞, M/M/1/∞, M/G/1/∞, and GI/G/1/∞, respectively, and we take the example of packets arriving sparsely. (a) The arrival of packets of M/D/1/∞ obeys the Poisson distribution with parameter $\lambda = 0.1$ and the service time obeys the uniform distribution with a fixed value $t = 1$ s. (b) The arrival of packets of M/M/1/∞ obeys the Poisson distribution with parameter $\lambda = 0.1$ and the service time obeys the Poisson distribution with parameter $\mu = 20$. (c) The arrival of packets of M/G/1/∞ obeys the Poisson distribution with parameter $\lambda = 0.1$ and the service time obeys the general random distribution. (d) The arrival of packets of GI/G/1/∞ and the service time obey the general random distribution. We set a fixed time interval $T_{\text{int}} = 60$ s, and the data flow interval is $N_{\text{int}} = 20$; the simulation time was ten hours.

(i) In terms of the intensive rate of packets arrival, based on the fixed time interval, the simulation results of valid data packets, dropped packets, and total packets for 4 queuing models M/D/1/∞, M/M/1/∞, M/G/1/∞, and GI/G/1/∞ are shown in Figures 3–6, respectively.

(ii) In terms of the sparse rate of packets arrival, we draw a comparison between μTESLA (based on the fixed interval) and our protocol (based on the data flow). The simulation results of key consumption for 4 queuing models M/D/1/∞, M/M/1/∞, M/G/1/∞, and GI/G/1/∞ are shown in Figures 7–10, respectively.

From Figures 3–6, we notice that the intensive rate of broadcast packets will cause the packets to be cached in the nodes and unable to be authenticated timely, which eventually results in the loss of packets. Also, the probability of

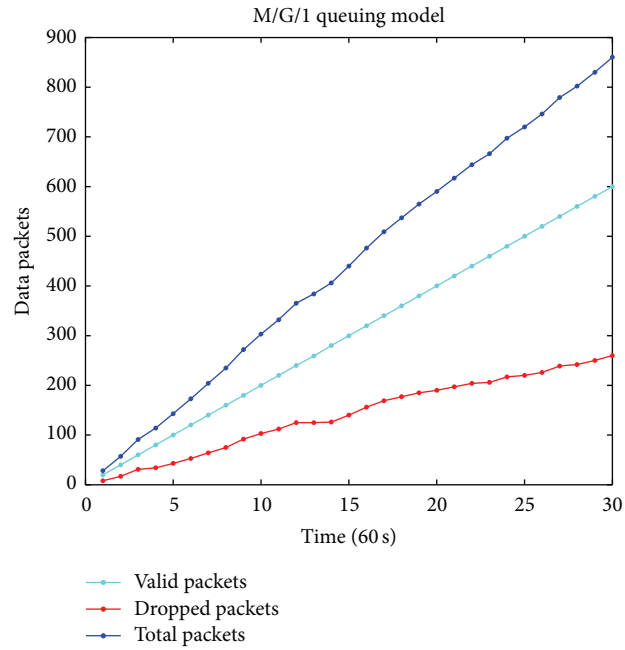

Figure 5: Packets of M/G/1.

choosing plaintext attack will become large if the number of packets exceeds the threshold N_{str}.

Furthermore, from Figures 7–10, the key consumption of our proposal is much lower than that of μTESLA. Consequently, the life cycle of the key chain would be prolonged, and the network overhead would be reduced.

(4) The calculation complexity of the proposed algorithm is low. From Figures 1 and 2, we can find that there is no fallback process in both μTESLA protocol and our algorithm.

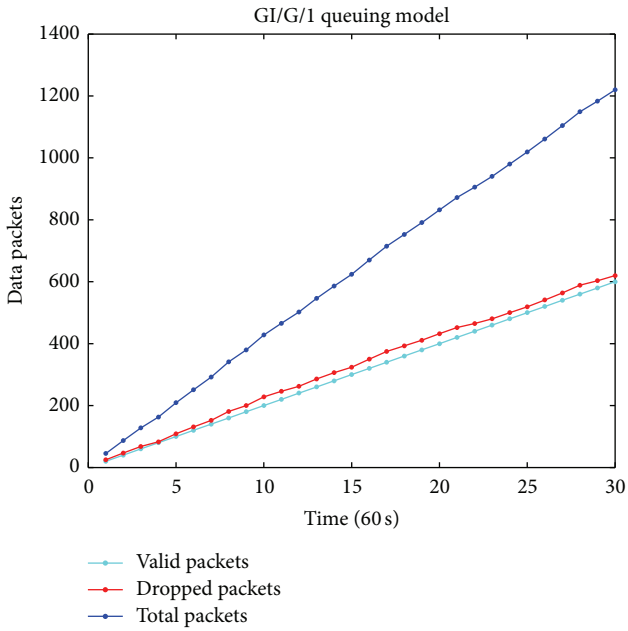

Figure 6: Packets of GI/G/1.

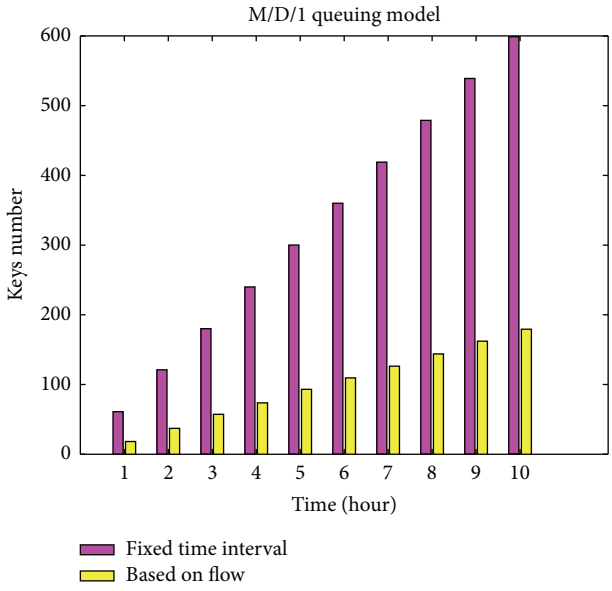

Figure 7: Keys consumption of M/D/1.

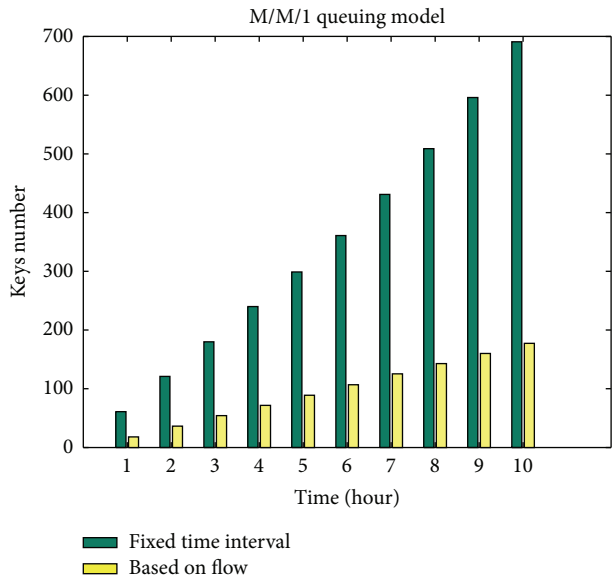

Figure 8: Keys consumption of M/M/1.

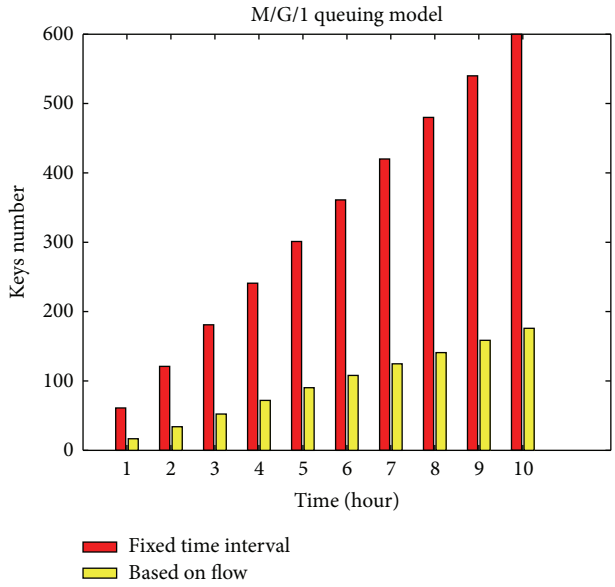

Figure 9: Keys consumption of M/G/1.

Although different network environments can contribute to different consumption of calculation, the proposed algorithm and μTESLA both keep $O(n)$, where n is the number of hash calculations during authentication processes. However, in the protocol of multilevel μTESLA [13], repeated hash operations are conducted to guarantee life time of keys at the expense of large amounts of calculations. For instance, m denotes the time of high-level calculation while n denotes that of low-level calculation in a 2-level μTESLA process, which leads to $m \cdot n$ times of calculation. When $n = m$, the complexity achieves $O(n^2)$; the order of magnitudes increases sharply and contributes to high calculation complexity if n becomes large. The variation tendency can be seen in Figure 11.

4.2. A Self-Renewal Hash Chain Based on Benaloh-Leichter SSS. In this section, we will present the security and performance analysis of the proposed hash chain in Section 3.

4.2.1. Security. The security of this scheme is based on one-way function and Benaloh-Leichter SSS. The purpose of *XOR* with hash value is to maintain the integrity and confidentiality of shadows. And the purpose of delaying key publication is to achieve nonrepudiation.

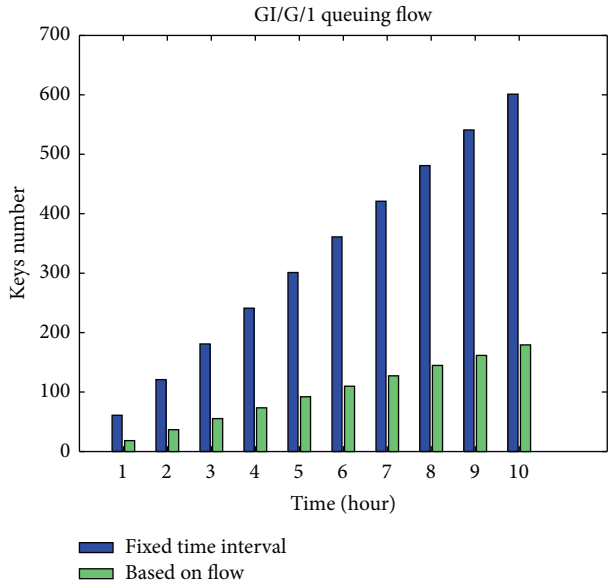

Figure 10: Keys consumption of GI/G/1.

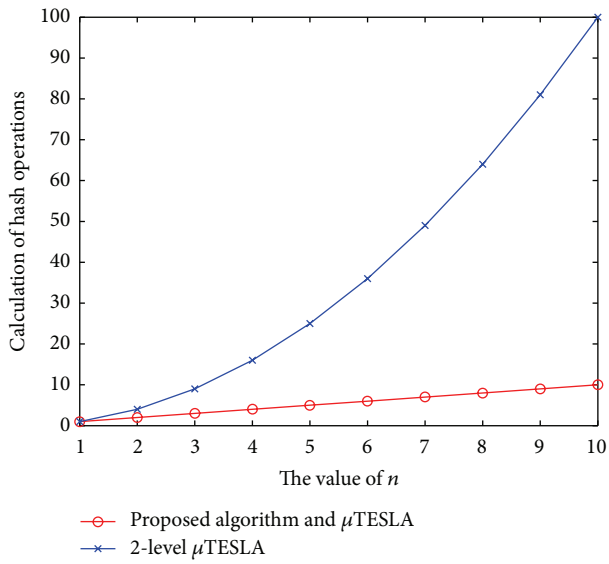

Figure 11: The calculation of three algorithms.

Meanwhile, Benaloh-Leichter SSS can efficiently generate a much richer family of access structures than the current schemes, and it is convenient to view an access structure as a function. Any monotone Boolean function over n variables can be computed by a monotone formula. Thus, every access structure can be realized by the scheme of Benaloh-Leichter SSS. On the other hand, for every set that does not belong to the access structure, the elements in the set do not have any information on s_i; hence they will not reveal any information about secret S.

Also in the phase of authentication, the tolerance of packet loss or fault is embodied in our proposal. However, in Benaloh-Leichter SSS, even some s_i was dropped or lost; secret S can still be verified by other valid s_i as long as the number of shadows is not less than k.

Moreover, dual authentication in our scheme can strengthen the security and integrity. The first authentication is that whether $h^{n-i}(s)$ and I_i are received in a valid interval and they will not be stored unless both of them are verified correctly. And the second authentication is to judge whether $h^{n-i}(s)$ is valid according to $h^{n-i+1}(s)$ which has been stored in the first authentication and whether I_i is valid by the exclusive-OR function. The shadow I_i will be accepted only if the packet passes the dual authentication.

Finally, our self-renewal hash chain has satisfactory confidentiality. However, the shadow I_i exists in the packet with the form of plaintext and the attacker can obtain the key shadow information by snooping the packet. However, the attacker is unlikely to recover the secret S unless he or she can get more than k pieces of shadow, which obviously increases the difficulty. And even though the attacker can finally recover the secret S, he or she is still unable to produce the fake broadcast packets to play the role of the base station. The reason is that the secret S, namely, $h^n(s')$, is the tail of the next hash chain, which can only be used to authenticate the subsequent keys. And due to one-way feature of the hash function, the attacker cannot generate $h^{n-1}(s'), h^{n-2}(s'), \ldots, s'$, so he or she is unable to fake the packet to deceive other sensor nodes. If the attacker does, these nodes can easily detect the validity of packets with $h^n(s')$.

4.2.2. Complexity. In this part, we will analyze the performance of our proposal. Before that, we first define some parameters which are mentioned as follows:

m: the output of hash function which is an m-bit string,

n: the length of hash chain,

i: the number of secret shadows in SRHC-BL,

M: the computation consumption of the hash function,

N: the computation consumption of the union operation,

R, R_A, $R_{A'}$: the computation consumptions of generating a random number in RHC, ERHC, and SUHC (or SRHC), respectively,

A, A': the computation consumption of obtaining one bit from a random number by hard core predicate in SUHC and SRHC, respectively,

C, I, P: the computation consumption of obtaining Shares(S, F), computing the shadows I_i, and picking secret shadows s_i from I_i in SRHC-BL successively,

E: the computation consumption of XOR,

L_M: the communication or memory consumption of m (bit),

L_s: the communication or memory consumption of the seed of hash chain,

L_r: the communication or memory consumption of the generated random number,

L_I: the communication or memory consumption of shadows I_i in SRHC-BL,

L_e: the communication or memory consumption of the secret shadows s_i in SRHC-BL.

Then, we compare the computation, communication, and storage cost of our scheme SRHC-BL with the current schemes RHC, ERHC, SUHC, and SRHC. The comparison results are shown as follows.

RHC is as follows:

Computation:

$$\frac{1}{2}\left(m^2 + 9m\right) M + 2mR. \tag{9}$$

Communication:

$$2L_M + 3m \times L_r + 6m - 2. \tag{10}$$

Storage:

$$2L_s + 3L_r + (m + 6) \times L_M + m. \tag{11}$$

SUHC is as follows:

Computation:

$$\frac{1}{2}\left(m^2 + 12m - 2\right) \cdot M + mR_A + mA. \tag{12}$$

Communication:

$$(6m - 1) \cdot L_M + 2m \cdot L_r. \tag{13}$$

Storage:

$$2\left(L_s + L_r\right) + (m + 6) L_M + m. \tag{14}$$

ERHC is as follows:

Computation:

$$\frac{1}{2}\left(n^2 + 5n + 5m + 5\left\lfloor \log_2 m \right\rfloor + 5\right) \cdot M + 2\left(m + \left\lfloor \log_2 m \right\rfloor + 1\right) \cdot R + 2N. \tag{15}$$

Communication:

$$2\left(n + m + \left\lfloor \log_2 m \right\rfloor + 1\right) \cdot L_M + \left(m + \left\lfloor \log_2 m \right\rfloor + 1\right) \cdot L_r. \tag{16}$$

Storage:

$$\left(n + 3m + \left\lfloor \log_2 m \right\rfloor + 1\right) \cdot L_M + m \times \left(1 + 2L_r\right). \tag{17}$$

SRHC is as follows:

Computation:

$$\frac{1}{2}\left(m^2 + 11m - 2\right) \cdot M + m \cdot R_{A'} + mA'. \tag{18}$$

Communication:

$$4m \cdot L_M + 2m \cdot L_r. \tag{19}$$

Storage:

$$2L_s + 3L_r + (3 + m) L_M + m. \tag{20}$$

SRHC-BL is as follows:

Computation:

$$\frac{1}{2}\left(n^2 + 7n - 2\right) M + 2nE + n(I + P) + C. \tag{21}$$

Communication:

$$(4n - 2) \cdot L_M + 2n \cdot L_I. \tag{22}$$

Storage:

$$(n + 3) \cdot L_M + n \cdot L_I + i \cdot L_e + 2m. \tag{23}$$

For simplicity, we assumed that $m \approx n$, $R \approx R_A \approx R_{A'}$, $A \approx A'$, $M > N$, $M > C > I > P$, and $L_M \gg L_s \approx L_r \approx L_I > L_c$, so that it is easy to know the performance of our SRHC-BL relative to RHC, ERHC, SUHC, and SRHC. Through comparison, we can draw the following conclusion: the consumption of SRHC-BL in the initialization phase is much less than other schemes, while, in the phase of key distribution and authentication, SRHC-BL's consumptions of communication and storage are a little more than SRHC's but much less than RHC's, ERHC's, and SUHC's.

5. Related Work

5.1. Improved μTESLA Protocol. Many hybrid broadcast authentication protocols have been proposed. Reference [14] proposed a broadcast authentication protocol with Bloom Filter compression to mainly reduce error rate of data broadcasting. Reference [15] introduced a multiuser broadcast authentication protocol to synchronously meet the requirements of multiuser. A lightweight secure authentication protocol was proposed in [16], which mainly focuses on the storage performance optimization. Reference [17] is a μTESLA-like scheme based on symmetric keys, but the signature takes a large storage cost. A secure protocol named GPLD (Global Partition, Local Diffusion) was proposed in [18]; this scheme based on the symmetric encryption system and the geographical location information allows the different multicast group to exist in wireless sensor networks, and nodes can also act as the broadcast source and relay. On the basis of [18, 19] a broadcast authentication scheme based on users, which achieves the promising security, scalability, and performance, was proposed. Reference [13] proposes an enhanced broadcast authentication protocol based on multilevel μTESLA, however, whose overhead has not achieved the satisfactory efficiency. Reference [20] put forward a broadcast authentication scheme with the Merkle tree; although it can

effectively resist the DoS attacks, the authentication delay seems to be inappropriate for most applications. Taking the tolerance of data loss into account, [21] presents a link-layer packet recovery algorithm which improves the reliability and minimizes the latency.

So we can see that μTESLA protocol and its improved protocols are the mainstream of broadcast authentication protocol research in wireless sensor networks.

5.2. Reinitializable Hash Chain. Hash function has the characteristics of one-wayness and high computational efficiency. Therefore, the hash chain mechanism has been widely used into many encryption applications and services. Furthermore, the length of the hash chain is limited, which makes it difficult to meet the requirement of sustainability. And extending the length of the hash chain is difficult because a secure channel established through other encryption mechanisms is needed, and a large overhead is required.

To solve this contradiction, researchers have proposed some hash chain schemes. Goyal introduced the reinitializable hash chain (RHC) scheme with the idea that a fire-new RHC will be regenerated safely and undeniably when the old RHC is exhausted. On the basis of RHC, [22] put forward the elegant reinitializable hash chain (ERHC) scheme, which uses the one-way hash function to regenerate the hash chain safely and infinitely instead of using the public key mechanism. However, due to the publication part of S_U to authentication for the next seed of hash chain, it is likely to be susceptible to the chosen plaintext attack. Reference [23] proposed the self-updating hash chain (SUHC) scheme based on the hard core predicate algorithm. The solution of SUHC is that the sender distributes the first chain's every key value with one bit in the seed of second. In such a way, while the first one is exhausted, the receiver would receive all bits of second chain's seed. On the basis of [23, 24] the self-renewal hash chain (SRHC) scheme was proposed. The main difference between the above two schemes is the generation method of the random numbers. The security distributions of the seed of SUHC and SRHC rely on the security distribution of k random numbers, where k denotes the length of chain. Furthermore, these two schemes require all the received random numbers to satisfy integrity and inevitability. And then the seed of a new chain can be reconstructed. However, both of them have given up the original fault tolerance of hash chain. Based on SUHC, [25] put forward a novel self-updating hash chain (NSUHC) scheme; afterwards, according to NSUHC, [26] proposed a new self-updating hash chain based on erasure coding (SUHC-EC). In the former scheme, the seed of a new hash chain is transformed from k-dimensional to n-dimensional ($k < n$) and the latter one is transformed from one-dimensional to n-dimensional. Therefore, two schemes select one of the n random values to release without repeating. The new seed can be resumed after k times. These two schemes seem to realize the renewable hash chain, but actually there is no difference from the conventional hash chain. Reference [27] proposed a new self-updating hash chain based on fair exchange idea (SRHC-FEI); this scheme uses one-time signature key to encrypt the first bit of the seed of a new hash chain in transmission when releasing the new hash value each time. It can enhance the security and fairness, but it inevitably increases the system time delay. After analysis, we can see that this scheme is also an enhanced scheme more than a strict hash chain renewable construction scheme.

From the analysis of the above typical schemes we can see that they all transform every bit of the new chain's seed into a random number and make the security of the new seed dependent on the security of distributed random numbers. Besides, they can successfully regenerate the new seed only when they receive all the random numbers correctly. As a result, they all weaken the security and increase the consumptions for reinitialization. On the other hand, NSUHC and SUHC-EC only expand the dimension of the seed of a new hash chain, but compared with RHC and ERHC and so forth, they increase the chance of encountering the man-in-the-middle attack. Above all, from a perspective of application of a hash chain, only RHC, ERHC, SUHC, and SRHC belong to the renewable construction scheme of hash chain.

6. Conclusion

This paper proposes a novel secret key release scheme based on the data flow, which addresses some problems of traditional key release schemes based on the fixed time interval, effectively improves the efficiency of the utilization of keys, prolongs the life cycle of hash chain, and reduces the network communication overhead and computational cost.

Moreover, we consider the scenario that when the number of packets using the same key to authenticate is greater than the threshold N_{str}, it may disable some packets to get a timely authentication and thus results in the loss of data. Also, the probability of chosen plaintext attack will be increased. To solve these problems, we introduce the flow threshold mechanism to prevent the attacks and enhance network security as well.

After that we put forward a new renewable hash chain based on Benaloh-Leichter SSS (SRHC-BL). The renewable process can be executed infinitely. And we have theoretically proved that SRHC-BL has better performance on integrity, confidentiality, and nonrepudiation by adopting the delay disclosure and one-wayness. In addition, our scheme can also tolerate message loss or fault due to the property of the shadows in Benaloh-Leichter SSS. Furthermore, the dual authentication and transformed secret shadows enable our scheme to have higher security than other schemes. Finally, the analysis of complexity has proved that SRHC-BL has less consumption than those typical schemes.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by grants from the National Natural Science Foundation of China (nos. 61373138 and 61272422), the Key Research and Development Program of Jiangsu Province (Social Development Program, no. BE2015702), the Natural Science Foundation of Jiangsu Province (no. BK20151511), Postdoctoral Foundation (nos. 2015M570468 and 2016T90485), the Sixth Talent Peaks Project of Jiangsu Province (no. DZXX-017), the Fund of Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks (WSNLBZY201516), and Science and Technology Innovation Fund for Postgraduate Education of Jiangsu Province (no. KYLX15_0853).

References

[1] L. Xu, M. Wen, and J. Li, "A bidirectional broadcasting authentication scheme for wireless sensor networks," in *Proceedings of the IEEE Conference on Collaboration and Internet Computing (CIC '15)*, pp. 200–204, Hangzhou, China, October 2015.

[2] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "SPINS: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[3] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Proceedings of the Network and Distributed System Security Symposium (NDSS '01)*, pp. 35–46, San Diego, Calif, USA, February 2001.

[4] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P '00)*, pp. 56–73, Berkeley, Calif, USA, May 2000.

[5] M. H. Eldefrawy, M. K. Khan, and K. Alghathbar, "One-time password system with infinite nested Hash chains," in *Security Technology, Disaster Recovery and Business Continuity*, pp. 161–170, Springer, Berlin, Germany, 2010.

[6] S.-H. Ou, C.-H. Lee, V. S. Somayazulu, Y.-K. Chen, and S.-Y. Chien, "On-line multi-view video summarization for wireless video sensor network," *IEEE Journal on Selected Topics in Signal Processing*, vol. 9, no. 1, pp. 165–179, 2015.

[7] G. Oligeri, S. Chessa, R. Di Pietro, and G. Giunta, "Robust and efficient authentication of video stream broadcasting," *ACM Transactions on Information and System Security*, vol. 14, no. 1, article 5, pp. 1–25, 2011.

[8] A. Huszti, "Anonymous multi-vendor micropayment scheme based on bilinear maps," in *Proceedings of the International Conference on Information Society (i-Society '14)*, pp. 25–30, IEEE, London, UK, November 2014.

[9] X. Zhang and J. Wang, "An efficient key management scheme in hierarchical wireless sensor networks," in *Proceedings of the International Conference on Computing, Communication and Security (ICCCS '15)*, pp. 1–7, Pamplemousses, Mauritius, December 2015.

[10] D. Liu and P. Ning, "Multilevel μTESLA," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 4, pp. 800–836, 2004.

[11] H. Jiang, J. Zhai, S. K. Wahba, B. Mazumder, and J. O. Hallstrom, "Fast distributed simulation of sensor networks using optimistic synchronization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 2888–2898, 2014.

[12] J. H. Lee, L. H. Kim, and T. Y. Kwon, "FlexiCast: energy-efficient software integrity checks to build secure industrial wireless active sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 6–14, 2016.

[13] X. Li, N. Ruan, F. Wu, J. Li, and M. Li, "Efficient and enhanced broadcast authentication protocols based on multilevel μTESLA," in *Proceedings of the 33rd IEEE International Performance Computing and Communications Conference (IPCCC '14)*, pp. 1–8, Austin, Tex, USA, December 2014.

[14] Y.-S. Chen, I.-L. Lin, C.-L. Lei, and Y.-H. Liao, "Broadcast authentication in sensor networks using compressed bloom filters," in *Distributed Computing in Sensor Systems*, pp. 9–111, Springer, Berlin, Germany, 2008.

[15] K. Ren, S. Yu, W. Lou, and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 4554–4564, 2009.

[16] M. Sharifi, S. S. Kashi, and S. P. Ardakani, "LAP: a lightweight authentication protocol for smart dust wireless sensor networks," in *Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS '09)*, pp. 258–265, Baltimore, Md, USA, May 2009.

[17] C. Benzaid, S. Medjadba, A. Al-Nemrat, and N. Badache, "Accelerated verification of an ID-based signature scheme for broadcast authentication in wireless sensor networks," in *Proceedings of the IEEE 15th International Conference on Computational Science and Engineering (CSE '12)*, pp. 633–639, Nicosia, Cyprus, December 2012.

[18] K. Ren, W. Lou, B. Zhu, and S. Jajodia, "Secure and efficient multicast in wireless sensor networks allowing ad hoc group formation," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, pp. 2018–2029, 2009.

[19] X. Cao, W. Kou, L. Dang, and B. Zhao, "IMBAS: identity-based multi-user broadcast authentication in wireless sensor networks," *Computer Communications*, vol. 31, no. 4, pp. 659–667, 2008.

[20] R. D. Pietro, F. Martinelli, and N. V. Verde, "Broadcast authentication for resource constrained devices: a major pitfall and some solutions," in *Proceedings of the 31st IEEE International Symposium on Reliable Distributed Systems (SRDS '12)*, pp. 213–218, Irvine, Calif, USA, October 2012.

[21] C. Qiu, H. Shen, S. Soltani, K. Sapra, H. Jiang, and J. O. Hallstrom, "CEDAR: a low-latency and distributed strategy for packet recovery in wireless networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 5, pp. 1514–1527, 2015.

[22] Y.-C. Zhao and D.-B. Li, "An elegant construction of re-initializable hash chains," *Journal of Electronics & Information Technology*, vol. 28, no. 9, pp. 1717–1720, 2006.

[23] H. Zhang and Y. Zhu, "Self-updating hash chains and their implementations," in *Web Information Systems-WISE 2006*, pp. 387–397, Springer, Berlin, Germany, 2006.

[24] H. Zhang, X. Li, and R. Ren, "A novel self-renewal hash chain and its implementation," in *Proceedings of the 5th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC '08)*, pp. 144–149, Shanghai, China, December 2008.

[25] M.-Q. Zhang, B. Dong, and X.-Y. Yang, "A new self-updating hash chain structure scheme," in *Proceedings of the International Conference on Computational Intelligence and Security (CIS '09)*, pp. 315–318, Beijing, China, December 2009.

[26] Z. Wei, "Self-updating hash chains based on erasure coding," in *Proceedings of the International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE '10)*, pp. 173–175, Changchun, China, August 2010.

[27] X.-Y. Yang, J.-J. Wang, J.-Y. Chen, and X.-Z. Pan, "A self-renewal hash chain scheme based on fair exchange idea(SRHC-FEI)," in *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT '10)*, pp. 152–156, Chengdu, China, July 2010.