ECM-IBS

A Chebyshev map-based broadcast authentication for Wireless Sensor Network

ECM-IBS: A Chebyshev map-based Broadcast Authentication for Wireless Sensor Networks

Yuling Luo[1], Yunqi Liu[1], Junxiu Liu[1*], Xue Ouyang[1], Yi Cao[2], Xuemei Ding[3, 4]
[1]*School of Electronic Engineering, Guangxi Normal University, Guilin, China, 541004*
[2]*Business School, University of Edinburgh, Edinburgh, UK, EH8 9JS*
[3]*School of Computing, Engineering and Intelligent Systems, Ulster University, UK, BT48 7JL*
[4]*College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China, 350108*
[*]*liujunxiu@mailbox.gxnu.edu.cn*

Received (to be inserted by publisher)

As a basic and crucial security requirement in Wireless Sensor Networks (WSNs), authentication is generally used to prevent various communication attacks such as Denial-of-Service (DoS) attack. A novel broadcast authentication framework is proposed in this paper, where an Identity-Based Signature schemes by using the Extended Chaotic Maps (ECM-IBS) is designed to authenticate all broadcast messages and specifically, a chaos-based hash function is used for message authentication in the WSNs. It is implemented using a WSN hardware device of CC2530 and its performance is analyzed under various methods. Performance analysis and experimental results show that the proposed ECM-IBS scheme has a quick signature generation speed, low energy consumption and short verification delay, and can be applied to WSN applications.

Keywords: Wireless Sensor Networks (WSNs), chaos, broadcast authentication, hash function.

1. Introduction

Wireless sensor networks (WSNs), consisting of plenty number of low-cost micro sensor nodes [Akyildiz *et al.*, 2002; Zhang *et al.*, 2018], are often deployed in unattended fields, i.e., only the legitimate users could access the network and get the data. As the WSNs have limited hardware resources, power supply capacity, communication bandwidth and other characteristics, the broadcast authentication becomes a basic security services in a resource-constrained environment before accessing data from the sensor nodes. In general, the identity authentication is based on three parameters: what you know (password based), what you possess (smart cards), and what you are (biometrics) [Cao *et al.*, 2016]. For the WSNs, the authentication is mostly based upon the first one (i.e., what you know). Besides, the password-based scheme must be closely related to the secret keys and any other cryptographic material that is usually used by identities to prove their uniqueness. Then the generation and distribution of the secret keys are the most important factors for all security services. Thus it should be considered for the design of authentication scheme in WSNs, especially for the sensor nodes in WSNs. It is known that the broadcast authentication technology for WSNs is mainly divided into symmetric key cryptography (SKC) and public key cryptography (PKC).

The TESLA is a widely used identity authentication protocol in the Internet. It is based on SKC and can enhance the speed of broadcast authentication and the intensity of computation. However, due to the limited energy and limited communication bandwidth, the TESLA scheme cannot be directly applied to the

WSNs. So a lightweight TESLA scheme, namely μTESLA [Perrig *et al.*, 2002], is proposed for broadcast authentication in the WSNs. The asymmetry of μTESLA is produced by the delayed transmission of symmetric keys, which can be used to produce effective broadcast authentication. Subsequently, other broadcast authentication schemes based on μTESLA have been proposed [Drissi & Gu, 2006; Liu & Ning, 2004; Liu *et al.*, 2005]. Although these SKC-based authentication methods have many advantages, such as less consumption of resources and communication, the possibility of attack is not taken into account, which will lead to the delay of the release of asymmetric keys and the failure of network transmission. Because broadcast authentication is a one-way authentication, so asymmetric cryptosystem is more suitable for broadcast authentication than the symmetric cryptography. The RSA and Elliptic Curve Cryptography (ECC) are widely used for PKC, and the PKC can be applied to WSNs after the improvement and optimization. For instance, Ren et al. [Ren *et al.*, 2007] and Du et al. [Du *et al.*, 2008] have proposed the PKC-based broadcast authentication for WSNs by using ECC. Besides, the public key of nodes can be easily calculated according to the identity information of wireless sensor nodes. Therefore, it can eliminate the need of the public-key certificates. The approaches of the authentication and key-exchange protocols which is based on Chebyshev chaotic map has been proposed in [Dariush & Morteza, 2018; Jiang *et al.*, 2016; Hsu & Lin, 2013]. Compared with the ECC-based and RSA-based schemes, the results of the above method showed that the Chebyshev chaotic map-based schemes are more suitable for devices due to the limited battery life and small computation power, because the signature computation requires smaller key size, and the semi-group property of Chebyshev chaotic map provides faster computation speed [Chatterjee *et al.*, 2018].

Chaotic systems have some inherent cryptographic characteristics, such as pseudo randomness and sensitivity to initial conditions which means that any tiny change or perturbation can cause the current trajectory a greatly different behavior [Wang *et al.*, 2016; Deng *et al.*, 2017; Liu *et al.*, 2015a, 2016; Attaullah & Shah, 2015; Li *et al.*, 2019], which is crucial and applicable for the chaos-based cryptosystems [Liu *et al.*, 2015b; Azzaz *et al.*, 2013; Hua *et al.*, 2018a; Li *et al.*, 2018a] and secure communications [Fontes & Eisencraft, 2016]. Specifically, the one-dimensional Logistic map is used in cryptography for the first time in [Matthews, 1989], which laid a foundation for the application of chaotic system in cryptography. Then, an improved method for the chaotic system in [Matthews, 1989] is presented, and it is applied in chaos-based cryptosystems [Habutsu *et al.*, 1991]. After that, a chaos-based image encryption is first introduced in [Fridrich, 1998]. Since then, with the development of chaotic cryptography, many chaos-based encryption methods have been proposed. In order to obtain more complex chaotic systems, researchers have begun to explore the chaotic map from low-dimensional to high-dimensional. Among them,the Chebyshev chaotic map has the semi-group property which make it is suitable for the design of digital signature based on public key cryptography. Specifically, a secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment has been proposed in [Chatterjee *et al.*, 2018], which is mainly about mutual authentication between users and servers. The authentication process is used of symmetric encryption and the semi-group characteristic of the Chebyshev map. Considering the above discussion, an identity-based signature scheme by using the Extended Chaotic Maps (ECM-IBS) to authenticate all broadcast messages is proposed in this work. Different from the method in [Chatterjee *et al.*, 2018], the aim of this work is to design a digital signature to realise broadcast authentication by using the multiple properties of the Chebyshev map. Specifically, a chaos-based hash function is constructed and applied in the whole ECM-IBS, and the corresponding test is executed and analysed. The simulation analysis demonstrates the proposed ECM-IBS scheme has a high security and efficiency, and the hardware implementation (in CC2530 of the WSN hardware platforms) illustrates it has a good practical feasibility because of requiring less storage cost, computation and communication overhead. Both of them indicate that the ECM-IBS scheme is suitable for the WSN application.

The rest of the paper is organized as follows. Section 2.1 gives the preliminaries work of Chebyshev polynomials. In Section 3, a chaos-based broadcast authentication scheme named ECM-IBS is presented. In Section 4, the ECM-IBS scheme is implemented, and the security performance and availability of the ECM-IBS are analyzed. Section 5 gives the conclusion.

2. Preliminary

2.1. *The first-kind Chebyshev polynomials T_n*

Due to some good properties, the first-kind Chebyshev polynomials was found to be used for the application of the public-key cryptosystems [Kocarev & Tasev, 2003], key agreement protocols [Qin *et al.*, 2010; Yan *et al.*, 2015; Bergamo *et al.*, 2005] and the authentication schemes [Bergamo *et al.*, 2005].

Definition 1: The n-dimensional Chebyshev polynomial is given by

$$T_n(x) = \cos(n\theta) \quad when \quad x = \cos(\theta), \tag{1}$$

where $x \in [-1, 1]$, and the parameter $\theta \in [0, \pi]$.

Definition 2: Let n be an integer, the first-kind Chebyshev polynomial $T_n(x) : [-1, 1] \to [-1, 1]$ can be defined by

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x),\ n \geqslant 2, \tag{2}$$

where the initial conditions are $T_0(x) = 1$, and $T_1(x) = x$, and the common expression of the first-kind Chebyshev polynomial would be

$$\begin{cases} T_2(x) = 2x^2 - 1, \\ T_3(x) = 4x^3 - 3x, \\ T_4(x) = 8x^4 - 8x^2 + 1, \\ T_5(x) = 16x^5 - 20x^3 + 5x. \end{cases} \tag{3}$$

Definition 3: Semi-group property of the Chebyshev polynomials can be described by

$$\begin{aligned} T_r(T_s(x)) &= \cos(r \cdot \cos^{-1}(\cos(s \cdot \cos^{-1}(x)))) \\ &= \cos(r \cdot s \cdot \cos^{-1}(x)) \\ &= T_{sr}(x) \\ &= T_s(T_r(x)), \end{aligned} \tag{4}$$

where r and s are two positive integers, and $x \in [-1, 1]$. Kocarev et al. [Kocarev & Tasev, 2003] extended the definition of the Chebyshev polynomial from the real domain to the finite field. For instance, in the finite field of $x \in Z_P$ (P is prime), the Chebyshev polynomial can be defined by

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x)) \bmod P, n \geq 2. \tag{5}$$

As for the extended Chebyshev polynomial, $T_r(T_s(x)) \equiv T_{rs}(x) \equiv T_{sr}(x) \equiv T_s(T_r(x)) \bmod P$ is still established.

Definition 4: The discrete logarithm problem (DLP): if the parameters x, y are given and they are satisfied by $T_r(x) = y$, it is difficult to calculate the integer r.

Definition 5: The Diffie-Hellman problem (DHP): if the parameters x, $T_r(x)$ and $T_s(x)$ are all given, it is difficult to calculate the integer $T_{rs}(x)$.

It is generally believed that the security of extended Chebyshev maps relies on the difficulties of the above computational problems which are not solvable in polynomial time.

2.2. *Counteracting dynamical degradation and discretization of digital chaotic system*

Theoretically, chaotic system or chaotic map have ideal complex dynamics. However, because of the finite precision of simulation software and digital devices during implementations, chaotic systems often undergo dynamical degradation, which hinders the further application of digital chaotic systems in many fields. Therefore, several methods have been proposed to address this problem [Li *et al.*, 2018b; Hua *et al.*, 2018b, 2019; Liu *et al.*, 2017].

In [Liu *et al.*, 2017], a new method has been introduced to counteract the dynamical degradation of the digital chaos. Specifically, the continuous Chen system is used to perturb both the inputs and parameters of the Chebyshev map to minimise the chaotic degradation phenomenon under the finite precision. Similarly,

Chen system is firstly discretized by using Euler algorithm for the digital implementation to get the digital system. For example, the digital Chen system is described by

$$\begin{cases} x_{i+1} = (1 - a \cdot T) \cdot x_i + a \cdot T \cdot y_i, \\ y_{i+1} = T \cdot (c - a) \cdot x_i + (1 + T \cdot c) \cdot y_i - T \cdot x_i \cdot z_i, \\ z_{i+1} = T \cdot x_i \cdot y_i + (1 - T \cdot b) \cdot z_i, \end{cases} \tag{6}$$

where $a = 35$, $b = 3$, $c = 28$ and the sampling interval is defined as 2ms (i.e. $T = 0.02$). Moreover, the outputs x_i, y_i of the digital Chen system are used to disturb the input and parameter n of the Chebyshev map. Thus the improved digital chaotic system can be described by

$$\bar{x}_{i+1} = \cos(F(\bar{x}_i, y_i) \arccos(H(\bar{x}_i, x_i))), \tag{7}$$

where the perturbing functions H and F are designed by

$$\begin{cases} H(\bar{x}_i, x_i) = \mathrm{mod}(\bar{x}_i + x_i, 1) \times 2 - 1, \\ F(\bar{x}_i, y_i) = \mathrm{mod}(\bar{x}_i + y_i, 1) + 6. \end{cases} \tag{8}$$

Owing to the limited accuracy, limited operation, limited memory and low power of the WSN notes, it becomes difficult to directly implement the floating-point arithmetic, division and other complicated computing. Then the above system is firstly optimized to make it suitable for the implementation of the WSN nodes, that is, both sides of the equation (6) are multiplied by g^2 to get

$$\begin{cases} x_{i+1} \cdot g^2 = (1 - a \cdot T) \cdot x_i \cdot g^2 + a \cdot T \cdot y_i \cdot g^2, \\ y_{i+1} \cdot g^2 = T \cdot (c - a) \cdot x_i \cdot g^2 + (1 + T \cdot c) \cdot y_i \cdot g^2 - T \cdot x_i \cdot z_i \cdot g^2, \\ z_{i+1} \cdot g^2 = T \cdot x_i \cdot y_i \cdot g^2 + (1 - T \cdot b) \cdot z_i \cdot g^2. \end{cases} \tag{9}$$

In order to further optimize Eq. (9), assume

$$\begin{cases} \ddot{x}_i = x_i / 80 \cdot g, \\ \ddot{x}_{i+1} = x_{i+1} / 80 \cdot g, \\ \ddot{y}_i = y_i / 80 \cdot g, \\ \ddot{y}_{i+1} = y_{i+1} / 80 \cdot g, \\ \ddot{z}_i = z_i / 80 \cdot g, \\ \ddot{z}_{i+1} = z_{i+1} / 80 \cdot g. \end{cases} \tag{10}$$

Combining Eq. (9) and Eq. (10), the renewed digital Chen system can be expressed by

$$\begin{cases} \ddot{x}_{i+1} = (1 - a \cdot T) \cdot \ddot{x}_i + a \cdot T \cdot \ddot{y}_i, \\ \ddot{y}_{i+1} = T \cdot (c - a) \cdot \ddot{x}_i + (1 + T \cdot c) \cdot \ddot{y}_i - \frac{80T}{g} \cdot \ddot{x}_i \cdot \ddot{z}_i, \\ \ddot{z}_{i+1} = \frac{80T}{g} \cdot \ddot{x}_i \cdot \ddot{y}_i + (1 - T \cdot b) \cdot \ddot{z}_i. \end{cases} \tag{11}$$

In order to remove the floating-point operation, Eq. (11) can be further optimized by

$$\begin{cases} \ddot{x}_{i+1} = (930 \cdot \ddot{x}_i + 70 \cdot \ddot{y}_i)/1000, \\ \ddot{y}_{i+1} = (14 \cdot \ddot{x}_i + 1056 \cdot \ddot{y}_i - 49 \cdot \ddot{x}_i \cdot \ddot{z}_i/10000)/1000, \\ \ddot{z}_{i+1} = (49 \cdot \ddot{x}_i \cdot \ddot{y}_i/10000 + 994 \cdot \ddot{z}_i)/1000. \end{cases} \tag{12}$$

Since the range of x_i, y_i, z_i of Chen system is in $[-80, 80]$, then the range of $\ddot{x}_i$, $\ddot{y}_i$, $\ddot{z}_i$ will be in $[-g, g]$, where $g = 2^S$ (S denotes the computer word size). Similarly, it is also difficult to calculate *cos* and *arccos* operations of the Chebyshev map for WSN nodes, then according to the specific introduction of Chebyshev polynomials as shown in Section 2.1, the Eq. (2) can be supposed to quantify by

$$\begin{cases} T_n(x) = \bar{T}_n(x)/a - 1, \\ T_{n-1}(x) = \bar{T}_{n-1}(x)/a - 1, \\ T_{n-2}(x) = \bar{T}_{n-2}(x)/a - 1, \\ x = \bar{x}/a - 1. \end{cases} \tag{13}$$

Then the recurrence relation of the first-kind Chebyshev polynomial can be renewed by

$$\bar{T}_n(x) = (\frac{2\bar{x}}{a} - 2)\bar{T}_{n-1}(x) - \bar{T}_{n-2}(X) - 2\bar{x} + 4a, \tag{14}$$

and the explicit expressions of the first three Chebyshev polynomials are

$$\begin{cases} \bar{T}_1(x) = \bar{x}, \\ \bar{T}_2(x) = \frac{2}{a}\bar{x}^2 - 4\bar{x} + 2a, \\ \bar{T}_3(x) = \frac{4}{a^2}\bar{x}^3 - \frac{12}{a}\bar{x}^2 + 9\bar{x}. \end{cases} \tag{15}$$

Then, because the range of $T_n(x)$ is in $[-1, 1]$, so the range of $\bar{T}_n(x)$ will be in $[0, 2a]$. For example, when the word size of the processor is S, i.e., $a = 2^{S-1}$, $\bar{T}_n(x) \in 2^S$ can be established which would just represent the unsigned integer with computer word size. Therefore, there will exist the shifting, multiplication, subtraction, complement, addition and other basic operations for the calculation process of $\bar{T}_n(x)$, which would be suitable for the WSN nodes.

As mentioned above, the outputs $\ddot{x}_i$, $\ddot{y}_i$ of the digital Chen system are used to disturb the parameter n and the input of the improved Chebyshev map, respectively. Specifically, the parameter n is obtained by checking the range of $\ddot{x}_i$, which allows the Chebyshev map have different forms. Thus, the chaotic system can achieve a better performance even if under a low finite precision. The parameter n and the final output can be expressed by

$$n = \begin{cases} 3, & if \ \ \ddot{x}_i \in (0, 5000] \\ 4, & if \ \ \ddot{x}_i \in (-5000, 0] \\ 5, & if \ \ \ddot{x}_i \in [-g, -5000] \bigcup [5000, g] \end{cases} \tag{16}$$

and

$$x_{i+1} = T_n(x_i + (\mathrm{abs}(\ddot{y}_i))^2), \tag{17}$$

where the final output is x_i, and i is the number of iterations. Therefore, the digital chaotic system can have a better performance and faster calculation speed under lower finite precision if it is implemented in the above disturbance and discretization method.

3. Chaos-based broadcast authentication scheme

This section presents a new broadcast authentication and it includes a broadcast authentication scheme that based on the Chebyshev chaotic system and a chaos-based one-way hash function. Specifically, Section 3.1 introduces the overall frame of the broadcast authentication in WSNs. As an important component of the broadcast authentication, one-way hash function can protect information and reduce the length of data in the transmission process. Specifically, the pre-operation of the plain-text is required by conventional hash function, and the length of plain-text after pre-processing will be greater than or equal to 512 bits. Of course, it is not necessary to do this relatively high complexity pre-processing if the communication data is small, and this case is most common in the WSNs. In addition, the WSNs do not require long digest of information, e.g., the digest length of SHA-1 is 128 bits. Based on the above consideration, a novel hash function for the WSNs is designed in Section 3.2, which is based on the improved chaotic system.

Table 1. The notations used in this paper.

Notation	Description
SK_{BS}	The master secret key
PK_{BS}	The public key
M	Message
ID_i	The identities of users
$h(\cdot)$	A secure one-way hash function
T_s	Current system time stamps
S_i	Signature
$\|$	The message concatenation operation
D_{ID_i}	The private key

3.1. *Design of the broadcast authentication scheme*

In order to better introduce the scheme, some notations which could be used for the broadcast authentication are firstly indicated in Table 1. Besides, the proposed broadcast authentication phase can be divided into five parts which is shown in Fig. 1.

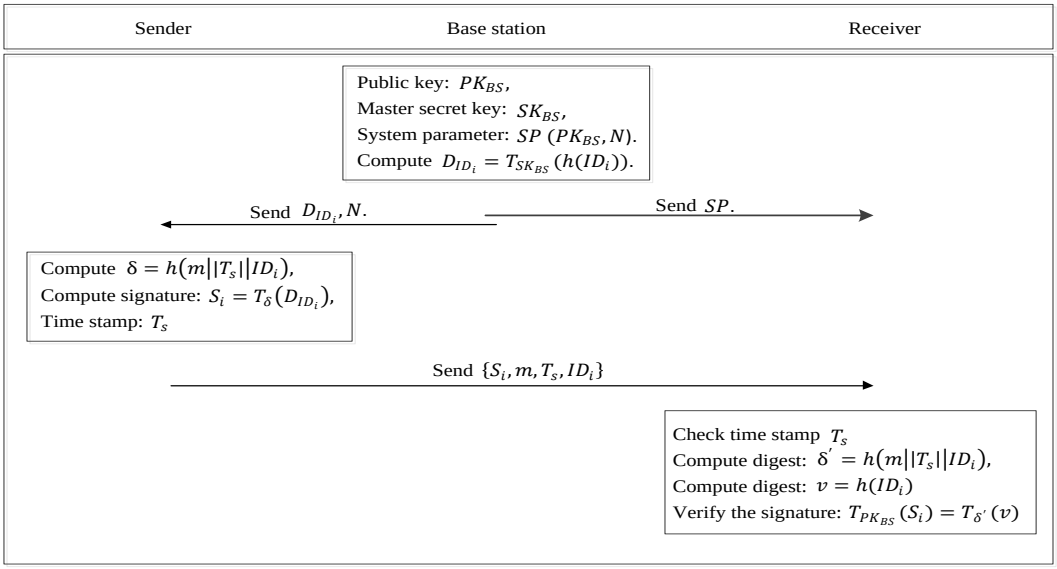

Fig. 1. Broadcast authentication phase.

System Initialization: as a private key generator, the base station is safe and trusted by default, and it is in charge of the initialization of the system. Specifically, randomly select two different large prime numbers P and Q which must have similar bits and compute $N = P \cdot Q$ and $L = (P^2 - 1)(Q^2 - 1)$. Then select the base stations secret key SK_{BS} $(0 < SK_{BS} < N)$, and compute the corresponding public key PK_{BS} to satisfy the qualification of $SK_{BS} \cdot PK_{BS} \equiv 1\ (mod\ L)$, where the master secret key SK_{BS} is stored in the base station and the system parameter $SP\ (PK_{BS}, N)$ is public.

Key generation: All sensor nodes that in the WSNs should have their own private key. As for the sensor node i, its private key $D_{ID_i} = T_{SK_{BS}}(h(ID_i))$, where the SK_{BS} is the master secret key, $h(\cdot)$ is a secure hash function, and the ID_i is the identity information of the sensor node. And each sensor node i must stores the information of $\{ID_i, D_{ID_i}, SP\}$.

Message broadcast: When any one sensor node i senses an event that is required to quickly report, the node i firstly must computes the message digest through $\delta = h(m \parallel T_s \parallel ID_i)$, where T_s is the current time stamp, and then computes the signature through $S_i = T_\delta(D_{ID_i})(mod\ N)$. In this method, only the nodes which own the private key from the base station can sign a message. Therefore, the final broadcast message must contain the message m, time stamp T_s, identity of the sensor node ID_i and the signature S_i, i.e., the final broadcast message is consistence of $\{S_i, m, T_s, ID_i\}$.

Authentication: When the node receives the broadcast messages, it will firstly check the time stamp T_s. If the T_s is correct, it indicates that the message is fresh and the node will perform signature verification on the fresh message; otherwise, the message will be discarded. This can be a good defence against replay attacks. Specifically, the process of signature verification S_i for the receiver can be detailed as follows. For example, the digest of the node can be computed by

$$\delta' = h(m \parallel T_s \parallel ID_i), \tag{18}$$

and

$$v = h(ID_i). \tag{19}$$

When the equation of $T_{PK_{BS}}(S_i) = v$ is established, the receiver will accept this message; otherwise it will be discarded. Specifically, $T_{PK_{BS}}(S_i)$ can be calculated by

$$\begin{aligned} T_{PK_{BS}}(S_i) &= T_{PK_{BS}}(T_\delta(D_{ID_i})) \\ &= T_{PK_{BS}}(T_\delta(T_{SK_{BS}}(h(ID_i)))) \\ &= T_{PK_{BS}\cdot SK_{BS}}(T_\delta(h(ID_i))) \\ &= T_{k(P^2-1)(Q^2-1)+1}(T_\delta(h(ID_i))) \\ &= T_\delta(h(ID_i)) \\ &= T_{\delta'}(v). \end{aligned} \tag{20}$$

As described above, any tiny change of the broadcast message will result in the failure of signature verification in the transmission process, and it guarantees the integrity of the message. In contrast, the signature information from the sensor node will not be verified if the sensor node does not own the private key from the base station. Therefore, every sensor node can verify the received broadcast messages through this method, and it can simultaneously speed up the dissemination of information on the premise of ensuring the security of information sources.

Sender revocation: when it is necessary to undo a stolen sensor node i, other nodes in the network obtain the ID_i through the broadcast of the base station, and these sensor nodes store this ID_i. In the later period, if anyone sensor node receives the message which contains this ID_i, this sensor node only need to reject this message instead of the authentication process. Specially, the situation that only a few sensor nodes in WSNs can be compromised by the adversary is assumed. Because if the greater part of the sensor nodes been intercepted by the opponents, the whole system will be in danger. Therefore, in addition to malicious nodes, other legitimate sensor nodes in the entire sensor network can periodically update system parameters and keys via the base station. However, this kind of update will lead to a high cost for the entire system. Therefore, the sensor node that is attacked or damaged is removed from the entire network. In theory, this is an effective solution.

3.2. *One-way hash function for WSN*

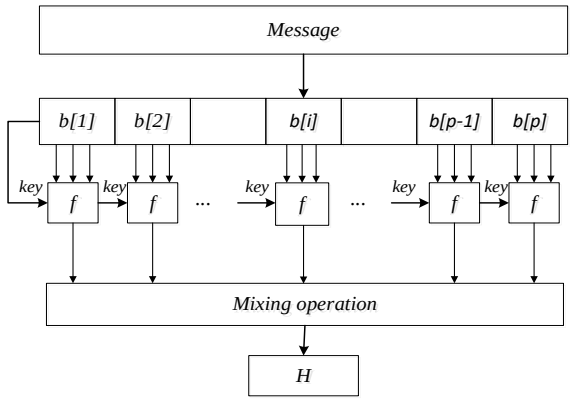

Fig. 2. The whole structure of hash function.

It is well known that even though The chaotic systems degrade slowly under high finite precision, it will be degraded dramatically if they are implemented on devices with low finite precision, such as sensor nodes. These dynamical degradation, i.e., short period phenomenon and strong correlations etc., can directly affect the application in chaos-based cryptography. Therefore, a novel method to improve chaotic degradation is proposed in this paper, and it can guarantee there still be a better performance for the digital implementation under low precision. Besides, in order to reduce the computational overhead, the chaotic system can be performed with integer operations which is described in Section 2.2. As discussed

before, the proposed chaotic system is suitable for the WSN nodes, and it can be used to design one-way hash function as well. Then a hash function is given by

$$H = H(M), \tag{21}$$

where M is a variable length message, $H()$ is a nonlinear function, and the H is a hash value with a fixed length. The whole design structure of the proposed one-way hash function is shown in Fig. 2. Specifically, a message with the arbitrary length is firstly divided into several blocks, and the length of each block is set as a character due to the short message length of the WSNs. Then each block is synchronously processed through a secure hash function to get corresponding outputs. These outputs are finally used to calculate the final hash value. The detailed process is given as follows.

Step 1: Transform each character in the message into the corresponding ASCII code $b[i]$ which is used as the input of each secure hash function f. And $b[i]$ is converted into three values as the three inputs of the Chen system by

$$\begin{cases} \ddot{x}_i = b[i] + i, \\ \ddot{y}_i = b[i] + 2i, \\ \ddot{z}_i = b[i] + 3i. \end{cases} \tag{22}$$

Moreover, the above proposed system is chosen as the mathematical model of hash function.

Step 2: Execute the compression operations by

$$H_i^1 = f(key, \ddot{x}_i, \ddot{y}_i, \ddot{z}_i, 13), \tag{23}$$

$$H_i^2 = f(key, \ddot{x}_i, \ddot{y}_i, \ddot{z}_i, 16), \tag{24}$$

$$H_i^3 = f(key, \ddot{x}_i, \ddot{y}_i, \ddot{z}_i, 19), \tag{25}$$

or

$$H_i^4 = f(key, \ddot{x}_i, \ddot{y}_i, \ddot{z}_i, 22), \tag{26}$$

where the function f has been designed in Section 2.2, key is the previous state value of the improved Chebyshev map in Eq. (17), and $\ddot{x}_i, \ddot{y}_i, \ddot{z}_i$ are three inputs of the Chen system shown in Eq. (12). And the number of 13, 16, 19 and 22 are the iteration numbers of improved Chen map and Chebyshev map.

Step 3: Then execute the following operations by

$$H_1 = H_1^1 \oplus H_2^1 \oplus H_3^1 \cdots \oplus H_i^1, \tag{27}$$

$$H_2 = H_1^2 \oplus H_2^2 \oplus H_3^2 \cdots \oplus H_i^2, \tag{28}$$

$$H_3 = H_1^3 \oplus H_2^3 \oplus H_3^3 \cdots \oplus H_i^3, \tag{29}$$

and

$$H_4 = H_1^4 \oplus H_2^4 \oplus H_3^4 \cdots \oplus H_i^4. \tag{30}$$

Step 4: Get $H = dec2bin(H_1)[1:32] \parallel dec2bin(H_2)[1:32] \parallel dec2bin(H_3)[1:32] \parallel dec2bin(H_4)[1:32]$ as the final 128 bits hash value H. The function $dec2bin(x)$ can convert a decimal number into a binary number represented by string. And the length of the binary number can be controlled to suit different requirements.

4. Implementation and performance analysis

In this paper, the protocols are implemented in Z-stack operating system. The hardware nodes are used of the self-made ZigBee network communication platform, and its main chip is CC2530 (8-bit, 8051 Micro-controller unit, 32 MHz, voltage 3V). Considered of the limited hardware resources, limited power supply capacity, and limited communication bandwidth of WSNs, the degree of the security is usually not needed very strict to some extent. In [Perrig *et al.*, 2002], they only use a 64-bit signature size. Therefore,in this paper the signature size is chosen as 160-bits. The collision resistant hash adopts 64-bits truncation of chaos-based hash function that is designed in Section 3.2.

4.1. *Security analysis*

The security of the proposed ECM-IBS scheme is firstly discussed from the aspects of authentication, verification, integrity and freshness: 1) Authentication. As described above, if the sensor node doesnt own the private key from the base station, the signature information from this sensor node will not be verified in this work, i.e. the authentication is achieved only when the legitimate broadcast senders can sign a message. 2) Verification. Every sensor node can verify a broadcast message from other senders. 3) Integrity. In the transmission process, any tiny change of the broadcast message will result in the failure of signature verification, which can guarantee the integrity of the message. 4) Freshness. Replayed data can be distinguished by checking the time-stamps, i.e. freshness of data can be provided.

Besides, the proposed method can also counteracts some common security threats. The analysis is presented as follows: 1) Active attack. The proposed method uses secure digital signature schemes to provide strong authentication and message integrity. This makes it impossible for an intruder to modify a valid message sent by another legitimate sender. In the meantime, the time stamp T_s can prevent duplication of a broadcast message. 2) DoS attack. The proposed broadcast scheme provides authentication without delay. Hence, it prevents DoS attack which is faced in μTESLA. 3) Node compromise attack. In symmetric key schemes, if a single key or a subset of keys are used by more than one sensor nodes to calculate a message authentication code, the compromise of a single node enables an intruder to impersonate all sensor nodes sharing that message authentication code key(s). However, in this scheme, an intruder can only impersonate the compromised node. Furthermore, for revocation process this node will be not able to broadcast more messages in the network. 4) False data injection attack. The proposed method enables all sensor nodes in the communication path to verify and filter out false injected data during multi-hop forwarding, thus the false data injection attack is counteracted.

4.2. *Characteristic analysis of compound chaotic sequence*

(1) Correlation analysis. The auto-correlation function describes the degree of dependence between two state values of any one sequence. The definition of correlation is

$$R_x(j) = \frac{1}{L}\sum_{i=1}^{L} X_i X_{i+j}, \tag{31}$$

where L is the sequence length, and j is the relevant interval. The auto-correlation of the original Chebyshev map is shown in Fig. 3(a) when $a = 2^{15}$.

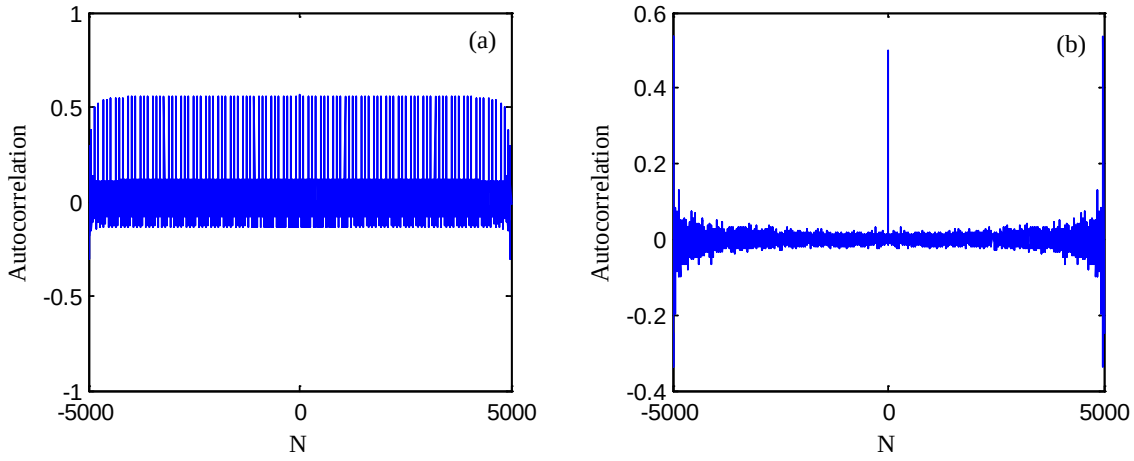

Fig. 3. Autocorrelation function of two comparative systems. (a) Discretized Chebyshev map. (b) Discretized compound system.

As shown in Fig. 3(a), it can be seen that the auto-correlation of the Chebyshev sequence is bad. Ideally, the auto-correlation function of any random series should be a delta function. Fig. 3(b) is the auto-correlation of the improved digital system, which shows that the correlation characteristics of the compound chaotic system are good.

(2) NIST test. The NIST test suite is an industry standard of random test through 15 tests, and it is used to verify the performance of the designed binary sequence. NIST test is for binary sequence, so the first step is to transfer the integer sequence into binary sequence, that is, each of these integers can be converted into a 16-bit binary data. Specifically, these 15 tests focus on evaluating the same sequence with n bits to obtain a p-value. For the specific experiment, the binary sequence is with a length of 10^7 bits. The significance level α is set at 0.01, which is commonly used in the NIST test. The result demonstrates that the binary sequence can pass the statistical test if p-value $\geq \alpha$; otherwise, it fails. Table 2 summarises the experimental result and it illustrates the produced binary sequence can pass all the standard, which means the binary sequence is randomness to some extent.

Table 2. Uniformity of the P-value under each test in the NIST suite (compound chaotic system)

Statistical tests	p-value	Conclusion
Frequency test	0.495025	pass
Block Frequency test (m=128)	0.677011	pass
Cusum test mode 1 (forward)	0.583273	pass
Cusum test mode 2 (reverse)	0.527143	pass
Rank test	0.514326	pass
Long runs of ones test	0.296117	pass
Runs test	0.535826	pass
FFT test	0.650647	pass
Non-overlapping Templates test	0.991656	pass
Overlapping Template test (m=9)	0.156512	pass
Universal test	0.354654	pass
Approximate entropy (m=10)	0.592749	pass
Random Excursions (x=+1)	0.518265	pass
Random Excursions Variant (x=-1)	0.457353	pass
Linear Complexity (M=500)	0.341668	pass
Serial (m=16)	0.649842	pass

4.3. *The security and performance analysis for hash function*

In this section, the final hash value is chosen as 128-bit in order to better compare the performance of this work with that of other schemes from the aspects of distribution, sensitivity, statistical performance, and collision resistance analysis.

(1) Distribution analysis. For a hash function, the uniformity of the generated hash value is an important criterion for evaluating its security. In this test, the initial plain-text message is: "In general, the identity authentication is based on three parameters: what you know (password based), what you possess (smart cards), and what you are (biometrics). And for WSNs, the authentication is mostly based upon the first situation (what you know). Besides, the password-based scheme must be closely related to the secret keys and any other cryptographic material that is usually used by identities to prove their uniqueness. Fig. 4 is the test result, in which Fig. 4(a) is the distribution of the ASCII code values of the original plain-text, and Fig. 4(b) shows the distribution of the corresponding hexadecimal hash values. It can be seen that the original plain-text is concentrated in $100 \sim 120$, and the distribution of the hash values is relatively uniform. Besides, in order to exclude the special circumstances, the distributions of the full "0" plain-text and its corresponding hash value are experimented and tested, and the result is shown in Fig. 5. These simulation results demonstrate that the proposed hash scheme has a good performance on confusion

and diffusion.

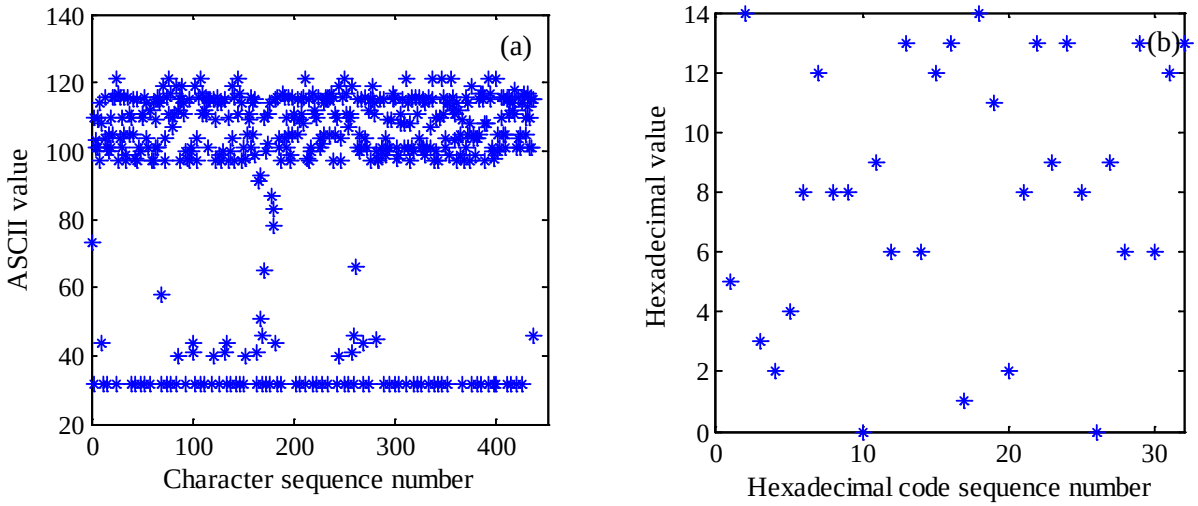

Fig. 4. Distribution: (a) the initial message and (b) the corresponding hash value.

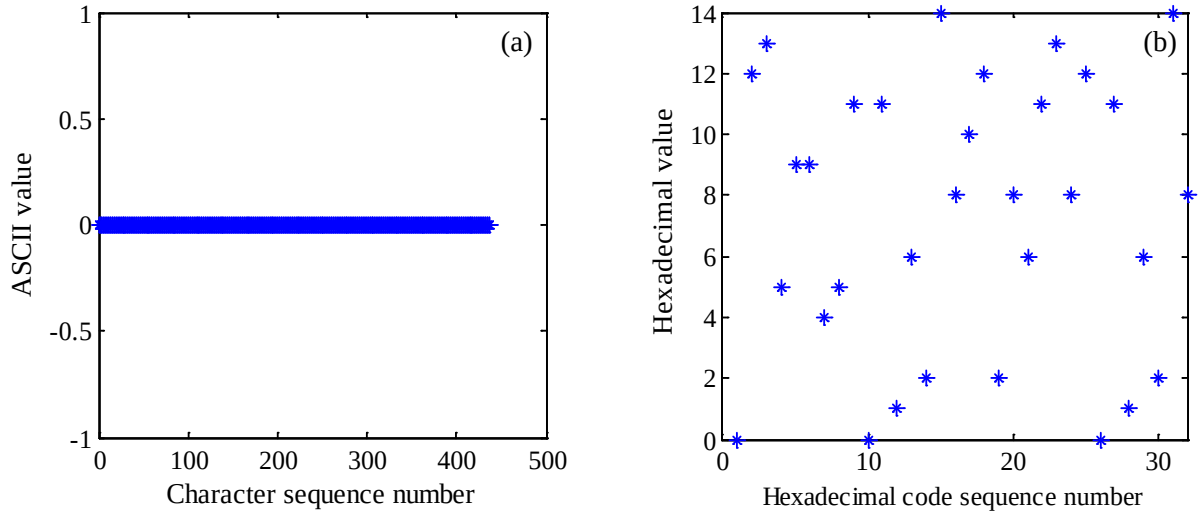

Fig. 5. Distribution: (a) of full "0" message and (b) the corresponding hash value.

(2) Sensitivity of hash value to message. Initial sensitivity is another important criterion to evaluate the security of the hash function. A good hash function should satisfy that any slight change of the plain-text will cause the generated hash value has a huge change. In this experiment, the original message is set as ASlKJK4654, and then it is changed with one bit to get different hash values. Table 3 illustrates the results of the sensitivity analysis of the message. Moreover, Fig. 6 is the corresponding binary distribution of hash values in Table 3. The results show that a tiny change of the plain-text will lead to an unpredictable change of hash value. Therefore, it can be seen that the proposed hash scheme has a strong sensitivity to plain-text.

(3) Statistical analysis. The security performance of the hash function is determined by the plain-text and its hash value. It is generally known that the hash value is supposed to be 0 or 1 in binary format, then the ideal diffusion effect is that in the same condition any subtle change in the original message will

cause 50% probability of change for each bit in the hash value [Qin *et al.*, 2010]. The experimental method is to firstly select a plain-text and get its hash value, then randomly switch one bit of the original plain-text and obtain a new hash value. Compare these two hash values and set B_i as the number of the changed bits. Fig. 7 shows the distribution of the number of bits (about 64 bits) and the probability of change after 2048 tests (about 50%).

Table 3. Sensitivity analysis of hash function to message.

No	*M*	Hahs values
C1	ASlKJK4654	8BECE71139E2867B77C050C239E2867B
C2	**B**SlKJK4654	790678D526516407C81D2E7426516407
C3	A**T**lKJK4654	273D3D522F8AFA803E9629682F8AFA80
C4	AS**2**KJK4654	D5C174A66169EF4F3E0A19776169EF4F
C5	ASlKJK465**3**	E57F8D87EBE098E0C408B0DDEBE098E0

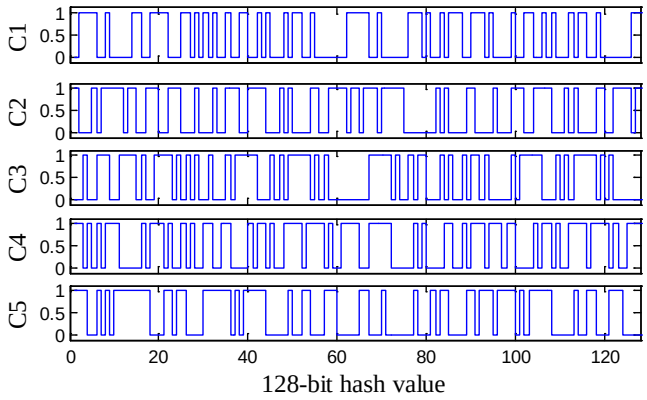

Fig. 6. Binary distribution of different hash values in Table 3.

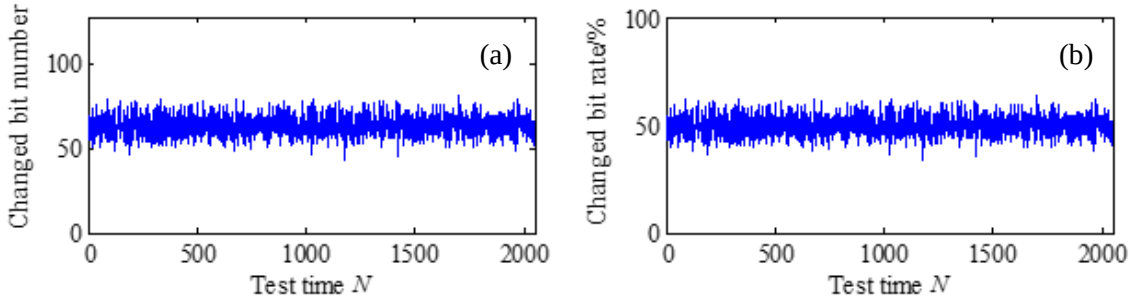

Fig. 7. Distribution of changed bit number and probability.

In addition, four statistical parameters are used to test the statistical performance which are mean changed bit number ($\bar{B}$), mean changed probability (P), standard deviation of the changed bit number (ΔB) and standard deviation of the changed probability (ΔP). Assuming N is the experimental times, the mean changed bit number ($\bar{B}$) can be calculated by $\bar{B} = \frac{1}{N}\sum_{i=1}^{B_i} B_i$. The mean changed probability (P) is calculated by $P = \frac{\bar{B}}{128} \times 100\%$. The standard deviation of the changed bit number (ΔB) is calculated

by $\Delta B = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(B_i - \bar{B})^2}$. The standard deviation of the changed probability (ΔP) is calculated by $\Delta P = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(\frac{\bar{B}}{128} - P)^2} \times 100\%$.

If N is 256, 512, 1024 or 2048, the corresponding test results are shown in Table 4. It can be seen from the statistics that the average change number of bits $\bar{B}$ is 63.9207, and the average change probability P is 49.94%, which are all closer to the theoretical value of 64 bits and 50%. In addition, the values of ΔB and ΔP are very small, which shows this hash function has good diffusion and confusion ability. Statistical simulation results show that the ability of the expected scheme can be ensured to mitigate any type of linearly related hash differential attack.

Table 4. Hash value statistics with one bit change of plain-text.

	N=256	N=512	N=1024	N=2048	Mean
$\bar{B}$	63.9297	63.7656	63.9658	64.0215	63.9207
P	5.9629	6.0716	5.9001	5.7572	5.9230
ΔB	49.95	49.82	49.97	50.02	49.94
ΔP	4.66	4.74	4.61	4.50	4.63

(4) Analysis of collision resistance. For a hash function, if the situation that different initial plain-text yields the same hash value happens, it will be called the collision. In this section, two methods are used to test the anti-collision capability of the proposed hash scheme [Liu *et al.*, 2015a].

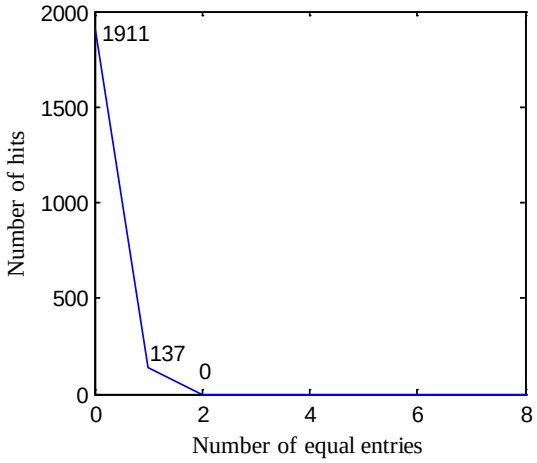

Fig. 8. The results of the hit number.

The first method is to firstly select an initial plain-text, and then randomly changing 1 bit of the initial plain-text to get a new plain-text. Through the hash function, two corresponding hash values are generated and stored in ASCII code form. Compare two hash values and count their number of identical ASCII codes at the same position, i.e., the number of collisions. Then the absolute distance is calculated between the two hash values by

$$d = \sum_{i=1}^{N} \mid m_i - m_i^{'} \mid, \tag{32}$$

where m_i and $m_i^{'}$ are the i-th ASCII character of the original and the new Hash value, respectively.

Table 5. Absolute distance between two hash values.

Absolute distance d	Maximal	Minimal	Mean
Our algorithm	2531	809	1544.7
Luo et al. [Luo & Du, 2012]	2418	796	1598.6
Ref. [Li & Sun, 2008]	2524	872	1675.0
Ref. [Yang & Gao, 2008]	2343	256	1314.0
Ref. [Xiao *et al.*, 2009]	1952	605	1227.8
Ref. [Ren *et al.*, 2009]	2455	599	1439.0
Ref. [Huang, 2011]	1882	650	1251.2
Ref. [Li *et al.*, 2011]	2220	687	1432.1
Ref. [Wang *et al.*, 2011]	2064	655	1367.0

The above process is repeated 2048 times, and the results of collision statistics and absolute distance are shown in Fig. 8 and Table 5, respectively. Fig. 8 shows the proposed scheme only 137 tests hit 1 times and other tests hit 0 times. Table 5 shows that the absolute distances between hash values in [Luo & Du, 2012] and [Li & Sun, 2008] are larger than the proposed method. This is because the proposed method abandons plain-text pre-processing operations and complex operations in the scrambling process in order to achieve a faster computation speed. However, the proposed method still has sufficient absolute distance, i.e. the collision probability of the algorithm is small.

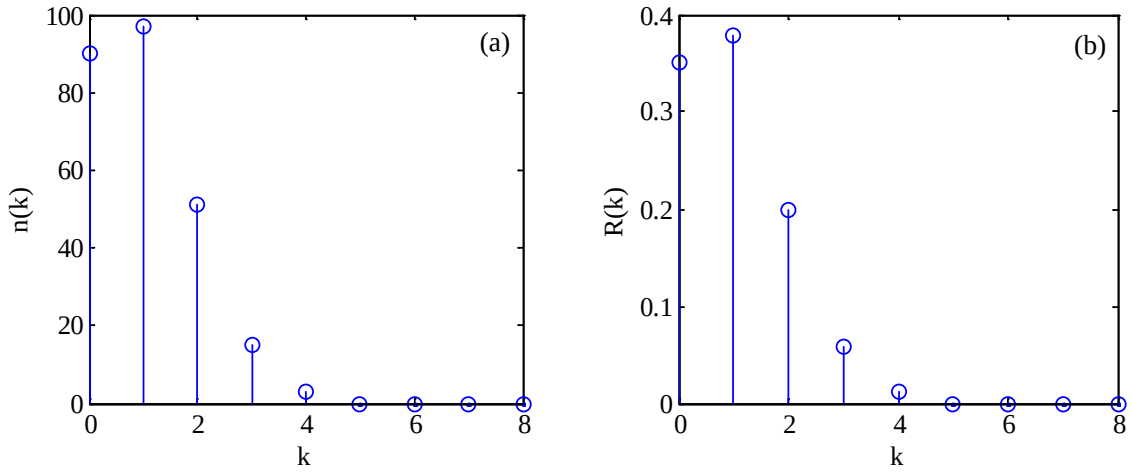

Fig. 9. Distribution of the n(k)-k and the R(k)-k.

The second test method is that the plaintext message and the corresponding hash value is firstly set to be one byte, and the range of the corresponding ASCII code is $0 \sim 255$, this will allows plaintext space to be equal to hash value space. The number of any value in the hash value space mapping into the plaintext space is k, and $n(k)$ denotes the number of k ASCII values in the hash value space. If $n(1)$ is greater, $n(0)$ and others are smaller which means the hash function algorithm has better confusion and diffusion performance, and its collision probability will be smaller. Therefore, the value of $n(1)$ can be used as the index to measure the anti-collision performance, and the anti-collision performance of the algorithm can be calculated by

$$R(k) = \frac{n(k)}{\sum n(k)}. \tag{33}$$

From Fig. 9, it can be seen that the proposed algorithm has good collision resistance performance.

4.4. *Overhead analysis*

The proposed broadcast authentication algorithm is further analysed from the perspective of the computation overhead, communication overhead and storage overhead.

(1) Computation overhead. T_{sign}, T_{verify} and T_{hash} represent the execution time that getting a signature, verifying a signature and computing a hash value, respectively. For ease of comparison, the computation overhead of the proposed scheme is measured by the average execution time at which the sender and receiver authenticate a broadcast message. Fig. 10 summarises the average time of T_{sign}, T_{verify} and T_{hash} for the broadcast message with the size of 20, 40, 60, 80, and 100, respectively. The experimental data illustrates the computation overhead is acceptable for the broadcast authentication in WSN nodes.

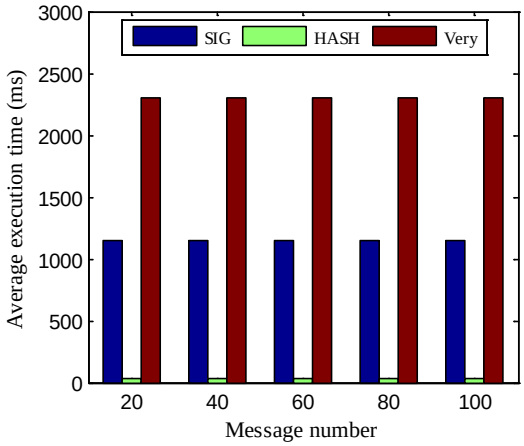

Fig. 10. Execution time of getting a signature, verifying a signature and computing a hash value.

(2) Communication overhead. As for ECM-IBS scheme, the final message broadcasted by one node usually includes the message m, time stamp T_s, identity of the sensor node ID_i and the signature S_i, i.e., $\{S_i, m, T_s, ID_i\}$, thus the total length of a transmitted message should be

$$\mid S_i \mid + \mid m \mid + \mid T_s \mid + \mid ID_i \mid = 160 + 80 + 16 + 64 = 320. \tag{34}$$

However, as for the ECM-IDS scheme in [Islam, 2014], the signature is $S_i = S_a + S_{a_1} + S_{a_2} = 160 + 160 + 160 = 480$, so the total length of a transmitted message is

$$\mid S_i \mid + \mid m \mid + \mid T_s \mid + \mid ID_i \mid = 480 + 80 + 16 + 64 = 640. \tag{35}$$

Therefore, the proposed ECM-IBS scheme is more suitable for WSNs with limited resources owing to a less communication overhead.

(3) Storage overhead. The storage space required for the master key is 1, and the network capacity is N. In the ZigBee original specification, the cost of storing the master key is $(N-1)$ in the key distribution scheme. In addition, for the SKKE protocol, the relevant parameters for the establishment of the link key need to be saved. In this paper, the master key pre-allocation process in the ZigBee original specification is avoided. Therefore, it is not required to preset the master key and store the common parameters of the ZigBee network, and the public and private key information of the nodes. Fig. 11 gives the storage overheads of the proposed ECM-IBS and ZigBee original specification. It can be seen that when the network capacity is not large, both the original ZigBee specification and the ECM-IBS scheme can maintain a small storage overhead. However with the increase of network capacity, the storage coast of the original ZigBee specification gradually increases, due to the stored redundancy of the master key. In this paper, because the authentication and key distribution process is based on identity, it does not increase the storage cost when the network capacity increases, and this maintains the scalability of the network.

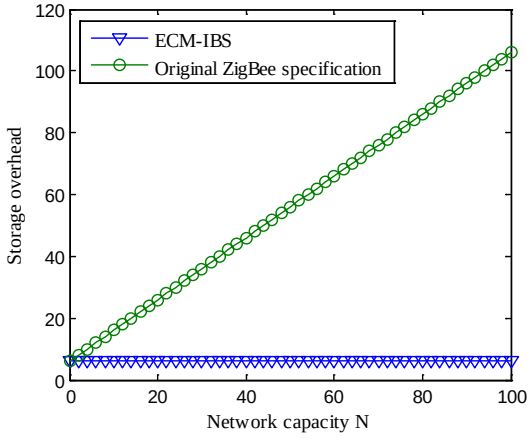

Fig. 11. Storage overheads of the ZigBee original specification and the proposed ECM-IBS scheme.

5. Conclusion

In this work, a novel authentication scheme namely ECM-IBS has been proposed, and it employs one signature to authenticate all broadcast messages. The simulation analysis demonstrates that the proposed ECM-IBS scheme has a high security and efficiency, and the hardware implementation (in WSN hardware platforms CC2530) illustrates that it has a good practical feasibility due to the low storage cost, computation and communication overheads. Therefore, the proposed ECM-IBS scheme is suitable for the WSN applications.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grants 61801131 and 61661008, the Guangxi Natural Science Foundation under Grants 2017GXNSFAA198180 and 2016GXNSFCA380017, the funding of Overseas 100 Talents Program of Guangxi Higher Education under Grant F-KA16035, the Doctoral Research Foundation of Guangxi Normal University under Grant 2016BQ005, the Science and Technology Major Project of Guangxi under Grant AA18118004, and the Innovation Project of Guangxi Graduate Education under Grant XYCSZ2018071.

References

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. & Cayirci, E. [2002] "A survey on sensor networks," *IEEE Communications Magazine* **40**, 102–114.

Attaullah & Shah, T. [2015] "An algorithm based on 1d chaotic system and substitution box," *Signal Processing* **117**, 219–229.

Azzaz, M. S., Tanougast, C., Sadoudi, S. & Bouridane, A. [2013] "Synchronized hybrid chaotic generators: Application to real-time wireless speech encryption," *Communications in Nonlinear Science & Numerical Simulation* **18**, 2035–2047.

Bergamo, P., D'Arco, P., Santis, A. D. & Kocarev, L. [2005] "Security of public-key cryptosystems based on chebyshev polynomials," *IEEE Transactions on Circuits & Systems I Regular Papers* **52**, 1382–1393.

Cao, L., Luo, Y., Bi, J., Qiu, S., Lu, Z., Harkin, J. & Mcdaid, L. [2016] "An authentication strategy based on spatiotemporal chaos for software copyright protection," *Security & Communication Networks* **8**, 4073–4086.

Chatterjee, S., Roy, S., Das, A. K., Chattopadhyay, S., Kumar, N. & Vasilakos, A. V. [2018] "Secure biometric-based authentication scheme using chebyshev chaotic map for multi-server environment," *IEEE Transactions on Dependable & Secure Computing* **15**, 824–839.

Dariush, A.-M. & Morteza, N. [2018] "Efcient anonymous password-authenticated key exchange protocol to

read isolated smart meters by utilization of extended chebyshev chaotic maps," *IIEEE Transactions on Industrial Informatics* **14**, 4815–4828.

Deng, Y., Hu, H., Xiong, W., Xiong, N. & Liu, L. [2017] "Analysis and design of digital chaotic systems with desirable performance via feedback control," *IEEE Transactions on Systems Man & Cybernetics Systems* **45**, 1187–1200.

Drissi, J. & Gu, Q. [2006] "Localized broadcast authentication in large sensor networks," *International Conference on Networking and Services*, pp. 25–25.

Du, X., Guizani, M., Xiao, Y. & Chen, H.-H. [2008] "Defending dos attacks on broadcast authentication in wireless sensor networks," *IEEE International Conference on Communications*, pp. 1653–1657.

Fontes, R. T. & Eisencraft, M. [2016] "A digital bandlimited chaos-based communication system," *Communications in Nonlinear Science & Numerical Simulation* **37**, 374–385.

Fridrich, J. [1998] "Symmetric ciphers based on two-dimensional chaotic maps," *International Journal of Bifurcation & Chaos* **8**, 1259–1284.

Habutsu, T., Nishio, Y., Sasase, I. & Mori, S. [1991] "A secret key cryptosystem by iterating a chaotic map," *Lncs* **547**, 127–140.

Hsu, C. L. & Lin, T. W. [2013] "Password authenticated key exchange protocol for multi-server mobile networks based on chebyshev chaotic map," *IEEE International Conference on Pervasive Computing & Communications Workshops*.

Hua, Z., Jin, F., Xu, B. & Huang, H. [2018a] "2D logistic-sine-coupling map for image encryption," *Signal Processing* **149**, 148–161.

Hua, Z., Zhou, B. & Zhou, Y. [2018b] "Sine-transform-based chaotic system with FPGA implementation," *IEEE Transactions on Industrial Electronics* **65**, 2557–2566.

Hua, Z., Zhou, B. & Zhou, Y. [2019] "Sine chaotification model for enhancing chaos and its hardware implementation," *IEEE Transactions on Industrial Electronics* **66**, 1273–1284.

Huang, Z. [2011] "A more secure parallel keyed hash function based on chaotic neural network," *Communications in Nonlinear Science & Numerical Simulation* **16**, 3245–3256.

Islam, S. K. H. [2014] "Design of identity-based digital signature schemes using extended chaotic maps," *IACR Cryptology ePrint Archive* **2014**, 276.

Jiang, Q., Wei, F., Fu, S., Ma, J., Li, G. & Alelaiwi, A. [2016] "Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy," *Nonlinear Dynamics* **83**, 2085–2101.

Kocarev, L. & Tasev, Z. [2003] "Public-key encryption based on chebyshev maps," *International Symposium on Circuits and Systems*, pp. 28–31.

Li, C., Feng, B., Li, S., Kurths, J. & Chen, G. [2019] "Dynamic analysis of digital chaotic maps via state-mapping networks," *IEEE Transactions on Circuits and Systems I: Regular Papers* **66**, x, doi: 10.1109/TCSI.2018.2888688.

Li, C., Lin, D., Feng, B., Lü, J. & Hao, F. [2018a] "Cryptanalysis of a chaotic image encryption algorithm based on information entropy," *IEEE Access* **6**, 75834–75842, doi:10.1109/ACCESS.2018.2883690.

Li, C., Lin, D., Lü, J. & Hao, F. [2018b] "Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography," *IEEE MultiMedia* **25**, 46–56, doi:10.1109/MMUL.2018.2873472.

Li, Y., Deng, S. & Xiao, D. [2011] "A novel hash algorithm construction based on chaotic neural network," *Neural Computing & Applications* **20**, 133–141.

Li, Y. & Sun, W. [2008] "Hash function based on the generalized henon map," *Chinese Physics B* **17**, 1685–1690.

Liu, D. & Ning, P. [2004] "Multilevel μtesla:broadcast authentication for distributed sensor networks," *ACM Transactions on Embedded Computing Systems* **3**, 800–836.

Liu, D., Ning, P., Zhu, S. & Jajodia, S. [2005] "Practical broadcast authentication in sensor networks," *International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pp. 118–129.

Liu, L., Miao, S., Cheng, M. & Gao, X. [2016] "A pseudorandom bit generator based on new multi-delayed chebyshev map," *Information Processing Letters* **116**, 674–681.

Liu, L., Miao, S., Hu, H. & Deng, Y. [2015a] "On the eigenvalue and shannon′s entropy of finite length random sequences," *Entropy* **17**, 154–161.

Liu, Q., Li, P., Zhang, M., Sui, Y. & Yang, H. [2015b] "A novel image encryption algorithm based on chaos maps with markov properties," *Communications in Nonlinear Science & Numerical Simulation* **20**, 506–515.

Liu, Y., Luo, Y., Song, S., Cao, L., Liu, J. & Harkin, J. [2017] "Counteracting dynamical degradation of digital chaotic chebyshev map via perturbation," *International Journal of Bifurcation & Chaos* **27**, 1750033.

Luo, Y. & Du, M. [2012] "One-way hash function construction based on the spatiotemporal chaotic system," *Chinese Physics B* **48**, 84–93.

Matthews, R. [1989] "On the derivation of a chaotic encryption algorithm," *Cryptologia* **8**, 29–41.

Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V. & Culler, D. E. [2002] "Spins: security protocols for sensor networks," *Wireless Networks* **8**, 521–534.

Qin, K., Zhou, M. & Feng, Y. [2010] "A novel multicast key exchange algorithm based on extended chebyshev map," *International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 643–648.

Ren, H., Wang, Y., Xie, Q. & Yang, H. [2009] "A novel method for one-way hash function construction based on spatiotemporal chaos," *Chaos Solitons & Fractals* **42**, 2014–2022.

Ren, K., Lou, W., Zeng, K. & Moran, P. J. [2007] "On broadcast authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications* **6**, 4136–4144.

Wang, Q., Yu, S., Li, C., Lü, J., Fang, X., Guyeux, C. & Bahi, J. [2016] "Theoretical design and FPGA-based implementation of higher-dimensional digital chaotic systems," *IEEE Transactions on Circuits and Systems I: Regular Papers* **63**, 401–412.

Wang, Y., Wong, K. W. & Xiao, D. [2011] "Parallel hash function construction based on coupled map lattices," *Communications in Nonlinear Science & Numerical Simulation* **16**, 2810–2821.

Xiao, D., Liao, X. & Wang, Y. [2009] "Parallel keyed hash function construction based on chaotic neural network," *Neurocomputing* **72**, 2288–2296.

Yan, S., Zhen, P. & Min, L. [2015] "Provably secure public key cryptosystem based on chebyshev polynomials," *Journal of Communications* **10**, 380–384.

Yang, Q. & Gao, T. [2008] "One-way hash function based on hyper-chaotic cellular neural network," *Chinese Physics B* **17**, 2388–2393.

Zhang, Y., He, Q., Xiang, Y., Zhang, L. Y., Liu, B., Chen, J. & Xie, Y. [2018] "Low-cost and confidentiality-preserving data acquisition for internet of multimedia things," *IEEE Internet of Things Journal* **5**, 3442–3451.