Sensors **2010**, *10*, 8683-8695; doi:10.3390/s100908683

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Broadcast Authentication for Wireless Sensor Networks Using Nested Hashing and the Chinese Remainder Theorem

Mohamed Hamdy Eldefrawy [1], Muhammad Khurram Khan [1,*], Khaled Alghathbar [1,2] and Eun-Suk Cho [3]

[1] Center of Excellence in Information Assurance (CoEIA), King Saud University, PO Box 92144, Riyadh 11653, Saudi Arabia; E-Mails: meldefrawy@ksu.edu.sa (M.H.E.); kalghathbar@ksu.edu.sa (K.A.)

[2] Information Systems Department, College of Computer and Information Sciences, King Saud University, Saudi Arabia

[3] Department of Multimedia, Hannam University, 133 Ojeong-dong, Daedeok-gu, Daejeon 306-791, Korea; E-Mail: eunsukk@empal.com (E.S.C.)

* Author to whom correspondence should be addressed; E-Mail: mkhurram@ksu.edu.sa; Tel.: +966-1-4696457.

Received: 23 July 2010; in revised form: 23 August 2010 / Accepted: 6 September 2010 / Published: 17 September 2010

Abstract: Secure broadcasting is an essential feature for critical operations in wireless sensor network (WSNs). However, due to the limited resources of sensor networks, verifying the authenticity for broadcasted messages is a very difficult issue. µTESLA is a broadcast authentication protocol, which uses network-wide loose time synchronization with one-way hashed keys to provide the authenticity verification. However, it suffers from several flaws considering the delay tolerance, and the chain length restriction. In this paper, we propose a protocol which provides broadcast authentication for wireless sensor networks. This protocol uses a nested hash chain of two different hash functions and the Chinese Remainder Theorem (CRT). The two different nested hash functions are employed for the seed updating and the key generation. Each sensor node is challenged independently with a common broadcasting message using the CRT. Our algorithm provides forward and non-restricted key generation, and in addition, no time synchronization is required. Furthermore, receivers can instantly authenticate packets in real time. Moreover, the comprehensive analysis shows that this scheme is efficient and practical, and can achieve better performance than the µTESLA system.

Keywords: wireless sensor network; authenticated broadcast; nested hashing chains; Chinese Remainder Theorem

1. Introduction

Achieving broadcast security is a must for wireless sensor networks; hence it is necessary for the base station to broadcast commands and data to sensor nodes. Without secure communication, sensors may be involved in incorrect operations and can't meet the network requirements. The current security solutions for wired and wireless networks cannot be utilized for a wireless sensor network because of the energy, memory and computation restrictions of the latter. These limitations make the design and operation completely dissimilar to those of regular wireless networks. Broadcast authentication based on asymmetric key cryptography cannot deal with the limited resource constrains. Symmetric key cryptography and hash functions are cheaper in their computational requirements and are more widely utilized in sensor networks [1,2]. WSNs' broadcast authentication was first covered by TESLA [3], and μTESLA [4] that provides the asymmetric cryptographic property of authenticated broadcast through delayed disclosing (time-varying) of symmetric keys. The base-station installs a key chain by repeatedly applying a one way hash function (OWHF) to an initial random value, called seed. The chain construction allows nodes to verify the authenticity of the disclosed keys. Loosely time synchronized and MAC (Message Authentication Code) generations are required. Revelation of session keys by the base-station is delayed, thus allowing nodes to verify the key validity.

Multilevel μTESLA [5] is proposed to reduce the need to reinitialize the network by implementing multiple levels of key chains, in which high-level keys are used to communicate root-keys (or commitments) for low-level chains, which are used in turn for broadcast authentication as in standard μTESLA. Network lifetime is extended. Significant computation and storage are required. Receivers can't deal with the received messages instantly and have to store them within one or several time intervals. Considering the broadcasting of urgent messages like alerts and alarms; the TESLA family has great shortcomings in dealing with such matters. Furthermore, the delayed authentication can be subject to Denial-of-Services (DoS) attacks. Merkle tree utilization [6] was introduced to overcome this shortage in bandwidth and storage resources utilization. TIK [7] was proposed to achieve immediate authentication based on sensitive time synchronization between the sink and the receiving nodes. However, this technique is not suitable for WSNs, as mentioned by its inventors. Sensor nodes have a limited battery life, which can make using asymmetric key techniques impractical as they use much more energy for their mathematical calculations. We propose a new algorithm that uses two different types of hash functions, which come with a nested chain and the Chinese Reminder Theorem in order to get a common broadcasting message. The resulting chain provides the forwardness and the infiniteness, and no process restarting is required. The proposed protocol is compared with others in terms of its computational cost and security attributes.

The rest of this paper is organized as follows: Section 2 discusses the related work, Section 3 discuses the required attributes, Section 4 proposes our new algorithm, Section 5 evaluates our

scheme's performance, Section 6 analyzes the security attributes, and finally Section 7 concludes the paper.

2. Related Work

The following subsection discuses some of the schemes related to WSN authentication broadcasting. Their efficiency and shortcomings according to the desirable security attributes that will be discussed will also be illustrated.

2.1. Lamport's Scheme

Hash chains were first proposed by Lamport [8]. They involve applying a hash function $h(\cdot)$ N times to a seed (s) to form a hash chain of length N:

$$h^1(s), h^2(s), \ldots, h^{N-1}(s), h^N(s) \tag{1}$$

The user calculates the i-th key according to this relation:

$$k_i(s) = h^{N-i}(s) \tag{2}$$

The host authenticates the user by checking that the following equality holds:

$$h\left(k_t(s)\right) = h^{N-i+1}(s) \tag{3}$$

where the value $h^{N-i+1}(s)$ is already saved in the host system's file from the previous i-th authentication. After any successful authentication, the system password file is updated with the new key. This scheme has a limitation on the number of authentications, so that after reaching N authentications, a process restart is required. In addition, it is vulnerable to an opponent who sends small challenge values to users that respond with the chain initial values [9]. This attack can be referred to as a small challenge attack. Also, the users are charged with computational processes through the initialization phase, which makes the system unsuitable for WSNs.

2.2. Bicakci et al.'s Scheme

The infinite length hash chains (ILHC) proposed by [10] use a public-key algorithm, A, to produce a forward and infinite one way function (OWF). Bicakci *et al.* utilized RSA [11], where d is the private key and e is the public key. The OTP originating from initial input "s" using the RSA public-key algorithm for the i-th authentication is:

$$k_i(s) = A^i\left(s, d\right) \tag{4}$$

and the verification of the i-th key is done by:

$$k_{i-1}(s) = A\ \left(k_i, e\right) \tag{5}$$

increasing the number of cascaded exponentiations increases the computational complexity, making this algorithm very difficult to implement in limited computation devices [12].

2.3. Chinese Remainder Theorem (CRT)

If the integers $n_1, n_2, \ldots, n_k$ are pair-wise relatively prime, then the system of simultaneous congruence:

$$\begin{aligned} x &\equiv r_1 \bmod n_1 \\ x &\equiv r_2 \bmod n_2 \\ &\vdots \\ x &\equiv r_k \bmod n_k \end{aligned} \tag{6}$$

has a unique solution: $x = \sum_{i=1}^{k} r_i N_i^{-1} N_i \bmod N$ where;

$$N = \prod_{i=1}^{k} n_i \tag{7}$$

$$N_i = \frac{N}{n_i} \tag{8}$$

$$N_i^{-1} N_i \equiv 1 \bmod n_i \tag{9}$$

2.4. TESLA Family Broadcast Authentication

Timed Efficient Stream Loss-tolerant Authentication (TESLA) [3] is a multicast stream authentication protocol. Keys used to authenticate the *i*-th message is disclosed along with *(i + 1)*-th message. μTESLA [4] provides authentication for data broadcasts, and requires that base station and sensor nodes be loosely time synchronized. According to Lamport's scheme, a base station (BS) randomly selects the last key k_n, the chain seed, and applies a one-way public function $h(\cdot)$ to generate the rest of keys: $k_0, k_1, \ldots, k_{n-1}$ as $k_i = h(k_{i+1})$. Given k_i, every sensor node can generate the sequence $k_0, k_1, \ldots, k_{n-1}$. However, given k_i, no one can generate k_{i+1}. At *i*-th time slot, BS sends an authenticated message $MAC_{k_i}(message)$. Sensor nodes store the message till the verification key in the *(i + 1)*-th time slot is disclosed. Sensor nodes verify disclosed key k_{i+1} by using key k_i as $k_i = h(k_{i+1})$. In μTESLA, nodes are required to store a message until the authentication key is disclosed. This operation may create storage problems, and encourages DoS types of attacks.

μTESLA has been expanded to Multi-level μTESLA [4] by simplifying the key distribution phase and introducing a new concept of a multi-level key chain generation using pseudo-random functions that improves the protocol efficiency. Multi-level μTESLA reduces the need to reinitialize the network (although re-initialization is still required) by implementing multiple levels of key chains, in which high-level keys are used to communicate root-keys (or commitments) for low-level chains which are used in turn for broadcast authentication as in standard μTESLA. The chains are further connected in that each root-key is derived from the corresponding high-level chain using another pseudo-random function. Network lifetime is extended many times over, but it is still limited. A problem would result if a receiver dropped a related commitment distribution message initializing a new low-level chain; it would be unable to verify any broadcast data received during this entire lifetime of the chain itself. The data would still be verifiable eventually as the receiver could use any later commitment distribution message to reconstruct all the lost high-level keys and the corresponding chains. This would require significant computation and storage.

2.5. CRTBA Broadcast Authentication

The scheme proposed in [13] is divided into three phases: Distribution, Message Signing, and finally Message Authentication phase. Before deployment all nodes are loaded with the chain seed, k_n, the OWHF $h(\cdot)$, and two different modules values, n_A and n_B for the CRT. When the BS needs to broadcast a message m to sensor nodes for the i-th session, BS calculates the MAC of the message m using k_i to get $M = MAC_{k_i}(m)$. After that BS cipher k_i and M using the two secrets values n_A and n_B through the CRT to get: $U \equiv k_i \bmod n_A$ and $U \equiv M \bmod n_B$, then it broadcast U. Upon the occurrence of U reception by sensor nodes, they recover k_i from U, and then apply the OWHF $h(\cdot)$, to check $k_j \stackrel{?}{=} h^{i-j}(k_i)$ where k_j is the last authentic key that sensor nodes have received. Finally, to verify the message integrity, the sensor nodes compute the corresponding MAC using k_i of the received message and then compare the result. Unfortunately, this scheme also has a length restriction considering the use of a backward hashing chain to generate keys.

3. Required Attributes

Here we list a number of desirable security attributes for authenticated broadcast:

3.1. Data Integrity

Data integrity ensures that data has not been altered by unauthorized entities.

3.2. Data Origin Authentication

Data Origin Authentication guarantees the origin of data. It is a fundamental step in achieving entity authentication in protocols as well as establishing keys. We may say that data origin authentication implies data integrity. So it is not possible to achieve data integrity without data origin authentication.

3.3. Freshness

Packets that have been captured and replayed at a later time should be ignored by the sensor nodes.

3.4. Delay Tolerance

No time synchronization should be required in the system for data verification. Each packet must be verifiable without having to wait for additional data.

3.5. Confidentiality

Confidentiality ensures that data is only available to those authorized to obtain it.

3.6. Denial-of-Service Attack

The denial of service attack is an attempt to make a node resource unavailable to its intended users.

3.7. Small Challenge Attack

This attack challenges the backward hashing with small values to respond with the chain initial values.

3.8. Limitation for an N times Authentications

Process re-initialization after *N* of authentications is necessary.

4. Our Approach

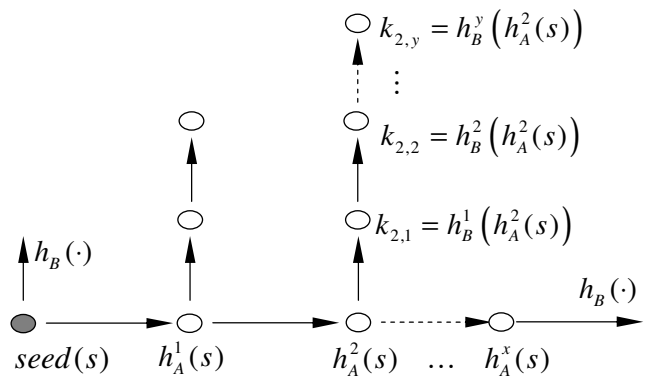
The basic idea of our scheme is to expand Lamport's scheme [8] with some modifications that produce the desirable infiniteness and forwardness, avoiding the use of public key cryptography. The shortcoming of those two parameters, infiniteness and forwardness, causes the insufficiency shown with respect to the previous work.

Table 1. The Proposed Scheme Notation.

Notation	Description
$h_A(\cdot)$	Represents the first hash function
$h_B(\cdot)$	Represents the second hash function
(x_i, y_i)	The nested hashing progress values for *i*-th authentication
$h_B^{y_i}\left(h_A^{x_i}(s)\right)$	Hashing the seed by $h_A(\cdot)$ for x_i times followed by $h_B(\cdot)$ hashing for y_i times for the *i*-th session
k_{x_i, y_i}	Session key for the *i*-th authentication
U	The encryption of the concatenated message with the session by the session key
P_i	The podcasted packet for the *i*-th authentication
X	The broadcasted chain indexes, calculated by the CRT
s_{crt}	The current seed
s_{nxt}	The next seed

Thus we need to integrate Lamport's scheme using two different one way hash functions, $h_A(\cdot)$ and $h_B(\cdot)$, one for the seed chain and the other for the session key's production, as shown in Figure 1.

Figure 1. Session key production considering a nested hash chain using two different hashes.



4.1. Key Pre-loading Phase

Each node n_j is loaded with two unique CRT modules $r_{n_j}^A$ and $r_{n_j}^B$. Those modules, regarding the all nodes, are relatively primes. Also all sensors are loaded with key seed $\langle s \rangle$ and the two different hash functions, $h_A(\cdot)$ and $h_B(\cdot)$. From the other way the base station is loaded with all this information considering the all the CRT modules for all the network's nodes, the key seed $\langle s \rangle$, and the two different hash functions $h_A(\cdot)$ and $h_B(\cdot)$.

4.2. Message Authentication.

Before the broadcasting operation, BS has to do the following:

(i) Calculate the session key $k_{x_i,y_i} = h_B^{y_i}\left(h_A^{x_i}(s)\right)$ for the *i*-th authentication.

(ii) Encrypt the broadcasted message *m* concatenated with the session key k_{x_i,y_i} with the session key to get $U = E_{k_{x_i,y_i}}\left(m \| k_{x_i,y_i}\right)$

(iii) Calculate the broadcasted chain indexes, X, for the all N nodes considering the CRT

$$\begin{aligned} X &\equiv x_i \bmod r_{n_1}^A \\ X &\equiv y_i \bmod r_{n_1}^B \\ X &\equiv x_i \bmod r_{n_2}^A \\ X &\equiv y_i \bmod r_{n_2}^B \\ &\vdots \\ X &\equiv x_i \bmod r_{n_j}^A \\ X &\equiv y_i \bmod r_{n_j}^B \\ &\vdots \\ X &\equiv x_i \bmod r_{n_N}^A \\ X &\equiv y_i \bmod r_{n_N}^B \end{aligned} \tag{10}$$

The BS constructs the broadcasted packet to be $P_i = \left\{E_{k_{x_i,y_i}}\left(m \| k_{x_i,y_i}\right) \| X\right\}$ and then broadcast it to all sensors.

4.3. Authentication Verification.

Upon the reception of P_i by the all sensors, they will need to ensure that the broadcast packets come from the authenticated BS. The verification process is done as follows:

(i) Each sensor node will extract X to perform the module operation to obtain the chain indexes, e.g., n_1 will get $x_i \equiv X \bmod r_{n_1}^A$ and $y_i \equiv X \bmod r_{n_1}^B$.

(ii) After getting the chain indexes, they will perform the key generation according to these indexes by using the two different hash functions to get this $k_{x_i,y_i} = h_B^{y_i}\left(h_A^{x_i}(s)\right)$.

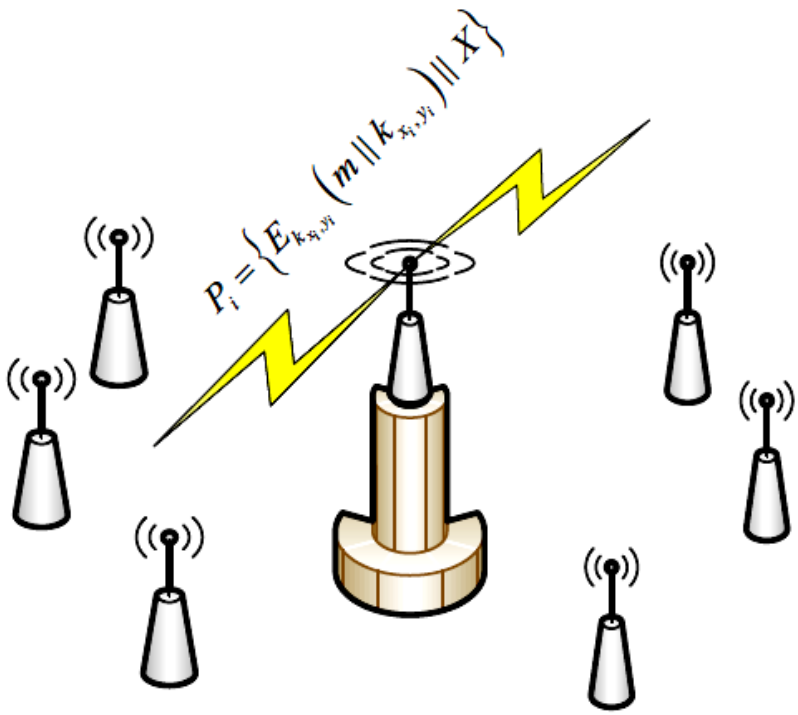
(iii) By decrypting $D_{k_{x_i,y_i}}(U)$, sensors will be able to get the message m and the session key k_{x_i,y_i}.

(iv) Then the sensor nodes need to compare the two sessions they have established and received, if the comparison is positive, then sensor nodes will recover the message. Otherwise the received broadcast message has been altered. The message integrity also checked implicitly through the authentication verification, that way tampering with U in a way of message modification will sequentially affect the received session key.

(v) After the completion of one session, sensor nodes and BS have to update the current seed to the next one:

$$s_{nxt} = \left(h_A^{x_i}\left(s_{crt}\right)\right) \tag{11}$$

Figure 2. The Proposed Broadcasting Authentication Scheme in Wireless Sensor Network.



5. Performance Analysis

In this section, we are going to analyze the performance of our algorithm with respect to the storage and computational cost [14].

5.1. Storage Analysis

The storage complexity is the amount of memory (RAM size) required to store security credentials. The storage complexity affects the hardware price of sensor nodes. Our proposal requires the base station to save two keys for each sensor nodes to build the conference X , two different hash functions $h_A(\cdot)$ and $h_B(\cdot)$, and one seed $\langle s. \rangle$ This storage overhead is neglected to the base station, since the base station regarded as resource-rich node. In the other way, sensor node n_j has to store two privet keys $r_{n_j}^A$ and $r_{n_j}^B$, and one seed $\langle s \rangle$, each one of them is 160-bit . This tells us that the memory required for credentials per module (RAM) is 160 × 3-bit = 480-bit = 60-bytes. Hash functions $h_A(\cdot)$ and

$h_B(\cdot)$ are implemented, written in nesC code for TinyOS, in approximately 20 Kbyte of memory (ROM.)

5.2. Computation Analysis

Considering the computational complexity, base station has to build the congruent equation (10) to reach the chain indexes for all sensors, X , also it has to perform two different hash operations to build the session key k_{x_i,y_i} this computation is affordable in the base station. Alternatively sensor nodes have to do two different modulo operation and to perform the same two different hash operations according to $h_A(\cdot)$ and $h_B(\cdot)$. This also is very easy to the sensor nodes. Rather than the previous techniques which use backward hash functions. Those previous techniques cost the sensor nodes to perform hashing operations for many times, especially through the chain initial values.

Example: Considering the chain length to be N = 1,000 the number of required hash operation considering Lamport scheme will be. (N + 1) × (N/2) = 500,500. On the contrary the usage of nested hashing will require the sensors to perform $2N$ hash operations which are equal to 2,000, according to our illustration. This could show how the nested hashing using two different hash chains is very cheap, in a very simple way.

Now, we consider the required execution time for a sensor node to calculate the session key $k_{x_i,y_i} = h_B^{y_i}\left(h_A^{x_i}(s)\right)$. The utilization of the microprocessor Sparc(400) as the sensor nodes' platform, will give us the following: the required time to digitize a plain text of size 80 bytes using MD5 will cost us a = 39 μs and also, the required time to digitize a plain text of size 64 bytes using SHA-1 will cost us b = 56 μs as shown in Table 2 [15], such that the total time required to calculate the session key $k_{x_i,y_i} = h_B^{y_i}\left(h_A^{x_i}(s)\right)$ is $t_{exec} = a \times x_i + b \times y_i$. Considering that the maximum values for x_i and y_i are $w = 10$, hence t_{exec} = 10(56 + 39) = 0.95 ms. Note we have considered the worst case, hence we have considered the largest input plaintext for the both two hash algorithms, but in fact the plain text size will be no more than 160-bits = 20-bytes, rather than the 80 bytes or 64 bytes.

Table 2. Execution times [μs] for two different hash algorithms, platforms and plaintext sizes [bytes].

Algorithm	Size	Atmega103	Atmega128	M16C/10	StrongARM	Xscale(400)	Xscale(200)	Sparc(440)
MD5	0	5,863	1,466	1,083	46	26	53	23
	1:26	5,890	1,473	1,075	46	26	53	23
	62:80	10,888	2,722	2,011	74	45	90	39
SHA-1	1	15,249	3,812	2,651	69	51	102	27
	3	15,781	3,945	5,303	69	50	103	27
	65	14543	3636	7955	133	102	205	55
	64	31,107	7,777	10,907	145	103	207	56

However, the time required for individual modulo operations $\bmod r_{n_j}^A$ and $\bmod r_{n_j}^B$ for node n_j is tiny compared to the calculation of the two different hash operations.

6. Security Analysis

According to the security attributes we have mentioned above, we are going to evaluate our approach:

6.1. Data Integrity

An implicit check for data integrity has been provided. Any data modifications that could be done will consequently affect the received vector $U = E_{k_{x_i,y_i}}\left(m \| k_{x_i,y_i}\right)$ which will be discovered through the key checking, by comparing the two sessions they have established and received.

6.2. Data Origin Authentication

Sending an original copy of the session key concatenated with the message and then encrypting them with the same key provides the originality authentication in a straightforward way. No one has the ability to build the broadcasted packet $P_i = \left\{E_{k_{x_i,y_i}}\left(m \| k_{x_i,y_i}\right) \| X\right\}$ except for the base-station or an intruder that has captured the entire congruence keys $r_{n_j}^A$ and $r_{n_j}^B$ for all nodes. This broadcast message has to provide the positivity authentication check considering the all sensor nodes.

6.3. Freshness

Our proposal allows the base station to challenge the sensor nodes with unpredictable uniformly distributed values of (x_i, y_i). According to these values, and according to the seed updating every session, new refreshed keys have been established every session, so the communication system has a new and refreshed session key, and previous messages cannot be replayed. If we suppose that x_i and y_i can take one value of forward m values, the probability of successfully guessing a challenge will be the joint probability of x_i and y_i, which is equal to $1/m^2$. We can refer to this property as the ability to resist predictable attacks.

6.4. Delay Tolerance

Our proposed scheme provides an instant authentication. Every broadcasted packet contains the authentication information for itself, independently of previous and following messages. The authentication process is done in the same session.

6.5. Confidentiality

Confidentiality cannot be guaranteed if one or more nodes have been compromised. If an intruder acquires the ability to capture one node or more he will be able to solve the congruent equation using the captured node n_j congruent keys $r_{n_j}^A$ and $r_{n_j}^B$. The CRTBA [13] algorithm also did not cover this property, furthermore the broadcasted messages are sent in the plain form without encryption. Actually, regarding certain applications like the broadcasting of urgent alert notifications and warning systems need instant message authentication rather than confidentiality.

6.6. Denial of Service Attacks

In μTESLA scheme, the sensor nodes can't authenticate the received message immediately after reception. The intruder can send a large amount of forged messages to consume the sensor nodes buffer. The instant authentication provided in our scheme, overcomes this weakness. The authentication process is done in the same session independently of the previous or the next sessions. This vulnerability is overcome without resources an extra bandwidth or an extra storage memory like [5] and [6].

6.7. Limitation for an N times Authentications

All TESLA families and also CRTBA, use backward hash chain. The backward chain has a restriction of an *N* time for authentications; a process restart is required after reaching this number of authentications. Our algorithm utilizes a new technique of employing two nested and different hash functions for the key production. This technique uses forward hashing and has no need for process restarting after reaching any number of authentications.

6.8. Small Challenge Attack

Utilizing a one way hash function to construct a hashing chain in the backward fashion encourages a new kind of attack called small challenge attack. This type of attack discloses the hash chain initial values. These initial values help the intruder to extract the remaining chain values by hashing those initial values. Our algorithm covers this vulnerability by the utilization of two different and nested hash functions in the forward fashion, which prevents this kind of attack.

6.9. Brute Force Attack

The ability of generating a truly random sequence of key bits can defeat a brute force attack, as a brute force attack would have no way of distinguishing one key from the other. Relying on the generation of random number can impede the brute force. The nested hashing progress random values for *i*-th authentication (x_i, y_i). play a great role in preventing this type of attacks according to the entropy of their random generation.

7. Conclusions

A new wireless sensor network broadcast authentication scheme based on forward hashing using two different nested hashes and the Chinese Reminder Theorem (CRT) has been presented. The broadcasting messages are built using the congruence of the CRT. The two different hashing systems are utilized in the session key generation in a forward and unlimited way. This scheme achieves better characteristics than the other schemes, we discussed. Our proposal is not limited to a certain number of authentications, and also does not involve computationally expensive techniques (PKC) to provide infiniteness. A detailed security analysis has been performed that covers many types of attacks that could influence our scheme. Our scheme satisfies all the security attributes, we have discussed, except for the confidentiality in case of one node or more has been captured. This scheme is applicable for alerting and warning systems that need instant broadcast authentication rather than message confidentiality.

References

1. Fan, Y.; Chen, I.R.; Eltoweissy, M. On Optimal Key Disclosure Interval for μTESLA: Analysis of Authentication Delay *Versus* Network Cost. In *Proceedings of International Conference on Wireless Networks, Communications and Mobile Computing*, Hawaii, HI, USA, 13–16 June 2005; Volume 13, pp. 304-309.
2. Shi, E.; Perrig, A. Designing Secure Sensor Networks. *IEEE Wirel. Commun.* **2004**, *11*, 38-43.
3. Perrig. A.; Canetti. R.; Tygar. J.; Song. D. The TESLA Broadcast Authentication Protocol. *CrytoBytes* **2002**, *5*, 2-13.
4. Liu, D.; Ning, P. Efficient Distribution Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks. In *Proceedings of the 10th Annual Network and Distributed System Security Symposium*, San Diego, CA, USA, 6–7 February 2003; Volume 2, pp. 263-276.
5. Liu, D.; Ning, P. Multi-level μTESLA: Broadcast Authentication for Distributed Sensor Networks. *ACM Trans. Embed. Comput. Syst.* **2004**, *3*, 800-836.
6. Liu, D.; Ning, P.; Zhu, S.; Jajodia, S. Practical broadcast authentication in sensor networks. In *Proceedings of Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, San Diego, CA, USA, 17–21 July 2005; pp. 118-129.
7. Hu, Y.; Perrig, A.; Honson, D.; Packet Leashes. A Defense against Wormhole Attacks in Wireless *Ad hoc* Networks. In *Proceedings of INFOCOM*, San Francisco, CA, USA, 30 March–3 April 2003.
8. Lamport, L. Password Authentication with Insecure Communication. *Comm. ACM* **1981**, *24*, 770-772.
9. Chefranov, A. One-Time Password Authentication with Infinite Hash Chains. In *Novel Algorithms and Techniques in Tele-communications, Automation and Industrial Electronics*; Springer: New York, NY, USA, 2008; pp. 283-286.
10. Bicakci, K.; Baykal, N. Infinite Length Hash Chains and Their Applications. In *Proceedings of 11th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborating Enterprises*, Pittsburgh, PA, USA, 10–12 June 2002; pp. 57-61.
11. Rivest, R.; Shamir, A.; Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Comm. ACM* **1978**, *21*, 120-126.
12. Khan, M.K; Alghathbar, K. Cryptanalysis and Security Improvements of "Two-Factor User Authentication in Wireless Sensor Networks". *Sensors* **2010**, *10*, 2450-2459.
13. Zhang, J.; Yu, W.; Liu, X. CRTBA: Chinese Remainder Theorem-Based Broadcast Authentication in Wireless Sensor Networks. In *Proceedings of Computer Network and Multimedia Technology*, Wuhan, China, 18–20 January 2009.
14. Eldefrawy, M.; Khan, M.K.; Alghathbar, K. A Key Agreement Algorithm with Rekeying for Wireless Sensor Networks Using Public Key Cryptography. In *Proceedings of International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, Chengdu, China, 18–20 July 2010.
15. Venugopalan, R.; Ganesan, P.; Peddabachagari, P.; Dean, A.; Mueller, F.; Sichitiu, M. Encryption Overhead in Embedded Systems and Sensor Network Nodes: Modeling and Analysis. In *Proceedings of International Conference on Compilers, Architecture and Synthesis for Embedded System*s, San Jose, CA, USA, 30 October–1 November 2003.

16. Khan, M.K.; Zhang, J. Improving the Security of "A Flexible Biometrics Remote User Authentication Scheme". In *Computer Standards and Interfaces*; Elsevier Science: North Holland, The Netherlands, 2007; Volume 29, pp. 84-87.