The ISC Int'l Journal of Information Security
ISeCure

January 2014, Volume 6, Number 1 (pp. 53–66)
http://www.isecure-journal.org

Toward an Energy Efficient PKC-Based Key Management System for Wireless Sensor Networks☆

Hamzeh Ghasemzadeh[1],*, Ali Payandeh[2], and Mohammad Reza Aref[3]

[1] *Department of Electrical Engineering, Islamic Azad University, Damavand Branch, Damavand, Iran*
[2] *ICT Department, Malek-e-Ashtar University of Technology, Tehran, Iran*
[3] *Information Systems and Security Lab, Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran*

ARTICLE INFO.

Article history:
Received: 27 November 2013
Revised: 3 October 2014
Accepted: 8 October 2014
Published Online: 11 October 2014

Keywords:
Broadcast Authentication, Key Management System, Public Key Cryptography, Wireless Sensor Networks.

ABSTRACT

Due to wireless nature and hostile environment, providing of security is a critical and vital task in wireless sensor networks (WSNs). It is known that key management is an integral part of a secure network. Unfortunately, in most of the previous methods, security is compromised in favor of reducing energy consumption. Consequently, they lack perfect resilience and are not fit for applications with high security demands. In this paper, a novel method is proposed to improve the security of key management system based on broadcast messages from the base station. Another problem with WSNs is the cryptographic materials (such as private keys) stored in dead nodes. Adversaries may exploit these nodes to mount more effective attacks. Any secure key management system should also address this problem. It is argued that in the proposed method keying materials of dead nodes lose their validity, and therefore are of no use for an adversary. Finally, it is shown through simulation that the proposed method is almost three times more energy-efficient than conventional certificate-based key management systems.

© 2014 ISC. All rights reserved.

1 Introduction

Wireless sensor networks are comprised of a large number of low-power sensor-actuator nodes equipped with radio transceivers. These networks serve as an interface to the real world. It means they watch their environment for physical information such as temperature, light, radiation, etc. and then send the gathered data to a sink node for further processing. These networks are decentralized and specialized in their nature. In addition, they are self-organized and do not require the existence of a supporting infrastructure [1]. Furthermore, their network topology is not known a priori, so an airplane or artillery could deploy them to otherwise unreachable regions [2]. These unique features have promised a wide range of possible applications for them. Health-care monitoring [3], protection of critical infrastructure [4], monitoring the environment for seismic sensing, flood and volcanic eruption [5, 6], military target tracking [7], and surveillance [8] are just a few applications of these networks.

Many applications of WSNs require secure communication. Unfortunately, wireless channels are open access. Also, to reduce cost of nodes, they are not equipped with tamper resistant mechanisms. Adver-

☆ This article is an extended version of an ISCISC'13 paper.
* Corresponding author.

Email addresses: hamzeh_g62@yahoo.com (H. Ghasemzadeh), payandeh@mut.ac.ir (A. Payandeh), aref@sharif.edu (M. R. Aref).

ISSN: 2008-2045 © 2014 ISC. All rights reserved.

sary can exploit all these features to mount different types of malicious attacks. In other words, the same unique features and characteristics that have promised such a broad range of applications for WSNs could be source of many security vulnerabilities. All in all, protocols of traditional wireless networks are usually useless in WSNs. So, with characteristics of these networks in mind, existing protocols and algorithms should be tailored or new ones should be devised. Key management as the core of secure communication is not an exception. Recently, many key management schemes have been proposed for wireless sensor networks.

Eschenauer *et al.* [9] proposed a method based on the probabilistic pre-distribution of subsets of a key-pool. Their method neither provides full connectivity nor perfect resiliency. Later, based on symmetric polynomials [10] and generating matrices of linear codes [11] other methods were proposed. These methods have full connectivity and they provide threshold resiliency. It means that, their resiliency is perfect as long as the number of compromised nodes are less than a threshold value. If this value is exceeded, security of these methods vanishes completely. A good overview of symmetric-based key management systems can be found in [12, 13]. Recently, other symmetric based key management system have been proposed one of them due to Delgado *et al.* provides perfect resiliency, but it has high overhead for adding new nodes to the network [14].

Hostile environments, unattended nature of WSNs and nodes not being tamper-resistant enable adversary to capture nodes, and read all their data, including their cryptographic keys. So, adversary can easily mount a cloning attack [15]. Therefore, using a key management system that is more resilient against this attack is highly favorable.

PKC-based key management systems provide perfect resiliency and are very flexible. In fact, it can be argued that their only downsides are speed and power demand [16]. Furthermore, because all the keys are unique, detection of cloning attack would be much easier.

Previous studies [17, 18] have shown that PKC is viable on sensor nodes. For example, it takes 1.61s to verify an Elliptic Curve Cryptography (ECC) signature on ATmega128, and it consumes 45.09mj energy [19]. Furthermore, using more advanced nodes such as Imote2, this energy could be reduced to 3.51mj [20].

Following these incipient works, different PKC-based systems were proposed. First, TinyPk an authentication protocol based on RSA and ECC was presented [21]. Then, its vulnerability against masquerade attack was detected [22]. Later, Ren *et al.* [23] proposed two different broadcast authentication methods, one based on ECC and Merkle hash tree, and the other using Hesss identity based signature [24]. Later, IMBAS, another identity-based authentication scheme reduced energy consumption of authentication [25]. To further reduce energy consumption of identity-based systems, Shim *et al.* employed a pairing-optimal identity based system with message recovery method [26]. Finally, Lim used Rabin-Williams signatures [27] to authenticate code dissemination [28].

In another path pursued by researchers, they showed that heavy-energy-cost operation of signature verification can be replaced with other low-cost operations. First, a method based on bloom filter and Merkle hash tree was presented [29]. Later, Liu *et al.* proposed another low-cost method to authenticate broadcast messages based on ECC and hash functions [30].

While PKC-based key management systems have many desirable characteristics, they consume lots of energy. Continuing on our seminal work [31], this paper tries to reconcile between high security demand of critical applications and high energy consumption of PKC-based key management systems. We will show that employing broadcast messages from base station (BS) can drastically reduce energy consumption of PKC-based key management systems.

This paper makes the following contributions:

- Based on broadcast messages from BS, a novel and low energy PKC-based key management system is presented. Then, the proposed system is compared with other PKC-based key management systems.
- Necessity of time synchronization in μTesla protocol is eliminated. To this end, time differences between consecutive broadcast messages along with non-deterministic timing schedule is employed.

The rest of this paper is organized as follows. In Section 2 a brief overview of some cryptographic mechanisms is given. Section 3 presents different PKC-based key management systems. The structure of the proposed method is extensively discussed in Section 4. Section 5 lays mathematical foundation of calculating number of neighbors, a factor that affects connectivity and energy consumption of the proposed method. Then, different aspects of the proposed method are inspected. In Section 6 capabilities and characteristics of the proposed method are discussed, and finally conclusion is drawn in Section 7.

2 Preliminaries

2.1 μTesla Broadcast Authentication

Generally to achieve, broadcast authentication, an asymmetric mechanism such as public key cryptography is needed. Perrig *et al.* proposed an efficient broadcast authentication mechanism based on one-way hash functions [32]. In their scheme, messages are appended with a Message Authentication Code (MAC) generated with a secret key (K), and then asymmetry is introduced by delaying disclosure of this key. Also, the keys constitute a key chain so knowing the initial key, K_0 , is sufficient for checking its authenticity. In this manner, nodes just need to apply hash functions and compare the result with K_0. They also introduced a security condition on the local time when the packet is received, so that the receivers could ensure that the packet was sent before the key was disclosed. First, this security condition is checked, and then receivers buffer the packets and authenticate them after receiving the corresponding disclosed key. Unfortunately, this mechanism requires loose time synchronization between sender and receiver, a condition that is hard to achieve in WSNs. Figure 1, gives an example of μTesla timing schedule.

Reviewing previous literature shows that a wide variety of systems based on μTesla protocol are proposed. First, multi-level μTesla enlarged life time of the system [33]. Then, localized μTesla scheme was proposed [34]. Later, based on cyclic redundancy check codes, overhead of μTesla was reduced [35]. In another work, researchers adapted μTesla protocol to achieve inter sensor broadcast authentication [36]. Kim *et al.* proposed a method for solving scalability problem of μTesla protocol for inter sensor authentication [37]. Finally, tree based μTesla provided an efficient technique for supporting larger number of broadcast senders [38].

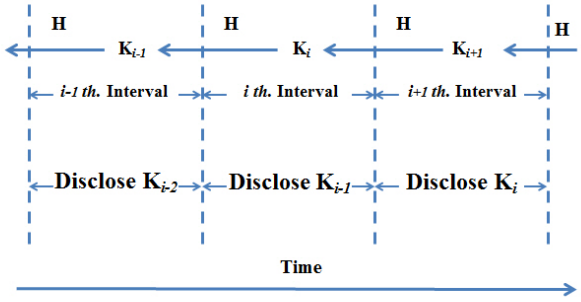

Figure 1. μTesla Broadcast authentication.

2.2 Bloom Filter

Bloom filter is an excellent and compact data structure that supports membership queries [39]. Bloom filter is an m-bit vector all initially set to 0. Now, for representing the set $S=\{s_1, s_2, \ldots, s_n\}$, k independent hash functions are selected such that $D \xrightarrow{h_i} [0, m-1], 1 \leq i \leq k$. Then, for each element $s_j \in S$ the bits $h_i(s_j), 1 \leq i \leq k$ of this vector is set to 1. After the bloom filter is constructed, if all bits $h_i(x), 1 \leq i \leq k$ of the bloom filter are equal to 1, then it is said that item x belongs to the set S. Apparently this scheme may yield a false positive which can be calculated as $f = (1 - e^{-kn/m})^k$ [40].

2.3 Merkle Hash Tree

Merkle invented an authentication scheme based on a tree of hash values [41]. Figure 2, shows this tree when there are four authentic data.

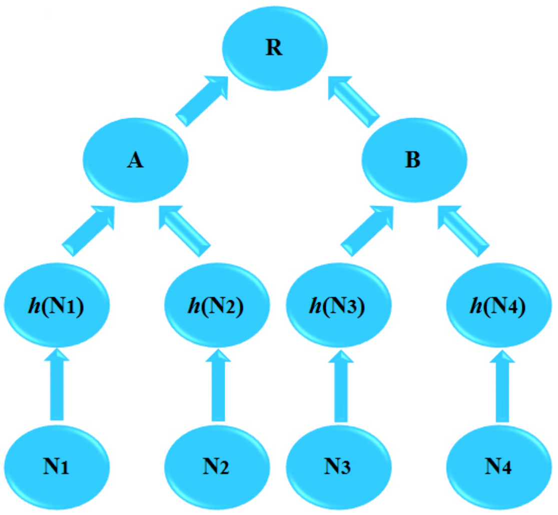

Figure 2. An example Merkle hash tree.

According to Figure 2, leafs are direct hash values of corresponding authentic data. Then, higher nodes are calculated as hash of their respective children. For example, value of node A is calculated as $A = h(h(N_1)||h(N_2))$, and the value of root is calculated as $R = h(h(A)||h(B))$. Now, any node can authenticate other nodes just by R and $\log_2^N$ auxiliary authentication information (AAI). For example, if node N_2 knows value of R, and wants to authenticate N_3, then all he needs are $< N_3, h(N_4), A >$. After receiving these data, N_2 calculates: $h(N_3), B' = h(h(N_3)||h(N_4))$ and finally $R' = h(A||B)$, and If the calculated $R' = R$ then N_2 authenticates N_3.

3 Authentication in PKC-Based Key Management Systems

Ren *et al.* identified four different schemes for PKC-based authentication of broadcast messages [29]. In this section, we employ them to construct four different PKC-based key management systems.

3.1 The Certificate-Based Authentication Scheme (CAS)

In essence, a certificate consists of a public key, an identifier and an expiration date. Then the whole block is signed by a trusted third party. Typically, the third party is a certificate authority, such as a government agency or a financial institution that is trusted by all users. In WSNs BS can play role of this trusted third party. Therefore, BS generates itself a pair of public-private key and then it uses them to issue each node a certificate. This certificate consists of the following contents: $Cert_A = \{A, P_{uA}, Exp., Sign_{P_{rBS}}(h(A, P_{uA}, Exp.))$, where A is node ID, P$_{uA}$ is the public key of A, *Exp.* is certificate expiration date, and $Sign_{P_{rBS}}(h(A, P_{uA}, Exp.))$ denotes BS signature over $h(A, P_{uA}, Exp.)$. When two nodes A and B want to share a key, they exchange their certificates. After checking validity of signatures, nodes run a protocol like ECDH and derive the shared key.

3.2 The Direct Storage Based Authentication Scheme (DAS)

If transmission and verification of certificates are eliminated, then communication and computation costs are extremely reduced. The simplest approach is to pre-load every node with all public keys of the network. In this manner, each node just needs to send its ID. Now, nodes just run ECDH to derive the shared key. Of course, this scheme has many drawbacks. For example, not only it supports limited number of nodes, but also adding new nodes to the network would be very difficult.

3.3 The Bloom Filter Based Authentication Scheme (BAS)

In this authentication scheme, BS generates public key of all nodes. Then, it concatenates ID of nodes with their public key to construct the set $S = \{< ID_{u1}, P_{u1} >, < ID_{u2}, P_{u2} >, \dots\}$. Now, BS constructs the bloom filter for this set and pre-loads all nodes with this bloom filter. After network deployment, nodes broadcast their ID and their public key. Neighbor nodes share a common key in two steps. First, they check authenticity of the received data using the stored bloom filter and then, they run ECDH and extract the shared key.

3.4 The Hybrid Authentication Scheme (HAS)

Earlier it was mentioned that bloom filter may yield a false positive. A value which depends on the length of bloom filter, a parameter that highly relates to memory complexity of this method. So, if memory of nodes and value of false positive are fixed, BAS can support specified number of nodes. For example, if $f = 6.36 \times 10^{-20}$ and the storage limit is 4.9 KiB, BAS can support up to 434 nodes [40]. Merkle hash tree can be used to alleviate this limitation. The resulting method is called hybrid scheme.

For HAS method to work, BS collects all public keys of the network and constructs their Merkle hash tree. Then, BS prunes this tree into a set of equal-sized smaller trees $\{h_r^i\} \quad i = 1, \dots, |S|$, where $|S|$ is the maximum number of supported nodes, if BAS scheme had been used. Now, BS constructs bloom filter of the $S = \{h_r^1, h_r^2, \dots, h_r^{|S|}\}$. Then, nodes are preloaded with bloom filter and their AAI. Herein, AAI is driven according to their location in the smaller Merkle hash trees.

When two nodes, A and B, want to share a key, they send each other their AAIs and their public keys. Then each node authenticates the other node in two steps. First, using received AAI, value of corresponding root is calculated. Then, authenticity of calculated root is checked using the stored bloom filter. After authenticating each other, nodes use ECDH to derive the shared key.

4 Broadcast Authenticated Key Management

In the proposed method, BS uses a MAC algorithm in conjunction with a key chain to generate one-time signatures. Then, nodes are preloaded with these data. After deployment, nodes send the appropriate signature for their neighbors, and wait for BS to broadcast the corresponding authentication key. Finally, nodes authenticate each other using the exchanged data and the revealed key. Table 1 describes the notations used in the rest of this paper.

4.1 System Preparation

First, BS generates two key chains:

(1) It generates μTesla authentication key chain. Nodes will use these keys to authenticate messages from BS.

$$K_{Authn} \to \cdots \to K_{Auth1} \to K_{Auth0} \to K_{Auth00} \tag{1}$$

(2) Also, BS generates a signature key chain. These keys are used to create one-time signatures for nodes.

$$K_{DSn} \to \cdots \to K_{DS1} \to K_{DS0} \tag{2}$$

Then, for every node x, BS does following steps:

(1) It generates the public-private parameters of ECDH scheme (P_{ux}, P_{rx}).

Table 1. Notations used in this paper.

Notation	Meaning
P_{ux}	Public key of Node x
P_{rx}	Private key of Node x
i	Cycle number
K_{DSi}	Key used to generate ith. signature
Tx_i	Time measured locally at node x
Δ_i	Time difference between two consecutive cycle
K_{Authi}	μTesla Key chain
$MAC_K(M)$	Message Authentication Code of message (M) using key (K)
$E_K(M)$	Symmetric encryption of message (M) using key (K)
K_{AB}	Pairwise key between node A and B
f	Some publicly agreed on function
g(K,l)	A function that casts out first (l) bits of (K)

(2) It creates a set of signature as the MAC of public ECDH parameter of node x using keys K_{DSi}.

$$Sign_{x_i} = \mathrm{MAC}_{K_{DSi}}(P_{ux}), \quad i = 1, \ldots, n \qquad (3)$$

Now, every node is preloaded with its ECDH parameters, its set of signatures, and the last key of broadcast authentication chain K_{Auth00}. After network deployment, BS saves its local time as T_{BS0} and starts protocol by broadcasting message (4).

$$BS \to x : \mathrm{E}_{K_{Auth0}}(K_{DS0}||0||0), T = T_{BS0} \qquad (4)$$

Upon receiving of this message every node saves its local time(Tx_0). Then, BS waits for t seconds in order to make sure that all nodes have received its broadcasted message, and then broadcast K_{Auth0} key.

$$BS \to x : K_{Auth0} \qquad (5)$$

Now, nodes first hash this key and compare the result with the stored K_{Auth00}. Then, they decrypt message (4) using K_{Auth0} and extract cycle number (here 0). After that, BS uses a schedule for initiating new authentication cycles and those nodes participating in them can authenticate each other. Cycle i would run like this:

$$x : Ticket_{xi} = [P_{ux}, Sign_{xi}],$$
$$Sign_{xi} = MAC_{K_{DSi}}(Pu_x) \qquad (6)$$

Then, BS locally calculates the time difference between previous cycle and current cycle as: $\Delta_i = (T_{BSi} - T_{BSi-1})$ and initiates a new cycle by broadcasting message (7).

$$BS \to x : E_{K_{Authi}}(K_{DSi}||i||\Delta_i) \qquad (7)$$

Upon receiving this message, every node saves its local time (Tx_i) and waits for BS to reveal K_{Authi}. After this key is disclosed, nodes first check its validity and then, they decrypt message (7). Furthermore, every node locally calculates time difference between this cycle and previous one $\Delta_{xi} = (T_{xi} - T_{xi-1})$ and compares it with the one sent from BS (Δ_i) to make sure that the packet was sent before the key was disclosed. If all the above conditions are met, nodes accept the K_{DSi} and use it for authenticating their neighbors by checking validity of their tickets. Finally, nodes run ECDH protocol and use a public function (f) to extract the shared key. For reducing communication cost, acknowledgment message is generated by adding one to the first l bits of the shared key. The complete protocol is presented in Table 2.

5 Security Analysis

We define a method secure if legitimate nodes reject bogus message from adversary (except with low probability) [42]. In authentication this is equal to preventing adversary from authenticating its nodes to the network. We show in the following subsections that proposed method can achieve security. Furthermore, we show that by detecting jamming and delaying authentication, proposed method withstands complex scenarios of replay attack.

To add new nodes to the network, adversary must generate valid signatures, since K_{Authi} is not yet disclosed; he has to wait until BS reveals it (T3). After that, adversary can generate valid signatures, but disclosure of K_{Authi} will terminate *ith.* authentication cycle. Thus, node will consider any received ticket after message T3 as belonging to the next authentication cycle. Consequently, this forged ticket will wait for the next cycle and it will be rejected.

Security and replay attack:

In this section, security of the proposed method against replay attack is investigated. Among different parts of the proposed protocol (Table 2), replaying of message T2 may lead to a security vulnerability. To this end, different scenarios are investigated.

First scenario:

Let us assume that adversary saves message T2 and replays it in the same authentication cycle but before message T3. Since message T2 is encrypted with K_{Authi}, adversary cannot read or modify its contents. In this scenario, if adversary replays message T2, in fact he will participate in the blind flooding employed by legitimate nodes to convey messages of BS to other nodes. Consequently, in this scenario network would benefit from replay attack.

Second scenario:

Let us assume that adversary saves message T2 and

Table 2. The proposed protocol.

$$A: Ticket_{Ai} = \left[P_{uA}, Sign_{Ai}\right], Sign_{Ai} = \mathrm{MAC}_{K_{DSi}}\left(P_{uA}\right)$$

$$B: Ticket_{Bi} = \left[P_{uB}, Sign_{Bi}\right], Sign_{Bi} = \mathrm{MAC}_{K_{DSi}}\left(P_{uB}\right)$$

$$BS: \Delta_i = T_{BSi} - T_{BSi-1}$$

$$T1 - \begin{cases} A \rightarrow B: Ticket_{Ai} \\ B \rightarrow A: Ticket_{Bi} \end{cases}$$

$$T2 - \begin{cases} BS \rightarrow A: \mathrm{E}_{K_{Authi}}(K_{DSi}||i||\Delta_i), T_{BS} = T_{BSi}, T_A = T_{Ai} \\ BS \rightarrow B: \mathrm{E}_{K_{Authi}}(K_{DSi}||i||\Delta_i), T_{BS} = T_{BSi}, T_B = T_{Bi} \end{cases}$$

$$T3 - \begin{cases} BS \rightarrow A: K_{Authi}, T_{BS} = T_{BSi} + t \\ BS \rightarrow B: K_{Authi}, T_{BS} = T_{BSi} + t \end{cases}$$

$$\begin{cases} A: K_{Authi} \xrightarrow{H?} K_{Authi-1}, T_{Ai} - T_{Ai-1} \overset{?}{\approx} \Delta_i, Sign_{Bi} \overset{?}{=} \mathrm{MAC}_{K_{DSi}}\left(P_{uB}\right) \\ B: K_{Authi} \xrightarrow{H?} K_{Authi-1}, T_{Bi} - T_{Bi-1} \overset{?}{\approx} \Delta_i, Sign_{Ai} \overset{?}{=} \mathrm{MAC}_{K_{DSi}}\left(P_{uA}\right) \end{cases}$$

$$\begin{cases} A: K_{AB} = f(P_{uB}.P_{rA}, i) \\ B: K_{AB} = f(P_{uA}.P_{rB}, i) \end{cases}$$

$$T4 - A \rightarrow B: \mathrm{E}_{K_{AB}}(g(K_{AB}, l) + 1)$$

replays it in another authentication cycle. Nodes will buffer the message and wait for message T3 to arrive. Apparently, the replayed message had been encrypted with K_{Authl}, whereas T3 has disclosed K_{Authi}. As these keys are different, after message T3 is received, nodes will discard replayed message.

Third scenario:

Last scenario assumes that adversary replays message T2 in the same authentication cycle, but after T3 is disclosed. This scenario is investigated thoroughly in the next two subsections.

Replay and jamming attack:

Let us assume that adversary is equipped with a jammer. In this case, adversary can jam node A and wait for K_{Authi} to be disclosed. Now, node A neither gets the key nor finds out that this cycle has been ended. Figure 3, shows such attack; legitimate nodes are in green whereas nodes of adversary are in red.

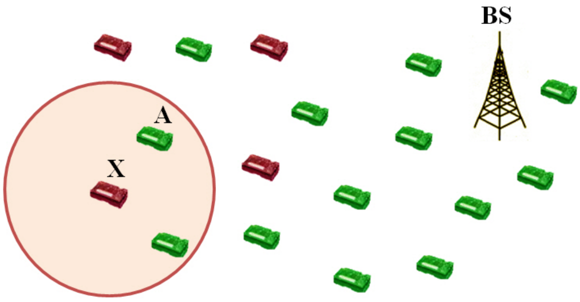

Figure 3. Attacking authentication protocol.

The following lines show the steps that adversary and node A will run.

$$X: {\Delta_i}' = (T_{Xi} - T_{Xi-1}) > \Delta_i + t \tag{8}$$

$$A \rightarrow X: Ticket_{Ai}, X \rightarrow A: Ticket_{Xi} \tag{9}$$

$$X \rightarrow A: \mathrm{E}_{K_{Authi}}(K_{DSi}||i||{\Delta_i}'), \quad T_{Xi} > T_{BSi} + t, T_A = T_{Ai} \tag{10}$$

$$X \rightarrow A: K_{Authi} \tag{11}$$

This attack is circumvented in μTesla protocol by condition on the time that packets arrive, but for this condition to work, nodes should be synchronized, an assumption that is hard to achieve in WSNs. We show using time differences can thwart this attack without any need to time synchronization. According to (10), adversary must replace value of Δ_i with Δ_i', so that node A would calculate the right time difference and accepts ticket of X. Because proposed method employs time differences, a chaining between messages exists. Therefore, modifying arrival time of a packet would affect later packets as well. In other words, this trick can be detected in the next cycles. Let us investigate next authentication cycle:

$$BS: \Delta_{i+1} = (T_{BSi+1} - T_{BSi}) \tag{12}$$

$$BS \rightarrow A: \mathrm{E}_{K_{Authi+1}}(K_{DSi+1}||i+1||\Delta_{i+1}), \quad T_{BS} = T_{BSi+1}, T_A = T_{Ai+1} \tag{13}$$

$$\Delta_{Ai+1} = T_{Ai+1} - T_{Ai} \approx T_{BSi+1} - T_{Xi} \approx \Delta_{i+1} - t \neq \Delta_{i+1} \tag{14}$$

Previous studies have shown that nodes can detect jamming [43]. So, when a node detects it is being jammed, it delays authentication of new nodes until next cycle. If node A calculates wrong time difference on the next cycle, it rejects authentication.

Replay and wormhole attack:

Another more sophisticated attack may combine previous attack with wormhole attack [44]. Let us assume that adversary has another node near BS, also his nodes are equipped with powerful transmitter and another frequency band for exclusive communication. Now, node X gets K_{Authi} key as soon as it is revealed, so instead of (8) he would have $\Delta'_i = (T_{Xi} - T_{Xi-1}) \approx \Delta_i$. A small modification of the protocol can render this attack obsolete. Nodes will terminate every cycle t seconds before message (T3), where t is the estimated time for message (T2) to reach all nodes of the network.

5.1 Simulations Parameters

To investigate different properties of the proposed method a series of simulations were conducted. To this end, different numbers of nodes were uniformly distributed over a square field of 500×500 meters. Furthermore, transceiver range of nodes was assumed to be 30 meters. Each simulation is run for 100 times, and then the final results are achieved by averaging over them. Table 3 shows average number of neighbors and average number of isolated nodes. It is noteworthy that isolated nodes were omitted from the rest of simulations.

Table 3. Simulations parameters

Network Size N	Neighbor No	Isolated No
500	5.33	3.61
600	6.39	1.85
700	7.4	1
800	8.52	0.65
900	9.57	0.19
1000	10.65	0.11

5.2 Number of Neighbors Analysis

WSNs highly rely on co-operation between nodes. They use it for sending messages back and forth, for localization, routing and etc. Therefore, determining number of neighbors is of great importance. In this section a mathematical approach for calculating this parameter is pursued.

Theorem I:

If nodes are uniformly distributed over a unit square, cumulative distribution function (CDF) of distance between node C and other nodes:

(1) Depends on the position of node C relative to the borders of network.
(2) It is calculated according to the Equation 15.

Corollary 1:

If N nodes with radio range of r are uniformly distributed over a square region of $a \times a$ then on average each node has $N \times F_Z(r/a)$ neighbors. Figure 5, shows how average number of neighbors varies with the size of networks and position of node (C). According to Figure 5, simulation results concur with results of corollary 1.

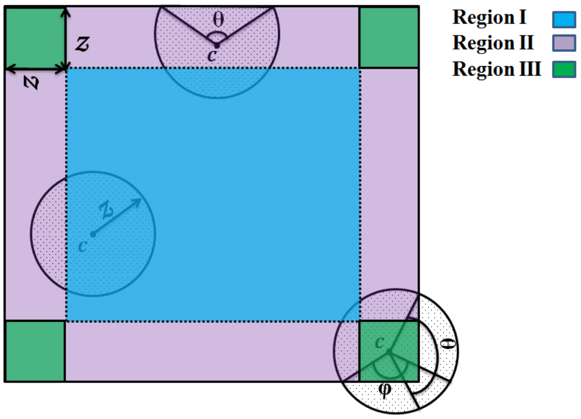

Figure 4. Different regions of the network.

$$F_Z(z) = \begin{cases} \pi z^2 & c \in RegionI \\ \pi z^2 - \frac{z^2}{2}(\theta - sin\theta) & c \in RegionII \\ \pi z^2 - \frac{z^2}{2}\left(\theta - \frac{sin\theta}{2}\right) - \frac{z^2}{4}(\varphi - sin\varphi) & \\ -\frac{z^2}{2}tan^{-1}\left(\cot\frac{\theta}{2}\right) + z^2 cos\frac{\varphi}{2}cos\frac{\theta}{2} & \\ & c \in RegionIII \end{cases} \qquad (15)$$

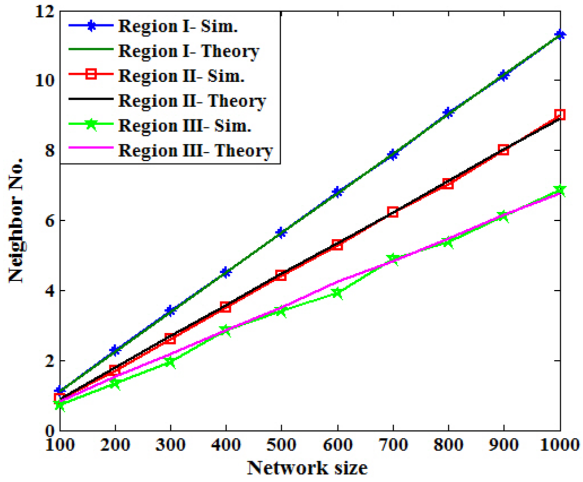

Figure 5. Theoretical number of neighbors vs. simulation results.

5.3 Connectivity of the Proposed Method

In the proposed scheme, it was assumed that every node receives messages of BS. In this section validity

Table 4. Signatures life-time

MAC size (Bit)	uniform timing			non-uniform timing		
Memory (KiB)	32	64	128	32	64	128
1	10.6	5.3	2.6	61.6	29.6	13.6
2	21.3	10.6	5.3	125.6	61.6	29.6
4	42.6	21.3	10.6	253.6	125.6	61.6
8	85.3	42.6	21.3	509.6	253.6	125.6

of this assumption is scrutinized. Let us assume that BS is not equipped with a high power transmitter and it is an ordinary node. In this scenario, nodes of the network will employ a method like blind flooding [45] to relay BS messages for other nodes.

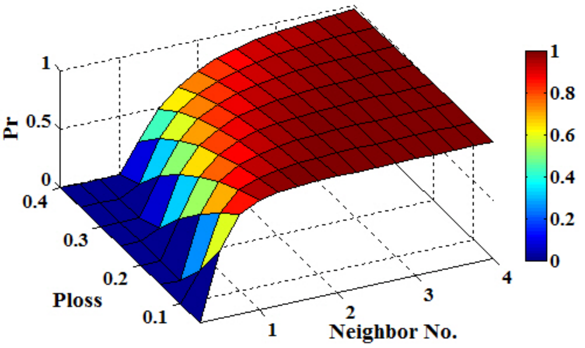

Figure 6. Probability of receiving BS messages.

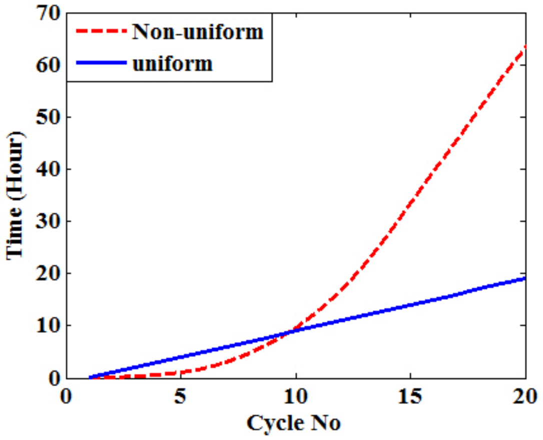

Figure 7. Two different timing schedules.

Theorem II:

If probability of packet loss is equal to p_{loss} and node C has k neighbors, then probability of receiving BS messages (p_r) satisfies Equation 16.

$$p_r = 1 - {p_{loss}}^{k.p_r} \tag{16}$$

A numerical method was conducted to solve Equation 16. Figure 6, shows result of this analysis.

Theorem III:

If two nodes participate in m authentication cycles, then probability of sharing a key is equal to:

$$P_m = 1 - (1 - p_r^4)^m \tag{17}$$

Equation 17 is a function of m and p_r. Also, according to theorem 2, p_r varies with the number of neighbors and the probability of packet loss. In order to show how value of Equation 17 varies, a four dimensional plot is employed, in which axes correspond to variables and value of p_m is depicted by color. As the color goes from blue (corresponding to value of 0) to red (corresponding to value of 1), it means higher value of probability.

According to Figure 8, it can be deduced that:

(1) If BS has a low power transmitter, then connectivity of the network will be highly affected by location of BS in the network. Furthermore, this effect becomes more obvious as the packet loss probability increases and node density decreases.

(2) If average number of neighbors is greater than 10, then every node will share a key with its neighbors after first cycle.

5.4 Memory Overhead of the Proposed Method

According to the length of MAC and available memory of nodes, every node can be preloaded with a specified number of signatures. This number, in accordance with BS timing schedule, determines life time of the proposed scheme. Two different timing schedules are considered, namely uniform and non-uniform timing schedule. It is assumed that in uniform one, BS initiates a new cycle every hour, whereas in non-uniform schedule time difference between consecutive cycles increases from 5 minutes to 360 minutes. Figure 7, depicts these two timing schedules.

Based on the available memory and length of MAC, life time of the proposed method can be calculated. Table 4 presents these results in days.

We conclude this section with some points:

(1) Because signatures are used once, after each

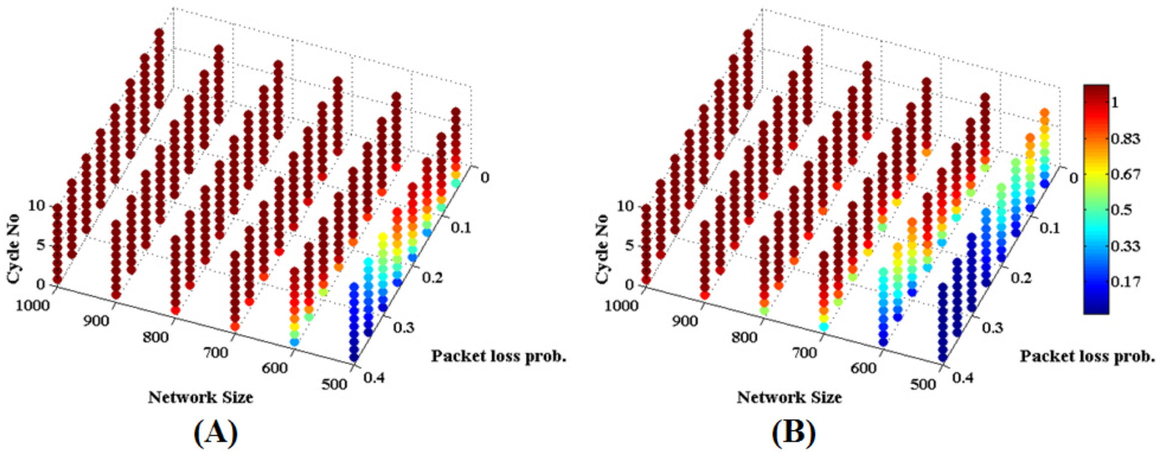

Figure 8. Connectivity of the proposed method after m authentication cycles. (A: BS is in region I, B: BS is in region III)

key is revealed corresponding signature may be deleted to free memory.

(2) Calculated lifetime indicates period of time that network deems corresponding public keys as valid.

(3) Timing schedule should be unpredictable for adversary, so that he cannot jam the network at critical moments, like when BS is broadcasting its messages.

(4) In highly dynamic networks, configuration of network changes rapidly (VANETs for example). Therefore, neighbors of nodes are constantly changing and new keys are constantly needed. This calls for short cycle duration. It is possible that in such a network memory overhead of the proposed method exceeds memory of the nodes. Consequently, proposed method suits networks with slow rate of network reconfigurations.

5.5 Scalability

To compare scalability of different PKC based key management systems, it is assumed that nodes have 64KiB of memory on board, ECC-160 is used and node ID is 2 bytes in length.

DAS scheme pre-loads every node with public key and ID of all other nodes. Consequently, it can supports up to 2978 nodes. CAS can support at most 32768 users, because only ID of revoked nodes should be stored. In BAS for a false positive probability equal to $2^{-63.8}$, network can accommodate up to 3948 nodes [29]. Assuming SHA-1 is used, HAS method can double the size of BAS method at cost of 20 more bytes message overhead. Also, it can support quadruple nodes at cost of 40 more bytes communication overhead [29].

Proposed method (designated by BA) has the same characteristics as CAS, thus only ID of the revoked nodes should be stored. These results are presented in Table 5.

Table 5. Maximum number of nodes

Scheme	Max. size
DAS	2978
CAS	32768
BAS	3948
D-HAS	7896
Q-HAS	15792
BA	32768

5.6 Energy Consumption

It was reported in [19], a Chipcon CC1000 radio used in Crossbow MICA2DOT mote consumes 28.6 and 59.2 μJ to receive and transmit one byte. Our simulation used a packet size of 41 bytes, 32 bytes for the payload and 9 bytes for the header [19]. Also, 128 bits for signature, 14 bits for time difference (to account for maximum time difference of 9.1 hours with resolutions of 2 s.), 10 bits for cycle number, and 128 bits for key are used. In addition, first 16 bits of the key are exploited for acknowledgment message.

To calculate energy consumption of different methods, authentication parts of Table 2 were replaced. Then, energy cost of transmission and executing cryptographic primitives [19, 46] are added. Table 6 presents details of communication and computation costs for different PKC-based key management systems.

In DAS, nodes just send their ID and then run two last part of Table 2. Therefore, it will cost 47.5 mj. According to [19], CAS consumes 187.6 mj energy. In BAS method, for a false positive probability of $2^{-63.8}$

Table 6. Energy overhead of different methods

Method	Communication cost (Bytes)	Computation cost
DAS	2 (ID)+T4	ECDH
CAS	86 (CERT) + T4	CERT verify + ECDH
BAS	20 (P_{ux}) + T4	Bloom + ECDH
D-HAS	20 (P_{ux})+20 (AAI) + T4	Bloom +1 SHA1 + ECDH
Q-HAS	20 (P_{ux})+40 (AAI) + T4	Bloom +2 SHA1 + ECDH
BA	T1 + T2 + T3 + T4	1 SHA1+1 AES+1 HMAC + ECDH

Table 7. Total energy consumption (mj)

Scheme	Energy cost
DAS	47.5
CAS	187.6
BAS	66.17
D-HAS	71.51
Q-HAS	75.26
BA	58.68

we should have $m/N = 92$. Also, checking bloom filter needs executing $m.ln2/N$ hash functions [29]; therefore, its energy cost will be 66.17 mj. HAS doubles the network size, at cost of 20 extra bytes of message overhead and an extra hash function operation. Thus, its energy cost would add up to 71.51 mj. For a quadruple size network, this cost will raise up to 75.26 mj. Table 7 presents these results. To put these numbers into perspective, total energy consumption of different methods for establishing a key between every node and all its neighbors were calculated. Figure 9 shows result of this simulation for different network sizes.

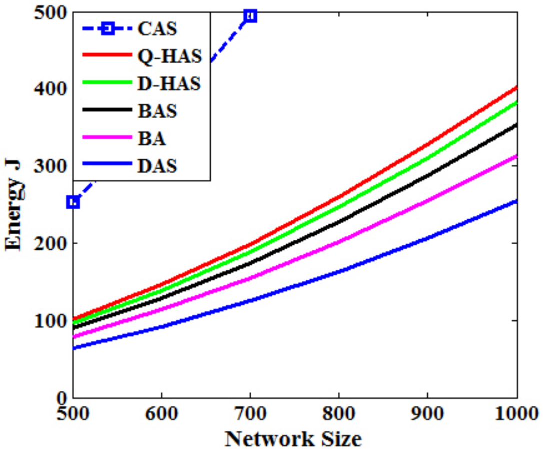

Figure 9. Comparing total energy consumption of PKC based key management system.

6 Discussion

According to [16] PKC-based key management systems have two major drawbacks to be used in WSNs. First, they are slow, and second they are not energy efficient. This paper tried to alleviate problem of energy deficiency. To this end, checking digital certificate was replaced with a symmetric based protocol. Simulation results showed that proposed method has many favorable characteristics.

The main concern of WSN is energy consumption. According to Figure 9, DAS method has the lowest energy consumption. Unfortunately, there are numerous problems with this system. According to Table 5, it supports the least number of nodes. Furthermore, adding new nodes to the network is almost infeasible. Thus, it can be inferred that DAS method is not a practical system. Therefore, according to Figure 9, among practical PKC-based key management systems, proposed method has lowest energy consumption.

Different applications of WSNs lead to networks with different number of nodes. So, key management system should support different network sizes. Consulting results of Table 5, proposed scheme supports the largest size of network.

Applications of WSNs are very wide, so their protocols should be flexible and provide different tradeoffs. Proposed method benefits from different possible trade-offs among memory, energy cost, and security level. For example, security level could be lowered (by using a shorter signature) in favor of reducing memory overhead or energy consumption of the system.

Key management is the heart of a secure communication system. Therefore, its security is vital. Considering this fact, it was shown that proposed method is secure and it can withstand different attacks including complex scenarios of replay attacks.

In WSNs, nodes are unattended and sometimes are deployed in hostile environments. Dead nodes in these networks may lead to many security threats. Adversary could collect these nodes and extract their cryptographic materials, such as their keys and their digital signatures. Then, he could use these data to program his own nodes and mount more effective attacks. A secure key management system, should address this problem as well. According to Table 4 a life time is assigned to every public-private key. When all tickets of a node get expired, its public-private key would be of no further use. Therefore, every node could be preloaded with proper number of signatures such that when its battery drains up there is not any valid signature left.

Proposed scheme relies on a broadcast authentication protocol to work. To achieve this goal, μTesla protocol was employed. This protocol is based on delayed disclosure of symmetric keys. Therefore, it is very energy efficient. On the other hand, authentication is delayed until the corresponding key is disclosed. This could make system vulnerable against denial of service attacks. To solve this vulnerability of μTesla different mechanisms such as multiple buffers random selection mechanism may be employed [33].

Another concern of the proposed method was reachability of BS for nodes of the network. Assuming that BS is equipped with a powerful transmitter is a prevalent notion in WSNs literature. Nevertheless, it was shown that this is not a prerequisite for the proposed method. According to analysis of Section 5.4, even if BS is an ordinary node, connectivity of the proposed method is satisfactory. It is noteworthy that in this situation connectivity of the network is affected by position of BS. According to corollary 1, the number of neighbors is related to the position of the node relative to the boarder of the network. Therefore, if BS is in region III, then it would have fewer neighbors. Thus, if value of packet loss is high, there would be good chances that message of BS do not propagate in the network.

In the proposed method, BS manages timing of authentication cycles; it can thus control key management system more efficiently. For example, if BS learns that adversary is doing energy exhaustion attack, he can prolong the cycles to reduce the effects of attack. Also, if new nodes are going to be added to the network, BS can initiate new cycles so that new nodes connect to the network much faster.

7 Conclusion

Although, it is known that PKC-based key management systems offer perfect resiliency, their energy consumption is a heavy burden on tight resources of nodes. Therefore, reducing energy consumption of PKC-based key management is desirable. Pursuing this goal, we proposed a novel PKC-based key management system based on broadcast messages from BS. Simulations results demonstrated that comparing to traditional certificate based systems; proposed method reduces energy cost of PKC-based system more than three times, while other desirable characteristics of PKC-based systems are maintained. Also, we showed that tagging messages with time-differences between consecutive cycles, removes the time synchronization need of μTesla based protocols. Furthermore, to solve vulnerability of dead nodes and cryptographic materials stored in them, we showed that proposed method intrinsically assigns a life time to every public-private key and it has the potency to solve this problem.

Acknowledgement

The authors would like to thank the anonymous reviewers for their constructive comments and valuable contributions. This work was partially supported by Iranian National Science Foundation (INSF) grant no. 92/32575.

References

[1] I. Akyildiz, W. Su, Y Sankarasubramaniam, and E. Cayirci, “A Survey on Sensor Networks”, IEEE Commun. Mag., vol. 40, no. 8, pp. 102-116, Aug. 2002.

[2] A.D. Wood and J.A. Stankovic, “Denial of service in sensor networks,” IEEE Computer, vol. 35, no. 10, pp. 5462, Oct. 2002.

[3] Y. M. Huang, M. Y. Hsieh, H. C. Chao, S. H. Hung, and J. H. Park, “Pervasive, Secure Access to a Hierarchical Sensor-Based Healthcare Monitoring Architecture in Wireless Heterogeneous Networks,” IEEE Journal on Selected Areas in Communications, vol. 27, no. 4, pp. 400-411, May 2009.

[4] L. Buttyan, D. Gessner, A. Hessler, and P. Langendoerfer, “Application of wireless sensor networks in critical infrastructure protection: challenges and design options [Security and Privacy in Emerging Wireless Networks],” IEEE Wireless Communications, vol. 17, no. 5, pp. 44-49, Oct. 2010.

[5] G. Wener-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, “Deploying a wireless sensor network on an active volcano,” IEEE Internet Computing, vol. 10, no. 2, pp. 18-25, March-April 2006.

[6] I.F Akyildiz, E.P. Stuntebeck, “Wireless underground sensor networks: Research challenges,” Ad Hoc Networks, vol. 4, no. 6, pp. 669-686, Nov. 2006.

[7] J. Yick, B. Mukherjee, and D. Ghosal, “Analysis of a Prediction-based Mobility Adaptive Tracking Algorithm,” in: Proc. 2nd International Conference on Broadband Networks (BroadNet 2005), pp. 753-760, 2005.

[8] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J. Sallai, and K. Frampton, “Sensor network-based countersniper system,”

in Proc. 2nd. International Conference on Embedded Networked Sensor Systems (Sensys), 2004, pp. 1-12.

[9] Eschenauer L, Gligor BD, “A key-management scheme for distributed sensor networks,” in: Proc. of the 9th. ACM conference on computer and communication security, Washington, DC, USA, pp. 41-47, 2002.

[10] Liu D, Ning P, “Establishing pairwise keys in distributed sensor networks,” in: Proceedings of 10th. ACM conference on computer and communications security (CCS03), Washington, DC, pp. 41-47, 2003.

[11] Du W, Deng J, Han YS, Varshney P, Katz J, Khalili A, “A pairwise key pre-distribution scheme for wireless sensor networks,” ACM Transactions on Information and System Security (TISSEC), vol. 8, no. 2, pp. 228-258, 2005.

[12] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu and M. Galloway, “A survey of key management schemes in wireless sensor networks,” Computer Communications, vol. 30, no. 11-12, pp. 2314-2341, Sept. 2007.

[13] J. Zhang and V. Varadharajan, “Wireless sensor network key management survey and taxonomy,” Journal of Network and Computer Applications, vol. 33, no. 2, pp. 63-75, Mar. 2010.

[14] O. Delgado-Mohatar, A. Fuster-Sabater, J. Sierra, “A light-weight authentication scheme for sensor networks,” Ad Hoc Networks, vol. 9, no. 5, pp. 727-735, July 2011.

[15] M. Conti, R. Di Pietro , L. V. Mancini, and A. Mei, “Distributed Detection of Clone Attacks in Wireless Sensor Networks,” IEEE Trans. Dependable and Secure Computing, vo. 8, no. 5, pp. 685-698, Sept.-Oct. 2011.

[16] C. Alcaraz, J. Lopez, R. Roman, H. H. Chen, “Selecting key management schemes for WSN applications,” Computers & Security, vol. 31, no. 8, pp. 956-966, 2012.

[17] N. Gura, A. Patel, A. Wander, H. Eberle, and S. Shantz, “Comparing elliptic curve cryptography and rsa on 8-bit cpus,” in Proc. 6th. International Workshop on cryptographic hardware and embedded systems (CHES 2004), pp. 119-132, 2004.

[18] W. Du, R. Wang, and P. Ning, “An Efficient Scheme for Authenticating Public Keys in Sensor Networks,” in Proc. 6th.ACM international symposium on Mobile ad hoc networking and computing (MobiHoc’05), May 2005, pp.58-67.

[19] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, “Energy Analysis of Public-Key Cryptography on Small Wireless Devices,” in Proc. 3rd. IEEE international conference on pervasive computing and communications (PerCom 2005), 2005, pp. 324-328.

[20] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, “Energy Analysis of Public-Key Cryptography on Small Wireless Devices,” in Proc. 3rd. IEEE international conference on pervasive computing and communications (PerCom 2005), 2005, pp. 324-328.

[21] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, “TinyPK: securing sensor networks with public key technology,” in Proc. 2nd ACM workshop on Security of ad hoc and sensor networks (SASN ’04), 2004, pp. 58-64.

[22] M.L. Das, “Two-Factor User Authentication in Wireless Sensor Networks,” IEEE Trans. on Wireless Communications, vol. 8, no. 3, pp. 1086-1090, March 2009.

[23] K. Ren, W. Lou and K. Zeng, and P. J. Moran, “On Broadcast Authentication in Wireless Sensor Networks,” IEEE Trans. on Wireless Communications, vol. 6, no. 11, pp. 4136-4144, Nov. 2007.

[24] F. Hess, “Efficient identity based signature schemes based on pairings,” in Proc. 9th. Annual International Workshop on Selected Areas in Cryptography (SAC02), 2003, pp. 310324.

[25] 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. 14, pp. 659667, March 2008.

[26] K. Shim, Y. Lee, and C. Park, “EIBAS: An efficient identity based broadcast authentication scheme in wireless sensor networks,” Ad Hoc Networks, vol. 11, no. 1, pp. 182189, Jan. 2013.

[27] stanard specification of public key cryptography-Amendment 1: additional techniques, IEEE P1363a, 2004.

[28] C. H. Lim, “Secure Code Dissemination and Remote Image Management Using Short-Lived Signatures in WSNs,” IEEE Communications Letters, vol. 15, no. 4, pp. 362-364, April 2011.

[29] K. Ren, S. Yu, W. Lou, and Y. Zhang, “Multiuser broadcast authentication in wireless sensor networks,” IEEE Trans. on Vehicular Technology, vol. 58, no. 8, pp. 4554-4564, Oct. 2009.

[30] Y. Liu, J. Li, and M. Guizani, “PKC Based Broadcast Authentication using Signature Amortization for WSNs,” IEEE Trans. on Wireless Communications, vol. 11, no. 6, pp. 2106-2115, June 2012.

[31] H.Ghasemzadeh, M.R. ARef, and A. Payandeh, “A novel and low energy PKC-based key agreement protocol for WSNs,” International conference on information security and cryptology (ISCISC 13), 2013, pp. 1-6.

[32] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, “Efficient authentication and signing of multicast streams over lossy channels,” in Proc. IEEE Symposium on Security and Privacy (S&P 2000), 2000,

pp. 56-73.

[33] D. Liu, P. Ning, “Multilevel μTESLA: Broadcast authentication for distributed sensor networks,” ACM Transactions on Embedded Computing Systems, vol. 3, no. 4, pp. 800-836, Nov. 2004.

[34] J. Drissi, Q. Gu, “Localized broadcast authentication in large sensor networks,” In International conference on Networking and Services, 2006. ICNS’06, pp. 25-25.

[35] W. H. Chen, Y. J. Chen, “A C-μTesla Protocol for Sensor Networks,” Journal of Informatics & Electronics, vol. 2, no. 2, pp. 29-32, Nov. 2008.

[36] W. H. Chen, Y. J. Chen, “A bootstrapping scheme for inter-sensor authentication within sensor networks,” IEEE Communications Letters, vol. 9, no. 10, pp. 945-947, 2005.

[37] J. W. Kim, Y. H. Kim, H. Lee, D. H. Lee, “A practical inter-sensor broadcast authentication scheme,” in Proc. 4th international conference on Universal access in human computer interaction, 2007, pp. 399-405.

[38] D. Liu, P. Ning, S. Zhu, S. Jajodia, “Practical broadcast authentication in sensor networks,” The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2005, pp. 118-129.

[39] B. Bloom, “Space/time tradeoffs in hash coding with allowable errors,” Communication of ACM, vol. 13, no. 7, pp. 422426, July 1970.

[40] M. Mitzenmacher, “Compressed Bloom Filters,” IEEE/ACM Transactions on Networks, vol. 10, no. 5, pp. 604-612, Oct. 2002.

[41] R. Merkle, “Protocols for public key cryptosystems,” in Proc. IEEE Symposium Research Security Privacy, 1980, p. 122.

[42] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, “A pairwise key predistribution scheme for wireless sensor networks,” ACM Transactions on Information and System Security, vol. 8, no. 2, pp. 228259, May 2005.

[43] A. Wood, J. A. Stankovic, and S. H. Son, “JAM: A mapping service for jammed regions in sensor networks,” in Proc. 24th. IEEE Real-Time Systems Symposium (RTSS 2003), 2003, pp. 286-297.

[44] Y. C. Hu, A. Perrig, and D.B Johnson, “Packet leashes: A defense against wormhole attacks in wireless ad hoc networks,” in Proc. 22nd. annual joint conference of the IEEE computer and communication (Infocom 2003), 2003, pp. 1976-1986.

[45] J. M. McCune, E. Shi, A. Perrig and M. K.Reiter, “Detection of Denial-of-Message Attacks on Sensor Network Broadcasts,” in Proc. IEEE Symposium on Security and Privacy, 2005, pp. 64-78.

[46] N. R. Potlapally, S. Ravi, A. Raghunathan and N. K. Jha, “Analyzing the Energy Consumption of Security Protocols,” in Proc. 2003 international symposium on Low power electronics and design (ISLPED ’03), 2003, pp. 30-35.

Appendix. Proof of theorems

Proof of theorem I:

Let us assume that location of nodes follow uniform distribution.

$$X \sim U[0,1], \qquad Y \sim U[0,1] \tag{18}$$

We want to find neighbors of node C located at $c|_{y_c}^{x_c}$. Apparently, both $\Delta X = X - x_c$ and $\Delta Y = Y - y_c$ also follow uniform distributions.

$$f_{\Delta X}(\Delta x) \sim U[-x_c, 1-x_c], f_{\Delta Y}(\Delta y) \sim U[-y_c, 1-y_c] \tag{19}$$

Distance between node C and another node is equal to:

$$z = \sqrt{\Delta x^2 - \Delta y^2} \tag{20}$$

Now,

$$F_Z(z) = P(Z \leq z) = P(\sqrt{\Delta x^2 - \Delta y^2} \leq z) = \\ P(\Delta x^2 - \Delta y^2 \leq z^2) = \int\int f_{\Delta X, \Delta Y}(\Delta x, \Delta y) dxdy \tag{21}$$

If X and Y have independent distributions, then:

$$f_{\Delta X,\Delta Y}(\Delta x, \Delta y) = f_{\Delta X}(\Delta x).f_{\Delta Y}(\Delta y) = \\ \begin{cases} 1 & -x_c \leq \Delta x \leq 1 - x_c, -y_c \leq \Delta y \leq 1 - y_c \\ 0 & Otherwise \end{cases} \tag{22}$$

According to Figure 4, if location of node C falls in the region I, then:

$$F_Z^I(z) = \int_{-z}^{z} \int_{-\sqrt{z^2-\Delta y^2}}^{\sqrt{z^2-\Delta y^2}} f_{\Delta X,\Delta Y}(\Delta x, \Delta y) dxdy = \pi z^2 \tag{23}$$

If location of node C falls in the region II, then:

$$F_Z^{II} = F_Z^I(z) - \int_{-zsin\frac{\theta}{2}}^{zsin\frac{\theta}{2}} \int_{zcos\frac{\theta}{2}}^{\sqrt{z^2-\Delta x^2}} \\ f_{\Delta X,\Delta Y}(\Delta x, \Delta y) dydx = \pi z^2 - \frac{z^2}{2}(\theta - sin\theta) \tag{24}$$

If location of node C falls in the region III, then:

$$F_Z^{III} = F_Z^{II}(z) - \int_{-zsin\frac{\varphi}{2}}^{zsin\frac{\theta}{2}} \int_{-\sqrt{z^2-\Delta x^2}}^{-zcos\frac{\varphi}{2}}$$

$$f_{\Delta X,\Delta Y}(\Delta x, \Delta y)dydx = \pi z^2 - \frac{z^2}{2}(\theta - \frac{sin\theta}{2}) -$$

$$\frac{z^2}{4}(\varphi - sin\varphi) - \frac{z^2}{2}tan^{-1}(\cot\frac{\theta}{2}) + z^2cos\frac{\varphi}{2}cos\frac{\theta}{2} \quad (25)$$

Proof of theorem II:

If node C has k neighbors and probability of receiving BS message is equal to p_r then on average $k.p_r$ nodes of them have received message of BS. Therefore, probability of node C not receiving message of BS is equal to:

$$p_{fail} = {p_{loss}}^{k.p_r} \quad (26)$$

Therefore,

$$p_r = 1 - p_{fail} = 1 - {p_{loss}}^{k.p_r} \quad (27)$$

Proof of theorem III:

Two nodes can share a common key if both of them receive both (T2, T3). If every node receives message of BS with probability of p_r, then probability of both nodes receiving both of these messages is:

$$p_{Success} = p_r^2.p_r^2 = p_r^4 \quad (28)$$

Now, if two nodes participate in m authentication cycles, probability of sharing a key would be equal to:

$$P_m = 1 - (1 - P_{Success})^m \quad (29)$$

Hamzeh Ghasemzadeh was born in Tehran in 1984. He received his B.S. degree in Electrical Engineering from Ferdowsi University of Mashhad in 2007. He received his M.S. degree in Communications Engineering from Malek-e-Ashtar University of Technology in 2011. His primary research interests are multimedia security, computer forensic, pattern recognition, and machine learning. He is currently a lecturer at Electrical Engineering Department of Islamic Azad University of Damavand.

Ali Payandeh received the M.S. degree in Electrical Engineering from Tarbiat Modares University in 1994, and the Ph.D. degree in Electrical Engineering from K.N. Toosi University of Technology (Tehran, Iran) in 2006. From 1996 to 2006, he was a director of research at the Applied Science Research Association, Iran, where he was involved in research for secure satellite communications. He is now an assistant professor in the Department of Information and Communications Technology at Malek-e-Ashtar University of Technology, Iran. He has published more than 75 papers in international journals and conferences. His research interests include information theory, coding theory, cryptography, security protocols, secure communications, and satellite communications.

Mohammad Reza Aref received the B.S. degree in 1975 from the University of Tehran, Iran, and the M.S. and Ph.D. degrees in 1976 and 1980, respectively, from Stanford University, Stanford, CA, USA, all in Electrical Engineering. He returned to Iran in 1980 and was actively engaged in academic affairs. He was a faculty member of Isfahan University of Technology from 1982 to 1995. He has been a Professor of Electrical Engineering at Sharif University of Technology, Tehran, since 1995, and has published more than 290 technical papers in communications, information theory and cryptography in international journals and conferences proceedings. At the same time, during his academic activities, he has been involved in different political positions. First Vice President of I.R. Iran, Vice President of I.R. Iran and Head of Management and Planing Organization, Minister of ICT of I.R. Iran and Chancellor of University of Tehran, are the most recent ones. His current research interests include areas of communication theory, information theory, and cryptography.