Securing Mobile Ad Hoc Networks Using Enhanced Identity-Based Cryptography

Kamal Adli Mehr and Javad Musevi Niya

Recent developments in identity-based cryptography (IBC) have provided new solutions to problems related to the security of mobile ad hoc networks (MANETs). Although many proposals to solve problems related to the security of MANETs are suggested by the research community, there is no one solution that fits all. The interdependency cycle between secure routing and security services makes the use of IBC in MANETs very challenging. In this paper, two novel methods are proposed to eliminate the need for this cycle. One of these methods utilizes a key pool to secure routes for the distribution of cryptographic materials, while the other adopts a pairing-based key agreement method. Furthermore, our proposed methods utilize threshold cryptography for shared secret and private key generation to eliminate the "single point of failure" and distribute cryptographic services among network nodes. These characteristics guarantee high levels of availability and scalability for the proposed methods. To illustrate the effectiveness and capabilities of the proposed methods, they are simulated and compared against the performance of existing methods.

Keywords: Identity-based cryptography, threshold cryptography, pairing-based cryptography, key pool, security-routing interdependency.

I. Introduction

Security of mobile ad hoc networks (MANETs) remains a major concern in both academia and industry. Despite years of research, there is no mature solution to the problem of security of MANETs that is widely accepted, and the growing availability of small, personalized mobile devices with peer-to-peer communication capability through wireless channels makes this problem even more important.

Proven security mechanisms that are widely used in wired networks are not always applicable to MANETs. Detectable intrusions in wired networks have caused big security challenges in MANETs.

W. Li and A. Joshi [1] completely characterized MANETs and discussed their pros and cons. Of the characteristics described in [1], the following make it particularly difficult to achieve security requirements: there is no network infrastructure or online administration available; dynamic network topology and node membership mechanism; possibility of an insider attack; constrained computational and communicational resources; and vulnerabilities of wireless channels. Due to the aforementioned characteristics, new security methods that are proposed for use with MANETs are required to guarantee availability, authentication, confidentiality, integrity, and so on [1]–[2].

Early security proposals in MANETs were designed to combat specific kinds of security attacks. These early proposals often recommended solutions that were designed to target several security attacks (including the specific kind of security attach being addressed at the time) at once. Such solutions would work well against designated attacks but would still be vulnerable to collapse under combined or unanticipated attacks.

Later security proposals incorporated cryptography into their

Manuscript received Feb. 15, 2014; revised Nov. 29, 2014; accepted Dec. 11, 2014.

Kamal Adli Mehr (corresponding author, k.adlimehr@tabrizu.ac.ir) and Javad Musevi Niya (niya@tabrizu.ac.ir) are with WiLap, the Faculty of Electrical and Computer Engineering, University of Tabriz, East Azerbaijan, Iran.

general design framework to achieve a general sense of security. There are two main categories of cryptography techniques used in MANETs — symmetric key and asymmetric key.

Identity-based cryptography (IBC) is a subclass of public key cryptography. IBC was designed to eliminate the need for a certification authority (CA) and public key certificates (PKCs) [3]. IBC was first proposed by A. Shamir [4] in 1984. In IBC, for a given user, both the user's public key and the user's private key are based on the identity of the user. In an IBC scheme, a user's public key is an easily calculated function of their identity (for example, the user's IP address, phone number, or e-mail address), while a user's private key can only be calculated by a trusted authority — that is, a private key generator (PKG). In comparison with a traditional public key infrastructure (PKI), an IBC is not required to store and transmit large volumes of public keys and certificates; thus, it is a far more attractive option to be used in MANETs.

In this paper, two novel methods are proposed to eliminate the interdependency cycle between secure routing and security services. One of these methods utilizes a key pool to construct secure routes for the distribution of cryptographic materials, while the other one is based on pairing-based key agreement. Furthermore, the proposed methods utilize threshold cryptography for shared secret and private key generation to eliminate the single point of failure and distribute cryptographic services among network nodes. These characteristics guarantee high levels of availability and scalability for the proposed methods.

The rest of this paper is organized as follows. Section II provides some related work. Section III gives the detailed procedure of the proposed methods. The simulation results and discussions are presented in Section IV. Finally, Section V concludes the paper.

II. Related Work

Designing efficient key management (KM) mechanisms is of paramount importance to those aiming to provide security services in MANETs, because security services that employ cryptographic techniques rely on such mechanisms. KM is the process that specifies how to distribute cryptographic keys to network nodes and update (if required), revoke, and so on such keys [5]. There is a rich literature on KM methods. Investigations by the authors in [6], within those publications that were available to them, have led to the classification of the current KM protocols into the following subsets: partially distributed certificate authority, fully distributed certificate authority, identity-based KM, certificate chaining-based KM, cluster-based KM, predeployment-based KM, mobility-based KM, and parallel KM. Each of these categories has its own pros and cons, but as mentioned in [7], the spectacular characteristics of IBC make it an ideal choice for utilization in MANETs. Some of these characteristics are mentioned in the following. First, IBC does not utilize certificates; hence, there is no need for certificate distribution and verification, which saves on communication and computation overheads (especially in large-scale MANETs). Second, IBC offers non-interactive key agreement. Finally, the public keys provided by IBC are self-proving and can carry much useful information.

KM in IBC consists of key generation and distribution methods and, ideally, key protection and revocation. There is rich literature about KM using IBC [8]–[19]. Most of the schemes within these literatures are derived from and are variants of [20] and are thoroughly investigated in [3]. These schemes suffer from some common drawbacks. For example, suppose a node in a network can communicate with a PKG to obtain its private key; then, a major problem is the secure transmission of the private key from the PKG to the requesting node.

In the aforementioned proposals, there are no shared secrets between PKGs and nodes (that is, common symmetric keys), nor do nodes have public/private key pairs. Therefore, it is not exactly determined how a node should obtain its private key from a PKG in the presence of an intruder. This problem can only be solved by means of a secure channel or pre-distributed common keying material, neither of which is ideal in ad hoc networks [6]. Furthermore, the aforementioned proposals have a common problem — security-routing interdependency [9]. In these approaches, KM relies on secure routing to establish secret keys, while behind such secure routing is the assumption that secret keys are pre-deployed to set up a routing table.

Recently, S. Zhao and others [21] proposed a KM and secure routing (KM-SR) integrated framework that utilizes system parameters of IBC to derive node-specific broadcast keys. In this proposal, public channels are used to distribute system parameters to authenticated nodes. The integrated node-specific broadcast keys are generated, and secure routing is set up by means of system parameters. The authors of [21] claim that because the authenticated distribution of system parameters and routing setup are all carried out through public channels and generation of integrated node-specific broadcast keys does not require any extra communication between nodes, there is no KM-SR interdependency cycle. However, despite the fact that the proposed method in [21] utilizes the novel idea of a KM-SR integrated framework, it cannot satisfy the required conditions of an appropriate KM system for MANETs. The major problem of this method is its deployment of a centralized PKG. Although, this avoids the problems that are caused by threshold cryptography, the fact still remains that the

centralized PKG can be a single point of failure and an ideal target for intruders. Furthermore, the deployment of a centralized PKG reduces the scalability of the system; thus, the use of this method in large-scale networks is infeasible. This method also lacks any sort of revocation or key update mechanism. The proposed method in [21] is an ideal choice for trusted and authenticated networks that have a trusted authority to act as a PKG; however, this cannot take away the fact that it fails to meet the security requirements of MANETs.

III. Proposed Methods

The proposed methods in this paper consist of the following five stages: initialization, shared secret generation, distributed master-key generation, private-key generation, and generation of a new master key for new-comer nodes and existing nodes that want to become a new PKG (Fig. 1). These methods utilize a key pool and pairing-based key generation techniques to eliminate the KM-SR interdependency cycle. To distribute a PKG task among network nodes, A. Shamir's [22] secret-sharing technique, with some slight modifications to it, is used.

1. Initialization Phase

Let p and q denote two large primes and $E/\mathbb{F}_p$ an elliptic curve of the form $y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_p$. Let G_1 be a q-order subgroup of the additive group of points over the elliptic curve and G_2 be a q-order subgroup of the multiplicative group of the finite field $\mathbb{F}^*_{p^2}$. It is assumed that the discrete logarithm problem (DLP) is hard over both G_1 and G_2. The proposed IBC system deploys the following bilinear mapping $e: G_1 \times G_1 \to G_2$. This mapping has the following properties [17]:

- *Bilinear.* The $e: G_1 \times G_1 \to G_2$ mapping is *bilinear* if for all $P, Q \in G_1$, and $a, b \in Z$ equation $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ holds.
- *Non-degenerate.* The mapping does not transform all the possible values of $G_1 \times G_1$ to the identity. Note that because G_1 and G_2 are prime groups, non-degeneracy implies that if P is a generator of G_1, then $\hat{e}(P, P)$ is a generator of G_2.
- *Computable.* For all possible values of $P, Q \in G_1$, there is an efficient algorithm to compute $\hat{e}(P, Q)$.

A bilinear map that satisfies the three aforementioned constraints is called an *admissible* bilinear map. According to the fact that $\hat{e}$ is bilinear and that G_1 is a cyclic group, it can be deduced that $\hat{e}$ is symmetric; that is, for all $P, Q \in G_1$, equation $\hat{e}(P, Q) = \hat{e}(Q, P)$ holds. Examples of such bilinear maps are the modified Weil pairing and the Tate pairing, for which the bilinear Diffie–Hellman problem (BDHP) is believed to be hard [17]. The algorithm $\mathcal{G}$ that is used to generate pairing parameters acts as follows:

1. For security parameter $k \in Z^+$ as its input, $\mathcal{G}$ generates a prime number (q); G_1 and G_2 (of order q); and an admissible bilinear map $\hat{e} = G_1 \times G_1 \to G_2$ according to the above constraints.
2. Generate the hash function $H_1 : \{0,1\}^* \to G_1$.
3. Choose P as the generator of G_1.
4. Public system parameters $\langle P, q, e, H_1, G_1, G_2 \rangle$ are available from the beginning for all.

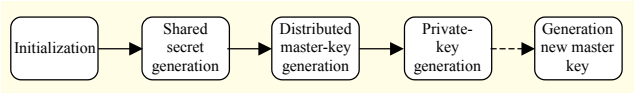

Fig. 1. Steps of proposed method.

2. Shared Key Generation

To operate properly in MANETs, IBC relies on a shared-key generation phase. In this phase, network nodes that have never met develop a shared key, and based on this shared key, further secure communications become feasible. For this phase, we developed two fundamentally different methods to achieve this goal. One of these methods is based on a key pool and uses several large pools of keys that are implemented in network nodes, whereas the other method generates a shared key based on public system parameters.

A. Key Pool–Based Method

The key pool–based method utilizes a variant of the KM scheme proposed by L. Eschenauer and V.D. Gligor [23]. This method consists of the following two stages: key pre-distribution and shared key discovery. The key pre-distribution stage is comprised of several offline phases, as follows:

- Offline authority A generates a large pool of K keys and their corresponding key identifiers.
- For each node, A randomly selects k keys without replacement and preloads them along with their respective key identifiers. These keys then form a key ring.

The shared key discovery phase begins after deployment of the nodes. Each node broadcasts its key identifiers without any form of encryption. The nodes that are in radio range of each other discover the shared keys by comparison. However, it is possible that nodes discover shared keys privately by hiding their key sharing pattern. For example, each node broadcasts a and $E_{K_i}(\alpha)$, where α is a random challenge and K_i for $i = 1, 2, \ldots, k$ represents all of the keys in the key ring of the node. The decryption of $E_{K_i}(\alpha)$ using the right key unveils

α and establishes a shared key with the transmitter [23].

B. Pairing-Based Method

The second method to establish a secure channel for transmission of IBC parameters utilizes pairing-based key agreement techniques. Similar to [21], the proposed method uses pairing-based techniques to set up secure links between all network nodes. The algorithm of the proposed pairing-based method is as follows:

- Each node C_i chooses a random value $s_i^0 \in Z_q$ and then calculates $P_i^{\text{pub0}} = s_i^0 P$.
- Then C_i broadcasts P_i^{pub0} to its neighbors.
- Node C_i can encrypt its data using $k_{ij}^0 = \hat{e}(s_i^0 . P_j^{\text{pub0}}, P)$. It can then send the encrypted data to C_j.
- Node C_j can decrypt the encrypted data by $k_{ji}^0 = \hat{e}(s_j^0 . P_i^{\text{pub0}}, P)$.

It is easy to show that $k_{ij}^0 = k_{ji}^0$. This can be proved by manipulating the properties of the admissible bilinear mapping.

$$\begin{aligned} k_{ij}^0 &= \hat{e}\left(s_i^0 . P_j^{\text{pub0}}, P\right) = \hat{e}\left(s_i^0 . s_j^0 P, P\right) \\ &= \hat{e}\left(s_j^0 . s_i^0 P, P\right) = \hat{e}\left(s_i^0 . P_i^{\text{pub0}}, P\right) = k_{ji}^0. \end{aligned} \quad (1)$$

Therefore, in this method, nodes share a common key without any additional communication.

3. Distributed Master-Key Generation

The proposed master-key generation mechanism does not rely on a trusted third party to securely generate a master key, divide a master key into shares, and distribute the shares among PKGs. A master key is calculated in a distributed fashion through the collaboration of n initial nodes. Similar to [11], the proposed method utilizes a variant of Shamir's secret-sharing method [16] that does not need a trusted authority. The proposed method uses the following algorithm:

- Node C_i chooses two secrets, x_i and $\overline{x}_i$, and polynomials $f_i(z)$ and $\overline{f}_i(z)$ of order $k-1$ over Z_q in such a way that $f_i(0) = x_i$ and $\overline{f}_i(0) = \overline{x}_i$.
- Node C_i calculates the sub-share of node C_j using $\text{ss}_{ij} = f_i(j)$ and $\overline{\text{ss}}_{ij} = \overline{f}_i(j)$ for $j = 1, 2, \ldots, n$ and sends them to C_j through the secure channels that are established in the shared-key generation phase.
- After reception of $n-1$ sub-shares, C_j can calculate its own share of the master private key using $S_j = \sum_{i=1}^{n} \text{ss}_{ij} = \sum_{i=1}^{n} f_i(j)$ and $\overline{S}_j = \sum_{i=1}^{n} \overline{\text{ss}}_{ij} = \sum_{i=1}^{n} \overline{f}_i(j)$; that is, the master-private-key share of node C_j is a combination of all received sub-shares from the initial n nodes.

Similar to Shamir's secret-sharing method [22], any k set of shareholders can reconstruct the secret using $\sum_{i=1}^{k} S_i l_i(z) \bmod q$ and $\sum_{i=1}^{k} \overline{S}_i l_i(z) \bmod q$, in which $l_i(z)$ are Lagrange coefficients that can be calculated using

$$l_i(z) = \prod_{j=1, j\neq i}^{k} \frac{z-j}{i-j}. \quad (2)$$

Two master private keys are $\text{SK} = \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} f_i(0)$ and $\overline{\text{SK}} = \sum_{i=1}^{n} \overline{x}_i = \sum_{i=1}^{n} \overline{f}_i(0)$; although, they are not reconstructed in any single node.

4. Private-Key Generation

The mechanism that is used for identity-based public/private-key generation for each node C_i, whether it is a PKG or not, is very important. Similar to [17], the proposed method consists of a number of continuous, non-overlapping key update phases that are denoted by p_m for $1 \le m < M$, where M is the maximum index of a phase. Each phase p_m is associated with a binary string, called a phase identifier and denoted by salt_m. In the initialization phase, a random seed, salt_1, is preloaded to every node. After deployment, each node can determine phase identifiers using $\text{salt}_m = \text{salt}_{m-1} + 1$ $(1 < m < M)$. In the proposed method, each public/private key pair is both node-specific and phase-specific. The key pair of node C_i in phase p_m is denoted by $\left\langle \mathcal{K}_{C_i,p_m}, \mathcal{K}_{C_i,p_m}^{-1} \right\rangle$. Each of $\mathcal{K}_{C_i,p_m}$ and $\mathcal{K}_{C_i,p_m}^{-1}$ is comprised of both a node-specific element and a phase-specific element. Furthermore, similar to [11] and unlike common IBC systems, a node-specific public key, and subsequently a private key, is a function of a node ID and MAC address so as to bind the identity of the node to its MAC address, which is assumed not to change during the lifetime of a network. The public/private key pair is calculated by

$$\begin{cases} \mathcal{K}_{C_i,p_m} = \left(\mathcal{K}_{C_i}, \mathcal{K}_{p_m}\right) = \left(H_1\left(\text{ID}_{C_i} \,\|\, \text{MAC}_{C_i}\right), H_1\left(\text{salt}_m\right)\right), \\ \mathcal{K}_{C_i,p_m}^{-1} = \left(\mathcal{K}_{C_i}^{-1}, \mathcal{K}_{p_m}^{-1}\right) = \left(\text{SK} \times H_1\left(\text{ID}_{C_i} \,\|\, \text{MAC}_{C_i}\right), \overline{\text{SK}} \times H_1\left(\text{salt}_m\right)\right). \end{cases} \quad (3)$$

Note that (3) is used only for public key calculation. A private key cannot be calculated using this equation because (3) uses SK and $\overline{\text{SK}}$; these are parameters that should not be reconstructed in any single node. Instead, to calculate a private key in a distributed fashion, the requesting node communicates with k PKGs and requests a private key share. After receiving the request, the PKG node C_j calculates the private key share of

C_i using $\mathcal{K}_{ji,p_m}^{-1} = f_{\text{extract}}\left(S_j, \overline{S}_j, (\text{ID}_{C_i} \,||\, \text{MAC}_{C_i}), \text{salt}_m\right) = (S_j \mathcal{K}_{C_i}, \overline{S}_j \mathcal{K}_{p_m})$ and sends it to C_i over the secure channels that are provided by the shared key generation phase. In this equation, f_{extract} denotes the process of private key share generation in PKG; S_j and $\overline{S}_j$ are master private key shares of C_j; and $\mathcal{K}_{C_i}$ and $\mathcal{K}_{p_m}$ are, respectively, node-specific and phase-specific public keys of C_i. Using (4), node C_i can calculate the private key after receiving k shares as follows:

$$\begin{cases} \mathcal{K}_{C_i}^{-1} = \prod_{j=1}^{k} S_j \mathcal{K}_{C_i}, \\ \mathcal{K}_{p_m}^{-1} = \prod_{j=1}^{k} \overline{S}_j \mathcal{K}_{p_m}, \end{cases} \tag{4}$$

where $\left\langle \mathcal{K}_{p_m}, \mathcal{K}_{p_m}^{-1} \right\rangle$ are public/private key elements corresponding to phase p_m and $\left\langle \mathcal{K}_{C_i}, \mathcal{K}_{C_i}^{-1} \right\rangle$ are public/private key elements corresponding to node C_i.

5. Generating New Master Key Share

Nodes in an ad hoc network may leave the network or enter it randomly; subsequently, this may mean that the number of PKGs in the network at any one time is less than a certain threshold value (k). Therefore, a MANET should be capable of setting up a new PKG. To become a PKG, a network node must satisfy the following conditions [13]:

- The lifetime of the node must be more than a constant value T, where T is an adequate period of time.
- The node must be well behaved throughout its lifetime; that is, it must not be found to be on a black list.
- The node must have enough capabilities, such as computational, communicational, and energy, to act as a PKG.

If a node meets the above-mentioned constraints, then it is believed to be a trusted and capable node. The procedure of generating a new PKG is as follows:

1. The node C_y requests k PKGs to become a new PKG. The requesting node signs and encrypts the request with its own private key $\mathcal{K}_{C_y,p_m}^{-1}$ and the public key of PKG $\mathcal{K}_{C_i,p_m}$, respectively. Then it sends it to the corresponding PKG.
2. When the ith PKG receives the request, it investigates the conditions of the requesting node. If C_y meets the required constraints to be a PKG, then the ith PKG calculates ss_{iy} and $\overline{\text{ss}}_{iy}$, using (5) below, and sends them to C_y.

$$\begin{cases} \text{ss}_{iy} = S_i l_i(ID_y) \bmod q, \\ \overline{\text{ss}}_{iy} = \overline{S}_i l_i(ID_y) \bmod q. \end{cases} \tag{5}$$

It is recommended to take an extra round of communications and shuffle the shares before they are sent to the requesting node so as to protect the secrecy of the coalition nodes' secret shares [13]. After shuffling ss_{iy} and $\overline{\text{ss}}_{iy}$, the ss'_{iy} and $\overline{\text{ss}}'_{iy}$ shares are obtained. Now, they are signed and encrypted, using the private keys of PKGs and the public key of the requesting node, respectively, and then sent to the requesting node.

Once the requesting node receives k valid shares from PKGs, it can then calculate its share of the master private key using (6) below.

$$\begin{aligned} S_y &= \sum_{i=1}^{k} \text{ss}'_{iy} \bmod q, \\ \overline{S}_y &= \sum_{i=1}^{k} \overline{\text{ss}}'_{iy} \bmod q. \end{aligned} \tag{6}$$

IV. Simulation Results and Discussion

In this section, our proposed methods are investigated and simulated in real network conditions.

1. Key Pool

The key pool technique is more robust than both those techniques that utilize a single key for all communications and those that use pair-wise private keys, because under node capture attacks, in the former case, all of the links between nodes are compromised, while in the latter case, all of the $n-1$ links that are connected to captured nodes are compromised. However, in the key pool technique only $k << n$ keys that belong to a key ring are compromised. In this case, an adversary can set up a successful attack with k/PO probability, in which PO is the number of keys in the key pool [23]. Due to wireless network constraints, the deployment of a fully connected network is infeasible. In fact, it is unnecessary to set up a fully connected network in the key discovery phase. It is enough to build a network with multi-hop links between every node; that is, a connected network is sufficient for the operation of our proposed method. Assume p_{ck} is the probability that there is a shared key between two network nodes, n is the number of network nodes, and $d_n = p_{\text{ck}} \times (n-1)$ is the expected degree of a node (that is, the average number of edges that connect a node to its neighbors). To build a connected network, the following questions should be answered [23]:

- What is the expected value of d_n required to set up a connected network of n nodes?
- According to d_n and wireless network constraints, what is the appropriate key ring size k and key pool size PO for a network of n nodes?

The first question can be answered using graph theory. The random graph $G(n, p_{\text{ck}})$ is a graph comprising n nodes, where there is a link (shared key) between each pair of nodes that has an associated probability p_{ck}. When p_{ck} equals zero, the graph has no edges, whereas, when p_{ck} equals one, the graph is fully

connected. The question is, what value of p_{ck} gives an almost-fully-connected graph G? Erdos and others [24] showed that for monotone properties of a graph, there is a value for p_{ck} for which any monotone property moves from nonexistent to certainly true. The function that defines p_{ck} is called the threshold function of the property. For an arbitrary probability of graph connectivity P_c, the threshold function of p_{ck} is defined by

$$P_c = \lim_{n\to\infty} \Pr[G(n, p_{ck}) \text{ is connected}] = e^{-e^{-c}}, \quad (7)$$

where

$$p_{ck} = \frac{\ln(n)}{n} + \frac{c}{n} \quad (8)$$

and c is a real constant. Therefore, for an arbitrary n, the values of p_{ck} and $d_n = p_{ck} \times (n-1)$ can be determined such that the resultant graph is connected with probability P_c.

$$d_n = \left\lceil \frac{n-1}{n} \right\rceil \left[\ln(n) - \ln\left(-\ln(P_c)\right) \right]. \quad (9)$$

According to (9), it is clear that $d = O(\log n)$. Figure 2 depicts the expected degree of a node, d_n, as a function of network size n, for different values of P_c. Figure 2 shows that the required number of neighbors for a graph, which is related to a high P_c value, is a finite value, and this can be achieved by using a finite number of keys.

For the second question, we note that the constraints of MANETs may cause the number of neighbors to be reduced to $n' \ll n$. For a given d_n, the required probability of an existing link between each pair of network nodes is $p'_{ck} = \frac{d_n}{(n'-1)} \gg p_{ck}$. So, the probability that two network nodes, with a key ring size of k, share at least one common key equals p'_{ck}, and the key pool size PO can be expressed as a function of k. To relate p'_{ck} to PO and k, we use (10) from [23].

$$p'_{ck} = 1 - \Pr[\text{two nodes do not share any common key}]. \quad (10)$$

To calculate the probability that two nodes do not share a common key, note that every key in a key ring is extracted from a key pool of size PO without replacement. Therefore, the number of possible key rings is $\frac{P!}{k!(P-k)!}$. After the formation of the first key ring, the number of possible key rings that do not share a common key with the first key ring is $\frac{(P-k)!}{k!(P-k)!}$; that is, the number of key rings that are formed from the remaining $(P-k)$ keys. Then, the probability that two nodes at least share one common key is

$$p'_{ck} = 1 - \frac{\left((P-k)!\right)^2}{(P-2k)!P!}. \quad (11)$$

The PO is a very large value, so the Stirling approximation can be used for $n!$ as follows:

$$n! \approx \sqrt{2\pi} n^{n+\frac{1}{2}} e^{-n}. \quad (12)$$

So, (11) can be approximated as follows:

$$p'_{ck} = 1 - \frac{\left(1-\frac{k}{P}\right)^{2\left(P-k+\frac{1}{2}\right)}}{\left(1-\frac{2k}{P}\right)^{\left(P-2k+\frac{1}{2}\right)}}. \quad (13)$$

Figure 3 illustrates the values that (13) gives for different values of PO. This figure shows that for different values of key pool size, by storing a few keys in each node, there is a high probability that there is a shared key between each pair of network nodes.

Next, the resilience of the key pool technique is examined against node capture attacks. For this purpose, we need to answer the following question: given two uncompromised nodes, one from network A and one from network B, what is the probability that an intruder can decrypt any form of

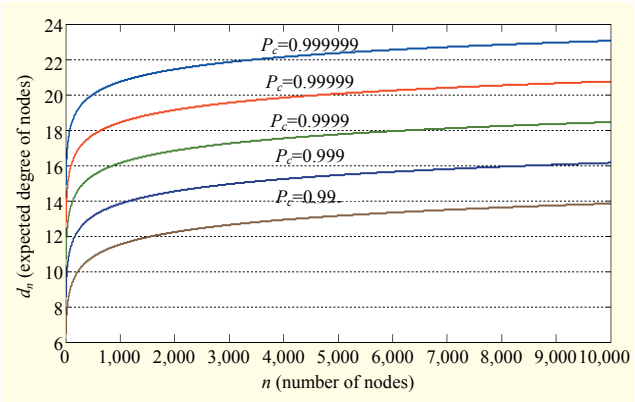

Fig. 2. Expected degree of nodes vs. number of network nodes for different graph connection probabilities.

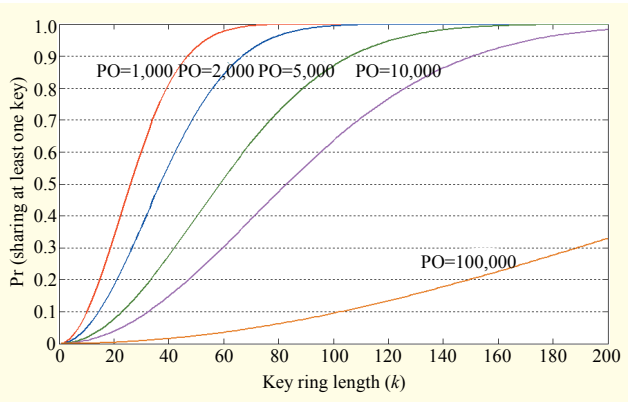

Fig. 3. Probability of existence of at least one shared key between every pair vs. key ring size.

communication between the two nodes by means of only the information that is acquired through the captured nodes?

Let xx be the number of captured nodes. Each node contains k keys; thus, the probability that a specific key is not compromised is $\left(1-\frac{k}{P}\right)^{xx}$. Therefore, the probability that a link is compromised is $1-\left(1-\frac{k}{P}\right)^{xx}$. In fact, this equation shows the additional information that an intruder obtains by capturing xx nodes. Figure 4 shows this parameter as a function of xx for different key ring size to key pool size ratios. Note that in this figure, the horizontal axis shows the total number of captured nodes, while the vertical axis shows the ratio of compromised links in the network.

The resilience of the key pool technique can be investigated from the point of view of an intruder; that is, how many network nodes should be captured, on average, so that the intruder can eavesdrop all network links with probability pp? To answer this question, $1-\left(1-\frac{k}{P}\right)^{xx}$ is used. It can be written as $xx=\ln(1-pp)/\ln\left(1-\frac{k}{P}\right)$ by slight modifications. Figure 5 depicts the average number of captured nodes required for an intruder to eavesdrop on any link as a function of pp, for different key ring size to key pool size ratios.

After discussing both the applied prospects and the security prospects of the key pool technique, the efficiency of it should be investigated. This technique is simulated in a real network scenario to examine its efficiency. Networks with 10, 20, 30, 40, and 50 nodes are selected. For each scenario, the parameters of the key pool technique are chosen in such a way that a given network is always almost certainly connected. Furthermore, each scenario is simulated for a fully connected network to show an upper bound of this technique. Table 1 illustrates the simulation parameters. The average time that is required for transmission of key identifiers and node ID is measured. The OPNET Modeler 14.5 is used for the simulations. The key load is simulated in Application layer and as an Application Demand. Network nodes are simulated in a wireless LAN workstation. These nodes utilize an antenna with 0.8 mW transmission power and –95 dB gain. The MAC and physical-layer protocols are 802.11b with 2 Mbps bitrate and direct sequence mode, respectively. The simulated nodes are mobile with 5 m/s velocity and 10 s pause time. They move on random patterns. Figure 6 illustrates the required time for the key discovery phase. Note that in this simulation only key identifiers, node IDs, and routing information are sent. This figure shows that the required time is within a reasonable bound, and due to the fact that these operations would take

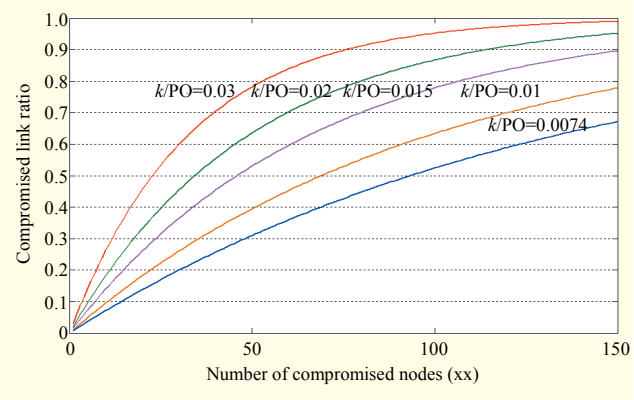

Fig. 4. Compromised link ratio vs. number of captured nodes.

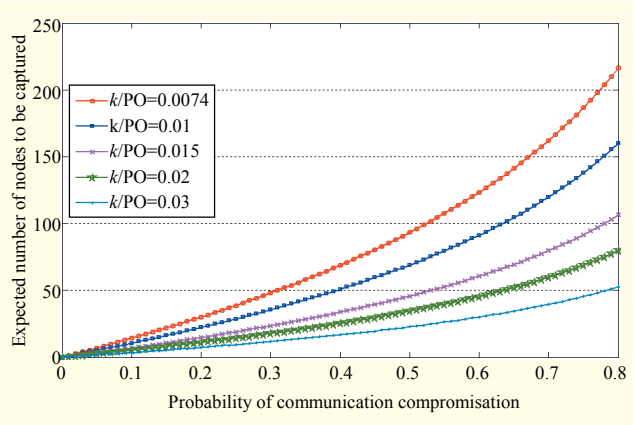

Fig. 5. Number of captured nodes required for an intruder to eavesdrop on any link with a given probability.

Table 1. Parameters of simulated networks.

Scenario	Number of network nodes (n)	Probability of network connectivity (P_c)	Key pool size (PO)	Key ring size (k)	Required number of neighbors (n')	Required key ring size (k')
1	10	0.99	1,000	34	9	95
2	20	0.999999	2,000	60	19	136
3	30	0.999999	5,000	65	29	218
4	40	0.999999	10,000	76	39	310
5	50	0.999999	20,000	94	49	377

place just once, it will not drastically affect the network performance. Using k keys in each node increases not only system reliability but also system cost. So, there exists a tradeoff between system reliability and communicational cost. However, an almost-certainly-connected network is enough in the case of our proposed method, but a network with high communicational capabilities may afford communicational cost. To investigate the effects of communicational capabilities

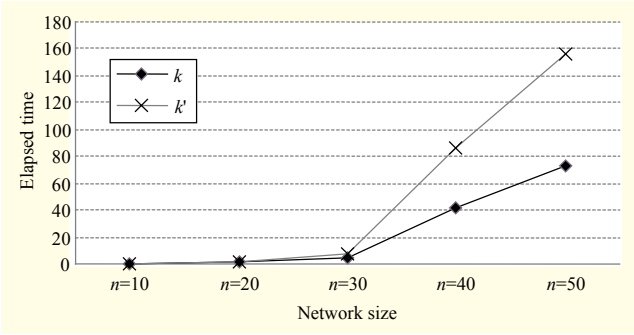

Fig. 6. Required time to discover shared key vs. different network sizes.

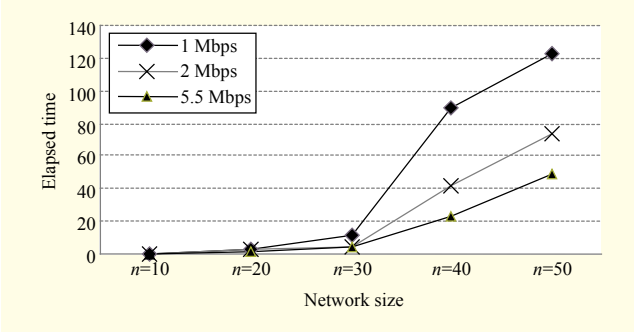

Fig. 7. Required time to discover shared key for different bitrates with key ring size k.

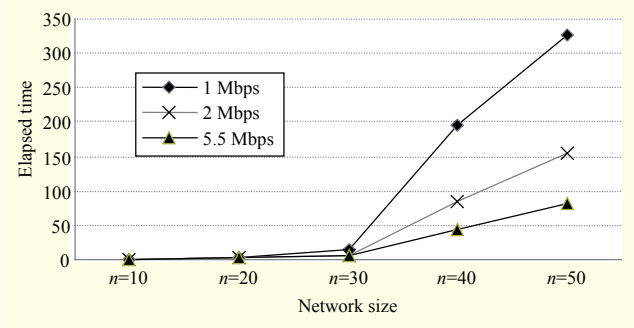

Fig. 8. Required time to discover shared key for different bitrates with key ring size k'.

on the required time for the key discovery phase, simulations are repeated for different bitrates. Figures 7 and 8 illustrate the required time for different network sizes.

2. Master Key Generation

The required time for master key generation in a distributed fashion has a significant impact on overall system performance, because a master key is significant to other cryptographic operations. The initial network nodes generate a master private key by sending master key subshares to shareholders. Figure 9

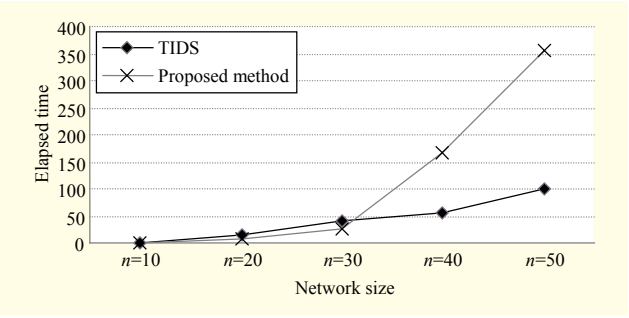

Fig. 9. Required time to generate master key in distributed fashion.

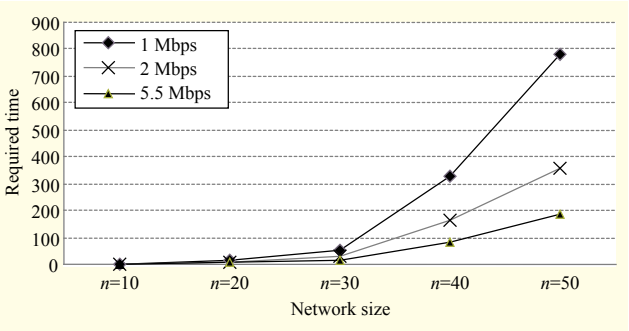

Fig. 10. Required time to generate master key for different bitrates.

illustrates the required time for distributed master key generation as a function of network size. In this figure, the plot of required time for the TIDS method [11] is depicted for the sake of comparison; the "required time" results of the simulations of the TIDS method are given for a 2 Mbps bitrate. Note that the additional cost of the proposed method is due to its additional security, since it generates two master keys; one for node-specific keys and another for phase-specific keys. This method greatly simplifies the key update and key revocation procedure.

3. Pairing-Based Key Agreement

The second proposed technique to set up secure links between network nodes for transmission of master key and private key shares is pairing-based key agreement. In this method, it is necessary that each node broadcasts its self-generated temporary public key. Again, the prominent factor is required time. To measure the required time, different network sizes are simulated. Figure 10 illustrates the required time for pairing-based key agreement as a function of network size for different bitrates.

To compare the key pool and pairing-based techniques, the required time for setting up a secure link is illustrated in Fig. 11. The figure shows that the pairing-based method and key pool with key ring size of k require almost equal amounts of time. However, the pairing-based method does not require key pre-

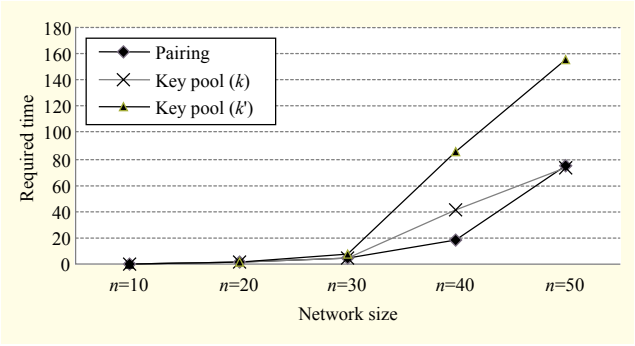

Fig. 11. Comparing required time to discover shared key using key pool method with time needed to agree on a key using pairing-based method.

distribution and storage and nodes can create a shared key on demand. Albeit, the pairing-based method seems to be an easy choice, but it is computationally more aggressive due to the bilinear mapping that it uses. However, the key pool technique needs fewer computations. Of course, pairing algorithms improve gradually; for example, recently an FPGA-based method was proposed [17], but it is not applicable to less capable nodes, and the key pool technique is a good alternative.

4. Security of Proposed Methods

The proposed security method utilizes a symmetric key method; for example, AES, to encrypt confidential data of IBC methods. The security requirements of a MANET depend on its cryptography system. In this section, the security features of the proposed method are investigated. The following assumptions are made for this purpose:

- The deployed symmetric key (for example, AES) is robust enough against intruders that do not have the encryption key.
- The BDHP over $(G_1, G_2, \hat{e})$ is hard enough; that is, by knowing $P, aP, bP, cP \in G_1$, in which $a, b, c \in Z_q^*$, there is no efficient algorithm to calculate $\hat{e}(P,P)^{abc} \in G_2$.

The DLP is hard in G_1; thus, it is infeasible to extract master private keys using a finite number of public/private key pairs. In other words, an intruder cannot determine the private key of an uncompromised node using the public/private keys of compromised nodes. Therefore, the proposed method is resilient against node capture.

A. Confidentiality

The confidentiality of key shares is guaranteed through two different techniques — key pool method and pairing-based method.

Key pool method. In this method, initial network nodes exchange key identifiers through public channels. In this stage, nodes may use puzzles (for an example, see [23]) to ensure that only privileged nodes (nodes that already have possession of a key) can understand a key pattern. For example, each node broadcasts α and $E_{K_i}(\alpha)$, where α is a random challenge and K_i for $i = 1, 2, \ldots, k$ represents all of the keys in the key ring of the node. Decryption of $E_{K_i}(\alpha)$ using the right key unveils α and establishes a shared key with the transmitter [23]. After the shared-key discovery phase, nodes share pair-wise common secrets; thus, they can use these secrets to encrypt the outgoing messages. According to assumption 1, the messages that are encrypted by these keys (for example, master key sub-shares or private key shares) are confidential. When the IBC system is deployed using secure links, all of the following transmissions between nodes that have got their private key are encrypted by the IBC system.

Assumption 2 guarantees the confidentiality of transmitted data, since it guarantees the security of the deployed IBC system.

Pairing-based method. In this method, initial network nodes broadcast their own temporary public keys without any encryption. Two nodes that know the public key of each other can establish a secure channel and agree on a common key without any further communication. According to assumption 2, an intruder cannot decrypt the encrypted data even if it knows the public keys of both nodes. Furthermore, assumption 1 guarantees the confidentiality of messages that are encrypted using an agreed key. These secure links are used for the transmission of master key sub-shares and private key shares. After deployment of the IBC system, confidentiality of communication is guaranteed, as explained in the key pool method.

B. Integrity

When the IBC system is deployed, transmitted messages are encrypted and signed using the public key of the recipient and the private key of the sender, respectively. Any malicious modifications to a transmitted message can be detected by a recipient using the digital signature of the transmitter. Therefore, the integrity of information is guaranteed. Though, before the deployment of the IBC system, there is no signature system and therefore integrity of information is not guaranteed.

C. Authentication

The messages that are sent using the IBC system are authenticated because they are signed by a digital signature that only the owner of a specific private key (sender) can generate. An intruder cannot extract the private key of a node using its public key (assumption 2); thus, it cannot generate a valid signature for a forged message. Therefore, the authentication of messages is guaranteed. Of course, the proposed methods are

not authenticated before the deployment of the IBC system, since there is no signature scheme.

D. Data Freshness

The *freshness* of signed messages can be guaranteed by using a time stamp; hence, out-of-order or replayed messages can be detected. However, the freshness of unsigned messages is not guaranteed; actually, this fact can be neglected, since these messages are used at the network formation stage.

E. Non-repudiation

The owner of a specific private key only can generate a valid signature. So, existence of a valid digital sign on any message declares the source of the message. Therefore, a node cannot deny its sign; hence, the proposed system is non-repudiation. However, non-repudiation only applies to messages that are sent after deployment of the IBC system; before this, there is no signature mechanism to create a non-repudiation system.

5. Future Work

Although the proposed methods offer data confidentiality before the deployment of an IBC system, they lack authentication, integrity, and so on. Therefore, they are still vulnerable to some form of attacks. It is recommended to investigate possible techniques to solve this problem. Certificate chaining methods [6] seem to be a good option for this purpose. Investigating the performance of the proposed methods in real nodes is also of great interest. As the next step, implementing a secure routing method based on the proposed methods is suggested.

Table 2. Comparison of existing and proposed methods.

Pulse	KM-SR	Integrity	Availability	Scalability
Proposed method in [17]	Yes	—	Good	Average
Proposed method in [10]	Yes	No	Good	Good
Proposed method in [11]	Yes	No	Good	Good
Proposed method in [12]	Yes	No	Good	Good
Proposed method in [13]	Yes	Yes	Good	Good
Proposed method in [21]	No	No	Bad	Bad
Proposed key pool–based method	No	No	Good	Good
Proposed pairing-based method	No	No	Good	Good

V. Conclusion

Security of MANETs is still a great challenge for researchers. Recently, security proposals that utilize IBC offered promising insights in how to tackle certain security problems. Despite these efforts, these proposals are not designed for truly ad hoc environments. Most commonly, they usually suffer from a security-routing interdependency cycle. In this paper, two novel methods were proposed to eliminate the interdependency cycle between secure routing and security services. One of these methods utilizes a key pool to construct secure routes for the distribution of cryptographic materials, while the other is based on pairing-based key agreement. Furthermore, the proposed methods utilize threshold cryptography for shared secret and private-key generation to eliminate the single point of failure and distribute the cryptographic services among network nodes. These characteristics guarantee high levels of availability and scalability for the proposed methods. To illustrate the effectiveness and capabilities of the proposed methods, they were simulated and compared with existing methods. Our simulation results showed that the proposed methods are well-suited for ad hoc environments. Also, these methods offer an acceptable degree of security for a MANET environment.

References

[1] W. Li and A. Joshi, *Security Issues in Mobile Ad Hoc Networks - A Survey*, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, 2008, pp. 1–23.

[2] D. Djenouri, L. Khelladi, and N. Badache, "A Survey of Security Issues in Mobile Ad Hoc and Sensor Networks," *IEEE Commun. Surveys Tutorials*, vol. 7, no. 4, Feb. 2006, pp. 2–28.

[3] S. Zhao et al., "A Survey of Applications of Identity-Based Cryptography in Mobile Ad-Hoc Networks," *IEEE Commun. Surveys Tutorials*, vol. 14, no. 2, May 2012, pp. 380–400.

[4] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes," in *Advances in Cryptology*, Berlin, Germany: Springer Berlin Heidelberg, 1985, pp. 47–53.

[5] X. Fan, "Efficient Cryptographic Algorithms and Protocols for Mobile Ad Hoc Networks," Ph.D. dissertation, University of Waterloo, Ontario, Canada, 2010.

[6] J.V.D. Merwe, D. Dawoud, and S. McDonald, "A Survey on Peer-to-Peer Key Management for Mobile Ad Hoc Networks," *ACM Comput. Surveys*, vol. 39, no. 1, Apr. 2007, pp. 1–23.

[7] Y. Zhang et al., "Securing Mobile Ad Hoc Networks with

Certificateless Public Keys," *IEEE Trans. Dependable Secure Comput.*, vol. 3, no. 4, Nov. 2006, pp. 386–399.

[8] A. Khalili, J. Katz, and W.A. Arbaugh, "Toward Secure Key Distribution in Truly Ad-Hoc Networks," *Proc. Symp. Appl. Internet Workshops*, Orlando, FL, USA, Jan. 27–31, 2003, pp. 342–346.

[9] R.B. Bobba et al., "Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks," *IEEE Global Telecommun. Conf.*, San Francisco, CA, USA, vol. 3, Dec. 1–5, 2003, pp. 1511–1515.

[10] H. Deng, A. Mukherjee, and D.P. Agrawal, "Threshold and Identity-Based Key Management and Authentication for Wireless Ad Hoc Networks," *Proc. IEEE Int. Conf. Inf. Technol.: Coding Comput.*, Las Vegas, NV, USA, vol. 1, Apr. 5–7, 2004. pp. 107–111.

[11] H. Deng and D.P. Agrawal, "TIDS: Threshold and Identity-Based Security Scheme for Wireless Ad Hoc Networks," *Ad Hoc Netw.*, vol. 2, no. 3, July 2004, pp. 291–307.

[12] Y. Zhang et al., "Identity-Based Threshold Key Management for Ad Hoc Networks," *Pacific-Asia Workshop Comput. Intell. Ind. Appl.*, Wuhan, China, vol. 2, Dec. 19–20, 2008, pp. 797–801.

[13] P. Xia et al., "Identity-Based Fully Distributed Certificate Authority in an OLSR MANET," *Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Dalian, China, Oct. 12–14, 2008, pp. 1–4.

[14] G. Li and W. Han, "A New Scheme for Key Management in Ad Hoc Networks," *Netw.-Int. Conf. Netw.*, Reunion Island, France, Apr. 17–21, 2005, pp. 242–249.

[15] Y. Zhang et al., "AC-PKI: Anonymous and Certificateless Public-Key Infrastructure for Mobile Ad Hoc Networks," *IEEE Int. Conf. Commun.*, Seoul, Rep. of Korea, vol. 5, May 16–20, 2005, pp. 3515–3519.

[16] Y. Ren et al., "Identity-Based Key Issuing Protocol for Ad Hoc Networks," *IEEE Int. Conf. Comput. Intell. Security*, Harbin, China, Dec. 15–19, 2007, pp. 917–921.

[17] Y. Zhang et al., "Securing Mobile Ad Hoc Networks with Certificateless Public Keys," *IEEE Trans. Dependable Secure Comput.*, vol. 3, no. 4, Nov. 2006, pp. 386–399.

[18] B. Lee et al., "Secure Key Issuing in ID-Based Cryptography," in *Proc. Workshop Australasian Inf. Security, Data Mining Web Intell., Softw. Int.*, vol. 32, Australia: Australian Computer Society Inc., 2004, pp. 69–74.

[19] N. Saxena, "Public Key Cryptography Sans Certificates in Ad Hoc Networks," in *Appl. Cryptography Netw. Security*, Berlin, Germany: Springer Berlin Heidelberg, 2006, pp. 375–389.

[20] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," in *Adv. Cryptology*, Berlin, Germany: Springer Berlin Heidelberg, 2001, pp. 213–229.

[21] S. Zhao, R. Kent, and A. Aggarwal, "A Key Management and Secure Routing Integrated Framework for Mobile Ad-Hoc Networks," *Ad Hoc Netw.*, vol. 11, no. 3, May 2013, pp. 1046–1061.

[22] A. Shamir, "How to Share a Secret," *Commun. ACM*, vol. 22, no. 11, Nov. 1979, pp. 612–613.

[23] L. Eschenauer and V.D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks," in *Proc. ACM Conf. Comput. Commun. Security*, USA: ACM, 2002, pp. 41–47.

[24] J. Spencer, "*The Strange Logic of Random Graphs*," Berlin, Germany: Springer-Verlag Berlin Heidelberg, 2001.

Kamal Adli Mehr is currently a PhD candidate at the University of Tabriz, East Azerbaijan, Iran. He received his BS degree in electrical engineering (communication systems) and his MS degree in communication systems from the University of Tabriz, in 2011 and 2013, respectively. His current research interests include network security, secure communication, cryptography, and wireless communication systems.

Javad Musevi Niya received his BS degree in electrical engineering from the University of Tehran, Iran and his MS and PhD degrees in communication systems from Sharif University of Technology, Tehran and the University of Tabriz, East Azerbaijan, Iran, in 1988, 1991, and 2006, respectively. Since September 2006, he has been with WiLab, the Faculty of Electrical and Computer Engineering, University of Tabriz. His current research interests include wireless communication systems, multimedia networks, and signal processing for communication systems.