Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 374713, 9 pages
http://dx.doi.org/10.1155/2013/374713

Research Article

Anonymous Cluster-Based MANETs with Threshold Signature

YoHan Park,[1] YoungHo Park,[2] and SangJae Moon[3]

[1] *School of Electrical Engineering and Computer Science, Kyungpook National University, Daegu, Republic of Korea*
[2] *School of Electrical Engineering, Kyungpook National University, Sangju, Republic of Korea*
[3] *School of Electronics Engineering, College of IT Engineering, Kyungpook National University, Daegu, Republic of Korea*

Correspondence should be addressed to YoungHo Park; parkyh@knu.ac.kr

Received 2 September 2012; Revised 20 February 2013; Accepted 21 February 2013

Academic Editor: Dan Kim

Security supports are a significant factor in the design of security system in ad hoc networks. It is particularly important to protect the identities of individual nodes to avoid personal privacy concerns. In this paper, we propose a security system for ID-based anonymous cluster-based MANETs to protect the privacy of nodes. Moreover, we propose a threshold signature scheme without pairing computations, which diminishes the computation load on each node. To the best of our knowledge, our proposed security system is the first in which the pseudonym is combined with cluster-based mobile ad hoc networks (MANETs) without a trusted entity. According to our protocol analysis, our proposal satisfies most properties for an anonymous security system and effectively copes with dynamic environments with greater efficiency by using secret sharing schemes. Therefore, it could be usefully applied to preserve privacy in dynamic MANETs without a trusted entity, such as military battlefields, emergency areas, mobile marketplaces, and vehicular ad hoc networks (VANETs).

1. Introduction

MANETs support communications in situations involving temporary self-organization and infrastructure-less situations, such as battlefields, disaster relief situations, and emergency rescue areas [1]. Recently, MANETs have been extended to intelligent transport systems, often called VANETs. However, MANETs are subject to various types of attacks because of the wireless and infrastructure-less environments in which they are used. Moreover, these network structures make it difficult to apply the certificate-based public key cryptosystem (CBC) to meet the requirements of the certificate authority (CA). As a powerful alternative to the CBC, the identity-based cryptosystem (IBC) proposed by Shamir [2] has been gaining momentum in recent years. It allows public keys to be derived from entities' known identity information, such as e-mail addresses, IP addresses, or codes, thus eliminating the need for public key distribution and certificates. In other words, a user's public key can be determined directly from his identifying information, rather than having to be extracted from a certificate issued by a CA.

However, a centralized public key generator (PKG) which generates a key pair for users in the IBC would obviously be easy to attack, and accessibility could not be guaranteed at all times for all participants in MANETs. To counter this, Boneh and Franklin [3] suggested spreading the PKG by means of distributed PKGs (D-PKGs), using threshold cryptography. Distribution of a signing key and CA functionality over multiple nodes using secret sharing and threshold cryptography is a possible solution to this problem. Most studies in cluster-based MANETs considering D-PKGs have been based on hierarchy topology structures, which classify the node into two types, namely, representative nodes, called clusterheads (CHs), and common nodes.

Furthermore, the threshold signature scheme in MANETs has attracted many researchers' attentions recently [4]. The threshold signature achieves the purpose for many individuals to cooperatively sign the same document. Some schemes with trusted party are proposed and applied to CBC-based networks [5], while these are unreasonable to MANETs where there are no infrastructure and control administration. Therefore, the threshold signature scheme

is necessary for secure message transmission in MANETs. A recent research on cluster-based MANETs has sought to address the privacy problem [6] and threshold signature schemes [7], but research on anonymous threshold signature schemes in cluster-based MANETs has been insufficient.

This paper proposes a security system for ID-based anonymous cluster-based MANETs to protect the privacy of nodes. Moreover, we propose a threshold signature scheme without pairing computation, which diminishes the computation load on each node in comparison with existing schemes. The major contributions of this study are summarized as follows.

(i) Security of a cluster key distribution scheme and key agreement scheme. We propose a secure cluster key distribution scheme and a key agreement scheme with anonymity for cluster-based MAENTs. Cluster key distribution scheme ensures that the compromise of an arbitrary number of nodes outside the target cluster does not jeopardize the secrecy of noncompromised nodes. Key agreement scheme also ensures secure communication between nodes in intra- and interclusters.

(ii) Consideration of threshold signature without pairing computation. Our threshold signature supports threshold signature. Comparing to existing threshold signatures, we diminish the computation load on signing nodes and a verification node; instead, CHs aid signature verification process. Thus, our threshold signature is suitable for the distributed PKGs architecture or the cluster-based network architecture.

(iii) Protection of personal privacy. Our schemes support entity anonymity. Only the entities especially of the matched session can know the identity of others with whom they are in communication. For instance, CHs could be dealers, and common nodes could be purchasers when considering temporary established mobile markets. It is no needed to hide the CHs' identity because every purchaser should recognize CHs as dealers and their information; therefore, identities of CHs are not quite important. Besides, the information of purchasers is much attractive to adversaries because the information could be abusable commercially and criminally.

The rest of the paper is organized as follows. In Section 2, we present preliminaries, and the system model is presented in Section 3. In Section 4, we describe ID-based anonymous cluster-based MANETs, and the threshold signature is discussed in Section 5. Finally, we analyze the proposed scheme in Section 6 and conclude our findings in Section 7.

2. Preliminaries

In this section and we present notations, then define the cryptographic system and primitives used as building blocks in our security system.

Table 1: Notations.

$\mathbb{G}_1, \mathbb{G}_2$	Cyclic groups of order q
$\hat{e}$	Pairing s.t. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$
P	Generator of $\mathbb{G}_1$
$\mathrm{CH}_j/\mathrm{ID}_A$	Network ID of clusterhead j/common node A
PS_A	Pseudonym of ID_A
S_j/Q_j	Private/public key pair of clusterhead CH_j
S_A/Q_A	Private/public key pair of common node ID_A
$S_{\mathrm{PS}_A}/Q_{\mathrm{PS}_A}$	Private/public key pair of pseudonym PS_A
U	Maximum update phase index
K_m	Cluster key at m-th update phase
$g_m(x)$	Polynomial for cluster key K_m at m-th update phase
$f_m^{\mathrm{CH}_j}(x)$	Polynomial of CH_j at m-th update phase
$v_M^{\mathrm{CH}_j}(x)$	Verifying polynomial of CH_j for message M
(t, n)	Secret sharing parameters for $g_m(x)$
(t_1, n_1)	Secret sharing parameters for $f_m^{\mathrm{CH}_j}(x)$
H_0	Mapping $\{0,1\}^* \rightarrow \mathbb{Z}_p^*$
H_1	Mapping $\{0,1\}^* \rightarrow \mathbb{G}_1$
H_2	Mapping $\mathbb{Z}_p^* \rightarrow \{0,1\}^*$
H_3	Mapping $\mathbb{G}_1 \rightarrow \mathbb{Z}_p^*$
$\mathrm{HMAC}_K(M)$	Hash message authentication code of message M with key K

2.1. Notations. Table 1 lists some important notations whose concrete meanings will be further explained where they appear for the first time.

2.2. ID-Based Cryptography. Let p, q be two large primes, and let $E/\mathbb{F}_p$ indicate an elliptic curve $y^2 = x^3 + ax + b$ over the finite field $\mathbb{F}_p$. We denote by $\mathbb{G}_1$ a q-order subgroup of the additive group of points of $E/\mathbb{F}_p$ and by $\mathbb{G}_2$ a q-order subgroup of the multiplicative group of the finite field $\mathbb{F}_{p^2}^*$. The discrete logarithm problem (DLP) is required to be hard in both $\mathbb{G}_1$ and $\mathbb{G}_2$. For us, a pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties.

(i) Bilinear: for all $P, Q, R, S \in \mathbb{G}_1$,

$\hat{e}(P + Q, R + S) = \hat{e}(P, R)\hat{e}(P, S)\hat{e}(Q, R)\hat{e}(Q, S)$.

Consequently, for all $a, b \in \mathbb{Z}_q^*$, we have

$\hat{e}(aP, bQ) = \hat{e}(aP, Q)^b = \hat{e}(P, bQ)^a = \hat{e}(P, Q)^{ab}$ and so forth.

(ii) Nondegenerate: if P is a generator of $\mathbb{G}_1$, then $\hat{e}(P, P) \in \mathbb{F}_{q^2}^*$ is a generator of $\mathbb{G}_2$.

(iii) Computable: there is an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

Note that $\hat{e}$ is also *symmetric*, that is, $\hat{e}(P, Q) = \hat{e}(Q, P)$, for all $P, Q \in \mathbb{G}_1$, which follows immediately from the bilinearity and the fact that $\mathbb{G}_1$ is a cyclic group. Modified Weil [3] and Tate [8] pairings are examples of such bilinear maps for which the *bilinear Diffie-Hellman problem (BDHP)* is believed to be hard.

Definition 1 (bilinear Diffie-Hellman (BDH) problem). The BDH problem for a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is defined as follows: given $P, xP, yP, zP \in \mathbb{G}_1$, where x, y, z are random numbers from $\mathbb{Z}_p^*$, compute $\hat{e}(P,P)^{xyz} \in \mathbb{G}_2$.

Definition 2 (decisional Bilinear Diffie-Hellman (DBDH) problem). The DBDH problem for a bilinear pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is defined as follows: given $P, xP, yP, zP \in \mathbb{G}_1$, where x, y, z are random numbers from $\mathbb{Z}_p^*$ and $W \in \mathbb{G}_2$ that determine if $\hat{e}(P,P)^{xyz} = W$ (if it holds), then the tuple $\langle P, xP, yP, zP, W\rangle$ is called a BDH tuple.

2.3. Threshold Schemes Based on Secret Sharing

2.3.1. Shamir's (t,n)*-SS.* In Shamir's (t,n)-SS [9], based on a Lagrange interpolating polynomial, there are n shareholders $\mathcal{U} = \{\mathcal{U}_1, \ldots, \mathcal{U}_n\}$ and a mutually trusted dealer D. The scheme consists of two algorithms.

(1) Share generation algorithm: dealer D does the following:

   (i) dealer D first picks a polynomial $f(x)$ of degree $(t-1)$ randomly: $f(x) = a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}$ in which the secret $K = a_0 = f(0)$ and all coefficients $a_0, a_1, \ldots, a_{t-1}$ are in a finite field $\mathbb{F}_p = GF(p)$ with p elements,

   (ii) D computes all shares: $K_i = f(s_i) \pmod p$ for $i = 1, \ldots, n$,

   (iii) Then, D outputs a subset Ω of size n, $(K_1, K_2, \ldots, K_n)$, and distributes each share K_i to corresponding shareholder s_i privately.

(2) Secret reconstruction algorithm: based on a Lagrange interpolation, any subset $A \subset \Omega$ of size t can reconstruct the polynomial $f(x)$ as

$$f(x) = \sum_{i \in A} \lambda_i(x) K_i \pmod q, \tag{1}$$

where $A = \{1, \ldots, t\} \subseteq \{1, 2, \ldots, n\}$, $\lambda_i(x) = \prod_{j \in A \setminus i}((s_j - x)/(s_j - s_i))$ is called a Lagrange coefficient. The secret K can be reconstructed by computing $f(0)$.

We note that the above scheme satisfies the basic security requirements of secret sharing schemes as follows: (1) with knowledge of a t or more than t shares, it can reconstruct the secret K easily; (2) with knowledge of fewer than $(t-1)$ shares, it cannot reconstruct the secret K. Shamir's scheme is *information theoretically secure* since the scheme satisfies these two requirements without making any computational assumption. For more information on this scheme, readers can refer to the original paper [9].

3. System Model

We describe the network architecture and the security requirements.

3.1. Network Architecture.

We divide the networks into several clusters to enhance the efficiency and availability. The clustering is a method that enables nodes to be organized on the basis of their relative proximity to one another. We envision a cluster-based MANETs consisting of n CHs without any prior contact, trust, or authority relation. In each cluster, one distinguished node, the CH, is responsible for establishing and managing the cluster. The size of the network may change dynamically according to the efficiency and the security. Let us consider an ad hoc network with n CHs that are selected to enable secure and robust pseudonym generation. We assume that compromised CHs will eventually exhibit detectable misbehavior. Studies [10, 11] discussed ways to detect the misbehavior of nodes or intrusions in detail.

Our schemes work securely and properly on the assumption, which is similar to assumptions made in [12–14], that adversaries compromise no more than $(t-1)$ out of n CHs in a given time period. In practice, it is hard to compromise, in a given time period, t CHs, which are more secure and powerful than common nodes and are geographically distributed over a wide area. In terms of nodes' ability, we also assume that CHs have more computation and communication power than common nodes. More precisely, CHs have an additional powerful radio to establish wireless links among themselves and strong resistance against malicious attacks.

Figure 1(a) illustrates the basic network architecture of our security system for cluster setup and pseudonym generation. CHs have secret sharing $g_m(\mathrm{CH}_j)$ generated by a PKG before implementation. CHs can generate polynomials $g_m(x)$ when at least t CHs collaborate in the secret reconstruction algorithm. The CHs that reconstruct $g_m(x)$ have the same cluster key, K_m. This cluster key is periodically updated according to the update phase. Using the same cluster key, CHs generate their own polynomial $f_m^{\mathrm{CH}_j}(x)$, called the respective polynomial later. Finally, common nodes in each cluster register and receive a pair of pseudonym from their CHs. We only consider the privacy of common nodes.

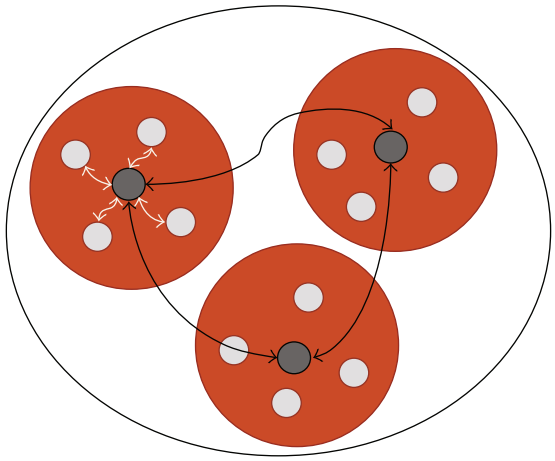
Figure 1(b) illustrates the threshold signature generation process in a cluster. Nodes that are member of the same cluster and try to send the same message generate a threshold signature and send the message with a threshold signature to a verifier and the CH. Then, the CH checks the validity of messages and signatures and generates and sends additional points to a verifier. Finally, a verifier checks the validity of signatures.

3.2. Security Requirements.

We define security requirements for our anonymous security system.

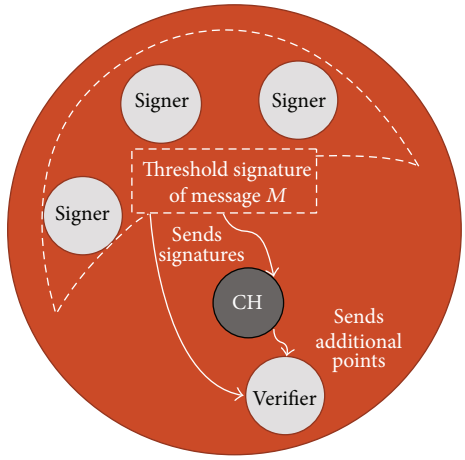
(1) Privacy: private information, such as the node's identity and location, should be protected against malicious adversaries. Formally, given two sets of legitimate identities, ID_0 and ID_1, the adversary should not have any significant advantage in guessing $d = 0$ or 1 for the pseudonym PS_d.

(2) Traceability: compromised nodes are identified, and the corresponding identity and pseudonym should be revoked to protect networks against further threats.



Cluster
Clusterhead (CH)
Common node

(a) Basic architecture



Cluster
Clusterhead (CH)
Common node

(b) Threshold signature generation process

FIGURE 1: Network Architectures.

(3) Nonmanipulation: no nodes or CHs can computationally manipulate a pseudonym from an identity.

(4) Verifiability: from the signature, the verifier can be convinced of the signer.

(5) Undeniability: once a signer creates a valid signature, he cannot repudiate the signature creation.

(6) Unforgeability: no nodes can forge a signature; it can only be replicated by the signer who creates it.

4. Anonymous Cluster-Based MANETs

4.1. System Setup. It is reasonable to assume that a trusted PKG could bootstrap the network, which itself is not a part of the resulting network. The basic operations consist of generating pairing parameters, private keys, and secret sharing.

(a) Generation of the pairing parameters $(p, q, \hat{e})$: to bootstrap the network, the PKG does the following.

(i) Generation of pairing parameters $(p, q, \hat{e})$. It selects an arbitrary generators $s \in \mathbb{Z}_q^*$ as its private key.

(b) Generation of secret sharing for cluster key: to generate secret sharing for cluster key, the PKG does the following.

(i) It submits identity information. A CH submits its identity information, CH_j, where $(1 \le j \le n)$, to PKG.

(ii) It chooses cluster key. The PKG selects an arbitrary cluster keys $K_m \in \mathbb{Z}_q^*$, where $(1 \le m \le U)$.

(iii) It determines polynomials of degree $(t-1)$. The PKG determines random polynomials, $g_m(x) = K_m + \sum_{i=1}^{t-1} a_i x^i \pmod q$.

(iv) It performs (t, n)-SS of K_m. The PKG computes $\mathrm{ch}_j = H_0(\mathrm{CH}_j)$ and secret sharing of $g_m(x)$: $g_m(\mathrm{ch}_j)$.

(v) Parameters U, $g_m(\mathrm{ch}_j)$ are preloaded to each CH_j securely.

(c) Generation of CH's ID-based private keys: to hand out CH's private key, the PKG does the following.

(i) It submits identity information. A CH submits its identity information, CH_j, to PKG.

(ii) It computes key pairs. The PKG computes a public/private key pairs: $Q_j = H_1(\mathrm{CH}_j)$ and $S_j = sH_1(\mathrm{CH}_j)$, $(1 \le j \le n)$.

(iii) Parameters $(p, q, G_1, G_2, \hat{e}, P, H_0, H_1, H_2, H_3)$ and key pairs (Q_j, S_j) are preloaded to each CH_j securely.

(d) Generation of common nodes' ID-based private keys: to hand out node's private key, the PKG does the following.

(i) It submits identity information. A common node submits its identity information, ID_A, to PKG.

(ii) It computes key pairs. The PKG computes a public/private key pairs: $Q_A = H_1(\mathrm{ID}_A)$ and $S_A = sH_1(\mathrm{ID}_A)$, $(1 \le j \le n_1)$.

(iii) Parameters $(p, q, G_1, G_2, \hat{e}, P, H_0, H_1, H_2, H_3)$ and key pairs (Q_A, S_A) are preloaded to each ID_A securely.

Due to the difficulty of solving the DLP in G_1, it is computationally infeasible to derive the network master secrets s from an arbitrary number of private keys. This means that no matter how many key pairs adversaries acquire from compromised nodes, they cannot deduce the private key of any noncompromised node. Colluding CHs (no more than $(t-1)$ out of n in a given time period) cannot compute a cluster key.

4.2. Cluster Setup. Cluster-based MANETs work without the help of PKG after completing the system setup. Instead of a PKG, CHs play this role using their secret sharing. In our security system, to provide security services to networks, each CH first generates respective polynomials, that cover within a cluster and a group secret key. Then, the CHs establish a secure channel with CHs or nodes to forward them. Secure channels are generated using their initial key pairs. Secure channels between CHs or between a CH and a node are established by the noninteractive key agreement scheme as follows:

$$\begin{aligned} D_{AB} &= \hat{e}\left(Q_B, S_A\right) \\ &= \hat{e}\left(H_1\left(\mathrm{ID}_B\right), H_1\left(\mathrm{ID}_A\right)\right)^s = \hat{e}\left(S_B, Q_A\right) = D_{BA}, \\ k_{AB} &= H_2\left(D_{AB}\right). \end{aligned} \quad (2)$$

Here, A, B can be a node or a CH. Using this channel, CHs first authenticate nodes and other CHs and then forward respective polynomials and a group secret key. Respective polynomials are for cluster reconfiguration, and the group secret key is for establishing secure channels with pseudonyms. Generation of respective polynomials is carried out as follows.

(1) Generate respective polynomials $f_m^{\mathrm{CH}_j}(x)$: to generate respective polynomials, CHs must reconstruct the polynomial $g_m(x)$ first. To reconstruct polynomials, $g_m(x)$, and generate respective polynomials, $f_m^{\mathrm{CH}_j}(x)$, CHs do the following.

(a) Pooling secret sharing. Every CH shares their secret sharing, $g_m(\mathrm{ch}_j)$.

(b) Performing secret reconstruction algorithm. Each CH reconstructs the polynomial $g_m(x)$ and computes cluster key K_m:

$$g_m(x) = \sum_{j \in A} \lambda_j(x)\, g_m\left(\mathrm{ch}_j\right) \pmod{q}, \quad (3)$$

where $A = \{1, \ldots, t\} \subseteq \{1, 2, \ldots, n\}$, $\lambda_j(x) = \prod_{k \in A \setminus j}((\mathrm{ch}_k - x)/(\mathrm{ch}_k - \mathrm{ch}_j))$ is called a Lagrange coefficient. The secret K_m can be reconstructed by computing $g_m(0)$.

(c) Generating respective polynomials of degree $(t_1 - 1)$. Each CH randomly generates a polynomial $f_m^{\mathrm{CH}_j}(x) = K_m + \sum_{i=1}^{t_1-1} b_i x^i \pmod{q}$. Each CH could change the polynomial by switching coefficients $b_0, b_1, \ldots, b_{t_1-1}$ arbitrarily.

To avoid cryptanalysis and malfunction of CHs, frequent key updates are needed. Our key update schemes consist of two parts: one is an update of the cluster key, and the other is the respective polynomial. The cluster keys, K_m, are refreshed periodically at a predefined time interval using secret sharing $g_m(\mathrm{ch}_j)$, where $(0 \le m \le U, 1 \le j \le n)$. The cluster key update can enhance the security level of the network. Intact CHs reject secret sharing of compromised CHs and, as a result, are isolated from the networks. Furthermore, pseudonyms also could be updated consistently regardless of the cluster key update. Each CH can generate different pseudonyms with the same identity by changing a polynomial $f_m^{\mathrm{CH}_j}(x)$ of its choice by replacing b_i with $c_i, d_i, \ldots$.

4.3. Generation of Pseudonyms. Pseudonym generation is an essential process to provide privacy of each node. Figure 1(a) shows a scenario of pseudonym generation process. Initial pseudonym generation starts when the registered nodes on the PKG try to get a pseudonym from an adjacent CH. The CH generates pseudonyms for common nodes within a cluster using its respective polynomial. Pseudonyms and secret sharing are generated as follows.

(1) Generation of pseudonyms and key pair: to generate pseudonyms, CHs do the following.

(a) Performing (t_1, n_1)-SS of K_m to generate pseudonyms. Each CH computes $\mathrm{id}_A = H_0(\mathrm{ID}_A)$ and secret sharing of $f_m^{\mathrm{CH}_j}(x)$: $\mathrm{PS}_A = f_m^{\mathrm{CH}_j}(\mathrm{id}_A)$, where $(1 \le A \le n_1)$.

(b) Computing key pairs. Each CH computes a public/private key pairs: $Q_{\mathrm{PS}_A} = \mathrm{PS}_A K_m P$ and $S_{\mathrm{PS}_A} = \mathrm{PS}_A$.

(c) Recording pseudonyms. Each CH records the identities and the pseudonyms with the corresponding key pairs at the pseudonym lookup table (PLT).

(d) Forwarding pseudonyms, key pairs, and group secret key to corresponding nodes. CHs forward pseudonyms to nodes, respectively, using a secure channel.

Other CHs cannot know pseudonyms from public keys of nodes even though they have knowledge of cluster key K_m because of the hardness of DLP in G_1. Using the pseudonym key pair, common nodes establish a secure channel by the noninteractive key agreement scheme as follows:

$$\begin{aligned} D_{AB} &= \hat{e}\left(Q_{\mathrm{PS}_B}, S_{\mathrm{PS}_A} P\right) = \hat{e}\left(Q_{\mathrm{PS}_A}, S_{\mathrm{PS}_B} P\right) = \hat{e}(P, P)^{\mathrm{PS}_A \mathrm{PS}_B K_m}, \\ k_{AB} &= H_2\left(D_{AB}\right). \end{aligned} \quad (4)$$

For noninteractive key agreement between a CH and a node, each CH randomly chooses $r \in \mathbb{Z}_q^*$ and computes a temporary public key as $f_m^{\mathrm{CH}_k}(r) K_m P$ then publishes it.

5. Threshold Signature in Anonymous Cluster-Based MANETs

We propose an anonymous threshold signature. The proposed scheme involves five roles: a clusterhead (CH), a set of members $\mathscr{M} = \{M_1, \ldots, M_{t_1}\}$ in a cluster (where M_{t_1} is the identity of the ith $(1 \le i \le t_1)$ member), a set of signer $\mathcal{S} = \{S_1, \ldots, S_{t_2}\}$ (where $\mathcal{S}$ is a subset of $\mathscr{M}$ and S_{t_2} is the identity of the jth $(1 \le j \le t_2)$ member), and a verifier V.

(1) Generate threshold signature: to generate a threshold signature regarding message M, a number of t_2 nodes among the members of $\mathscr{M}$ perform the following steps.

(a) Requesting threshold generation. The initiator, one of signers, sends a threshold generation request to the CH with a list of signers as $\{TS_1, \ldots, TS_{t_2}\}$.

(b) Sending tokens. The CH chooses an arbitrary tokens $T_A \in \mathbb{Z}_q^*$, where $(1 \le A \le t_2)$, and sends them to corresponding signers securely.

(c) Generating signature. Then, each signer generates a signature: $\mathrm{Sig}_{\mathrm{PS}_A} = H_0(M) \cdot S_{\mathrm{PS}_A}$ and computes with a corresponding token: $T \cdot \mathrm{Sig}_{\mathrm{PS}_A} = T_A \cdot \mathrm{Sig}_{\mathrm{PS}_A}$.

(d) Sending the signature and a pseudonym public key. Each node sends the message tuple to the verifier V and the CH_j:

$$\left(M, Q_{\mathrm{PS}_A}, Q_{\mathrm{PS}_V}, T \cdot \mathrm{Sig}_{\mathrm{PS}_A}, \mathrm{HMAC}_{\mathrm{PS}_A}\left(M, Q_{\mathrm{PS}_V}, T \cdot \mathrm{Sig}_{\mathrm{PS}_A}\right)\right), \tag{5}$$

(2) Generate a verifying polynomial: to check the validity of signatures, the CH_j performs the following steps:

(a) Check the validity of a set of messages. The CH_j searches corresponding pseudonyms with pseudonym public keys using the pseudonym lookup table (PLT) and then checks the validity of HMAC respectively.

(b) Checking the validity of signatures. The CH_j recovers signatures $\mathrm{Sig}_{\mathrm{PS}_A}$ from $T \cdot \mathrm{Sig}_{\mathrm{PS}_A}$ using corresponding tokens and generates additional $(t_1 - t_2)$ points on $H_0(M) \cdot f_m^{\mathrm{CH}_j}(x)$. Then, it performs a secret reconstruction algorithm using t_2 received signatures and $(t_1 - t_2)$ generated additional points. The reconstruction algorithm is as follows:

$$H_0(M) \cdot f_m^{\mathrm{CH}_j}(x) = \sum_{i \in N} \lambda_i(x) Y_i \pmod q,$$
$$Y_i = \begin{cases} \mathrm{Sig}_{\mathrm{PS}_i}, & (1 \le i \le t_2), \\ H_0(M) \cdot \mathrm{PS}_i, & (t_2 + 1 \le i \le t_1), \end{cases} \tag{6}$$

where $N = \{1, \ldots, t_1\}$, $\lambda_i(x) = \prod_{j \in N \setminus i}((\mathrm{id}_i - x)/(\mathrm{id}_j - \mathrm{id}_i))$ is called a Lagrange coefficient. If the reconstructed polynomial has $H_0(M) \cdot K_m$ at $(x = 0)$, the CH_j accepts signatures as valid.

(c) Generating a verifying polynomial. If all messages and signatures are valid, the CH_j generates an extra polynomial as follows:

(i) generating a verifying-polynomial. The CH_j generates $v_M^{\mathrm{CH}_j}(x)$ of degree t_2 which passes points $(\mathrm{id}_A, T \cdot \mathrm{Sig}_{\mathrm{PS}_A})$ where $(1 \le A \le t_2)$ and the point $(\mathrm{id}_V, H_0(M) \cdot \mathrm{PS}_V)$.

(ii) generating points. The CH_j generates additional points $(P_1, \ldots, P_{t_2}) = ((x_1, y_1), \ldots, (x_{t_2}, y_{t_2}))$ on the verifying-polynomial and then sends the tuple to the verifier V:

$$\left(T \cdot \mathrm{Sig}_{\mathrm{PS}_1}, \ldots, T \cdot \mathrm{Sig}_{\mathrm{PS}_{t_2}}, P_1, \ldots, P_{t_2}, \mathrm{HMAC}_{\mathrm{PS}_V}\left(v_M^{\mathrm{CH}_j}(0), T \cdot \mathrm{Sig}_{\mathrm{PS}_1}, \ldots, T \cdot \mathrm{Sig}_{\mathrm{PS}_{t_2}}, P_1, \ldots, P_{t_2}\right)\right). \tag{7}$$

(3) Generate verification: to check the validity of a signature, the verifier does the following.

(a) Performing secret reconstruction algorithm. The verifier performs a secret reconstruction algorithm using points $(P_1, \ldots, P_{t_2})$ and the point $(\mathrm{id}_V, H_0(M) \cdot \mathrm{PS}_V)$. The verifying polynomial can be reconstructed as follows:

$$v_M^{\mathrm{CH}_j}(x) = \sum_{i \in N_1} \lambda_i(x) Z_i, \pmod q,$$
$$Z_i = \begin{cases} P_i, & (1 \le i \le t_2), \\ H_0(M) \cdot \mathrm{PS}_i, & (i = V), \end{cases} \tag{8}$$

where $N_1 = \{1, \ldots, t_2, V\}$, $\lambda_i(x) = \prod_{j \in N \setminus i}((\mathrm{id}_i - x)/(\mathrm{id}_j - \mathrm{id}_i))$ is called a Lagrange coefficient.

(b) Checking the validity of HMAC. The verifier checks the validity of HMAC using received t_2 points and generated $v_M^{\mathrm{CH}_j}(0)$. If HMAC is correct, the verifier identifies signatures as valid.

6. Analysis

In this section, we provide analysis of our system with respect to correctness, performance, and security.

6.1. Correctness. Note that

$$f_m^{\mathrm{CH}_j}(x) = K_m + \sum_{i=1}^{t_1 - 1} b_i x^i = \sum_{i=1}^{t_1 - 1} \lambda_i(x) \mathrm{PS}_i,$$
$$\mathrm{Sig}_{\mathrm{PS}_i} = H_0(M) \cdot S_{\mathrm{PS}_i} = H_0(M) \cdot \mathrm{PS}_i, \tag{9}$$

where PS_i is the secret sharing of K_m.

Table 2: Comparisons among other ID-based threshold signature schemes.

	[7]	[18]	[19]	Proposed scheme
Distributed PKG	∘	×	×	∘
Entity anonymity	×	×	×	∘
Cost of threshold signature				
Cost of signature generation (for one signer)	$4T_P+3tT_E+2T_{M_1}+3tT_{M_2}$	$T_P+tT_E+T_{M_1}+tT_{M_2}$	$2T_P+2tT_E+(2t+2)T_{M_1}+2T_{M_2}$	$2T_{M_2}$
Cost of signature validity check	$3tT_P+tT_{M_1}+tT_{M_2}$ (for one signer)	$T_P+2tT_E+2tT_{M_2}$ (for one signer)	Include in signature generation	$(t^2+1)T_{M_2}$ (for one CH)
Cost of signature verification (for the verifier)	$6T_P+2T_M+2T_S$	$2T_P+T_E+T+H$	$2T_P+2T_{M_1}$	$(t^2-t)T_{M_2}$

∘ denotes the scheme, and considers the property; × denotes the scheme and does not consider the property; T_P denotes that of one pairing operation; T_E denotes that of one exponentiation operation; T_{M_1} denotes that of scalar multiplication in G_1; T_{M_2} denotes that of one scalar multiplication in G_2; t denotes that signing nodes.

Therefore, a CH can verify the validity of each signature as follows:

$$\begin{aligned}\sum_{i\in N}\lambda_i(x)\,Y_i &= \sum_{i=1}^{t_2}\lambda_i(x)\,\mathrm{Sig}_{\mathrm{PS}_i}+\sum_{i=t_2+1}^{t_1}\lambda_i(x)\,H_0(M)\,\mathrm{PS}_i\\ &= H_0(M)\cdot\left\{\sum_{i\in N}\lambda_i(x)\,\mathrm{PS}_i\right\}\\ &= H_0(M)\cdot f_m^{\mathrm{CH}_j}(x),\\ H_0(M)\cdot f_m^{\mathrm{CH}_j}(0) &= H_0(M)\cdot K_m.\end{aligned}\tag{10}$$

The verifier also can check the validity of threshold signature as above, similarly, because $v_M^{\mathrm{CH}_j}(x)$ passes a set of points $(P_1, P_2, \ldots, P_{t_2})$. The verifier can reconstruct the polynomial and find $v_M^{\mathrm{CH}_j}(0)$ and, consequentially, check the validity of HMAC received from a CH.

6.2. Performance. This section presents our efficiency analysis. Table 2 compares our proposed schemes with other schemes. For simplicity, we omit private key distribution and secret sharing distribution process in comparison to computation load.

Most schemes use pairing algorithm to generate and verify the signature except the proposed scheme. The pairing algorithm and the exponentiation generally consume heavy computation loads rather than scalar multiplications. Cao et al. [15] showed a pairing that consumes about double computation load with those of an exponentiation and three times computation load with those of scalar multiplication in G_1. According to this experiment, our scheme is much lighter than previous threshold signature schemes. Moreover, our threshold scheme is comparable with existing threshold signatures in non-ID-based cryptosystem [16].

6.3. Security

6.3.1. Privacy. Our security system supports the anonymity of nodes using pseudonyms. In our proposed security system, every node does not reveal real identities after the pseudonym generation. An adversary cannot correctly match an identity with a pseudonym even though the identity and the pseudonym are released to them because of the hardness of DLP [17]. The pseudonym is in the form of $\mathrm{PS}_A = f_m^{\mathrm{CH}_j}(\mathrm{id}_A)$ in the mth update phase for node A in cluster j. To match a real identity with a pseudonym, adversaries should reconstruct a polynomial $f_m^{\mathrm{CH}_j}(x)$. However, no malicious nodes at most number of $(t-1)$ can reconstruct respective polynomials, although they have known about a cluster key K_m because (t_1, n_1)-SS is information theoretically secure against at most (t_1-1) adversaries. Thus, as long as the CH does not reveal the respective polynomials, anonymity of each node is guaranteed.

6.3.2. Traceability. Each CH records the relation of the identity and the pseudonym of common nodes in its cluster at the PLT. Pseudonyms of compromised and revoked nodes are rejected from the PLT, and these nodes cannot be updated any more. Therefore, our system enables the tracing of violators when unlawful actions are notified to the CH, and they are eventually isolated from the network.

6.3.3. Nonmanipulation. Our proposed security system ensures nonmanipulation in case of at most $(t-1$ or $t_1-1)$ compromised nodes. Only the CH who has a respective polynomial can generate valid pseudonyms in a cluster, and no other nodes and CHs can do it. Secret sharing, $g_m(x)$, is generated by (t, n)-SS; therefore, no more than t CHs who are colluding can reconstruct $g_m(x)$ and learn K_m; and, to conclude, generate valid respective polynomials $f_m^{\mathrm{CH}_k}(x)$. Moreover, the respective polynomials generated by (t_1, n_1)-SS are used to generate pseudonyms. In conclusion, adversaries who know more than t secret sharing of $g_m(x)$ or more than t_1 secret sharing of $f_m^{\mathrm{CH}_k}(x)$ can carry out manipulation; however, it is impractical. Thus, as long as (t or t_1) or more than (t or t_1) nodes and CHs are not colluding, non-manipulation is guaranteed.

6.3.4. Verifiability, Undeniability, and Unforgeability. The CH works as a subverifier in our threshold signature scheme by generating a verifying polynomial. From message tuples from signers, the CH can check the validity of signers. First, only valid and registered nodes could generate correct HMAC with its pseudonym. Second, the CH could verify signatures by reconstructing a polynomial $H_0(M) \cdot f_m^{\mathrm{CH}_j}(x)$. This polynomial returns a correct value $H_0(M) \cdot f_m^{\mathrm{CH}_j}(0) = H_0(M) \cdot K_m$ when signatures are generated by signers who have a valid and registered pseudonym key pair. Finally, the reconstructed verifying polynomials generated by the CH and by the verifier return a same value, $v_M^{\mathrm{CH}_j}(0)$, when the CH satisfies two former conditions and has a valid pseudonym key pair of the verifier. Thus, verifiability is guaranteed if these conditions are acceptable. Undeniability and unforgeability are similarly guaranteed. The message containing the signature is in the form of $\mathrm{HMAC}_{\mathrm{PS}_A}$. Unless the pseudonym is released, only the node A can generate valid HMAC; thus, it cannot deny the signature and others cannot forge it. Thus, undeniability and unforgeability are guaranteed.

6.3.5. Forward/Backward Secrecy. Our security system supports the forward/backward secrecy using key update scheme in case of at most ($t - 1$ or $t_1 - 1$) compromised nodes. The key update scheme regularly updates pseudonyms, and, consequently, pseudonym public/private key pairs of nodes also are updated along to the updated pseudonym. And there is no relation in past key pairs and update key pairs because CHs periodically change a polynomial $f_m^{\mathrm{CH}_j}(x)$ or a cluster key K_m. Therefore, although the updated private/public key pairs are exposed to adversaries, these do not affect past and future session keys.

7. Conclusions

Concerns for personal privacy and security in wireless environments are increasing rapidly as mobile devices are becoming more popular. Cluster-based MANETs are being seriously considered to pioneer new markets; however, there are urgent unresolved security problems. Fundamental aspects of security, such as authentication and signature, are challenging for secure security systems for cluster-based MANETs. In addition, the protection of personal privacy has become increasingly important as the wireless networks have become personal and popular; therefore, secure security system designs with anonymity are required for cluster-based MANETs.

In this paper, we presented a secure security system with anonymity for cluster-based MANETs and a threshold signature under practical assumptions. To the best of our knowledge, our proposed security system is the first in which the pseudonym is combined with cluster-based MANETs without a trusted entity. According to our protocol analysis, our proposed system satisfies most properties for an anonymous security system and successfully copes with dynamic environments with greater efficiency by using secret sharing schemes. We believe that the proposed system improves upon the security of previously proposed security systems, and that it is suitable for a wider variety of applications. It could be usefully applied to preserve privacy in dynamic MANETs without a trusted entity, such as military battlefields, emergency areas, mobile marketplaces, and privacy-preserving VANETs.

Acknowledgments

This paper was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012R1A1A4A01002603) and was supported by the Kyungpook National University Research Fund, 2012.

References

[1] R. Dutta and T. Dowling, "Provably secure hybrid key agreement protocols in cluster-based wireless ad hoc networks," *Ad Hoc Networks*, vol. 9, no. 5, pp. 767–787, 2011.

[2] A. Shamir, "Identity-based cryptosystems and signature schemes," in *CRYPTO 84, LNCS 196*, pp. 47–53, Springer, New York, NY, USA, 1985.

[3] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *CRYPTO 01, LNCS 2139*, pp. 213–229, Springer, New York, NY, USA, 2001.

[4] X. Meng and Y. Li, "A novel verifiable threshold signature scheme based on bilinear pairing in mobile ad hoc network," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA '12)*, pp. 361–366, IEEE, 2012.

[5] Y. Takehana, I. Nishimura, N. Yosaki, T. Nagase, and Y. Yoshioka, "Building trust among certificate management nodes in mobile ad-hoc network," in *Proceedings of the 26th International Conference on Advanced Information Networking and Applications Workshops (WAINA '12)*, pp. 564–568, IEEE, 2012.

[6] J. Freudiger, M. Raya, and J. P. Hubaux, "Self-organized anonymous authentication in mobile ad hoc networks," in *SecureComm 2009, LNICST 19*, pp. 350–372, Springer, New York, NY, USA, 2009.

[7] X. Chen, F. Zhang, D. M. Konidala, and K. Kim, "New ID-based threshold signature scheme from bilinear pairings," in *INDOCRYPT 2004*, pp. 371–383, Springer, New York, NY, USA, 2004.

[8] P. Barreto, H. Kim, B. Bynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," in *CRYPTO 02*, pp. 354–369, Springer, New York, NY, USA, 2002.

[9] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[10] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: how to cope with perpetual leakage," in *CRYPTO 95*, pp. 339–352, Springer, New York, NY, USA, 1995.

[11] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 275–283, ACM, August 2000.

[12] M. Bechler, H. J. Hof, D. Kraft, F. Pählke, and L. Wolf, "A cluster-based security architecture for ad hoc networks," in *Proceedings of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 2393–2403, IEEE, March 2004.

[13] L. C. Li and R. S. Liu, "Securing cluster-based ad hoc networks with distributed authorities," *IEEE Transactions on Wireless Communications*, vol. 9, no. 10, pp. 3072–3081, 2010.

[14] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Securing mobile ad hoc networks with certificateless public keys," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 4, pp. 386–399, 2006.

[15] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Information Sciences*, vol. 180, no. 15, pp. 2895–2903, 2010.

[16] J. Hu and J. Zhang, "Cryptanalysis and improvement of a threshold proxy signature scheme," *Computer Standards and Interfaces*, vol. 31, no. 1, pp. 169–173, 2009.

[17] L. Harn and C. Lin, "Authenticated group key transfer protocol based on secret sharing," *IEEE Transactions on Computers*, vol. 59, no. 6, pp. 842–846, 2010.

[18] J. Baek and Y. Zheng, "Identity-based threshold signature scheme from the bilinear pairings (extended abstract)," in *Proceedings of the International Conference on Information Technology: Coding Computing (ITCC '04)*, pp. 124–128, IEEE, April 2004.

[19] H. Yuan, F. Zhang, X. Huang, Y. Mu, W. Susilo, and L. Zhang, "Certificateless threshold signature scheme from bilinear maps," *Information Sciences*, vol. 180, no. 23, pp. 4714–4728, 2010.