A Novel Secure Session Key Generation using two-level architecture For Cluster-Based Ad Hoc Networks Based On ID-Based Bilinear Paring

*Jue-Sam Chou [1], Yalin Chen [2], Tsung-Heng Chen [3]

[1] Department of Information Management, Nanhua University Chiayi 622 Taiwan, R.O.C

*: corresponding author

jschou@mail.nhu.edu.tw

Tel: 886+ (0)5+272-1001 ext.56226

[2] Institute of information systems and applications, National Tsing Hua University

d949702@oz.nthu.edu.tw

Tel: 886+(0)3-5738997

[3] Department of Information Management, Nanhua University Chiayi 622 Taiwan, R.O.C

wesker.harry@msa.hinet.net

Tel: 886+ (0)5-2721001 ext.2017

Abstract

In 1997, Ruppe R. et al [17] first proposed a Near-Term Digital Radio (NTDR) network system which is a cluster-based ad hoc network intended to be used efficiently for military missions. In the same year, Zavgren J. [18] proposed a management protocol for the NTDR network system. But they both lack the security considerations. In 2003, Varadharajan et al [4] proposed a secure cluster-based ad hoc network protocol using public key infrastructure (PKI). However, in 2005, Chang et al pointed out that using PKI would be a heavy burden for the computation of each mobile node. Hence, they proposed a protocol [5] based on Diffie-Hellman method for securing network, in the same year, Liaw et al. proposed a secured key exchange protocol [20] for securing nodes communication in mobile ad hoc networks (MANETs). In 2006, also for security purpose, Chang and Lee [6] proposed the other scheme by using nodes' identities. But after our analysis, we find that both of their protocols have some mistakes. Accordingly, we propose a new protocol based on ID-based bilinear pairing to get rid of nowadays unsolved security problem in NTDR network. After our analysis, we conclude that our scheme is not only secure but also very efficient.

Keywords: *the NTDR network system, PKI, cluster-based ad hoc network system, ID-based, bilinear pairing*

1. Introduction

Mobile ad hoc networks (MANETs) are networks which are organized by hosts and routers and do not need or require less fixed infrastructure in an open environment. It can be constructed quickly and nodes in it may change frequently to form a so-called dynamic topology. Hence, it is suitable for some missions such as military, emergency, or rescue. But due to its inherent properties, like dynamic topology, limited bandwidth and resource, and the lack of fixed infrastructure, designing a secure and efficient routing protocol in such a network becomes a challenge.

Recently, there were many applications of routing protocols developed for MANETs. During 1999 to 2004, there were three major types of routing protocols proposed. We list three proposals for representation of each type, respectively. They were: (1) Ad hoc on-demand distance vector routing (AODV) [1], (2) The dynamic source routing protocol for mobile ad hoc networks (DSR) [2], and (3) Authenticated routing for ad hoc networks (ARAN) [3], but all did not take routing efficiency and security into consideration except for [3] which intends to satisfy all of the security requirement, but it still has security flaws [12] for the source node can not authenticate all intermediate nodes in the routing path as indicated in [3].

NTDR network is a kind of MANET. But in it, mobile nodes are assigned into different clusters. Therefore, it is suitable for nodes communicating efficiently in a large area. In 2003, Varadharajan et al. proposed a scheme for securing cluster-based ad hoc networks based on PKI [4]. However, in 2005, Chang et al pointed out that using PKI would be a heavy burden for the computation of each mobile node. Hence, they proposed a secure protocol [5] based on Diffie-Hellman method to get rid of heavy computation burden, in the same year, Liaw et al. proposed a secured key exchange protocol [20] without using PKI. In 2006, Chang and Lee [6] proposed the other scheme by using a node's identity. But after our analysis, we find that all of their protocols have some mistakes. For this reason, we propose a novel secure protocol for NTDR network based on ID-based bilinear pairing which is not only very efficient but also can satisfy all of the security requirements.

This paper is organized as follows. The introduction is presented in Section 1 and the background is shown in Section 2. In Section3, we review two protocols of Chang and Lee et al.. After that, we show our protocol in Section 4. In Section 5, we make the security analysis of our proposal and finally a conclusion is given in Section 6.

2. Background

In 1984, Shamir [19] proposed an ID-based encryption and signature scheme. This is the forerunner of an ID-based cryptosystem. In an ID-based cryptosystem, each user can use his identity to create his public key to make the key distribution easier than the conventional ones. We briefly introduce the concept of ID-based bilinear pairing and the NTDR network system in subsection 2.1 and section 2.2, respectively. Then the security requirements for secure communications in MANETs will be presented in subsection 2.3.

2.1 Bilinear pairings

Let P be a generator of G_1 that is a cyclic group whose order is a prime q and G_2 be a cyclic multiplicative group of the same order q. We assume that solving the discrete logarithm problem (DLP) in both G_1 and G_2 is difficult in polynomial time. Let e: $G_1 \times G_1 \rightarrow G_2$ be a bilinear pairing satisfying the following conditions.

(1) Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$, for any $a, b \in Z$ and $P, Q, R \in G_1$.
(2) Computability: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.
(3) Non-degenerate: there exists $P \in G_1$ and $Q \in G_1$ such that $e(P, Q) \neq 1$.

2.2 Environment of a NTDR network system

In this subsection, we introduce the environment of a NTDR network and represent node operations in this network.

A NTDR network can use limited resources efficiently in a large environment in which mobile nodes are assigned into different clusters. Each cluster is composed of both the clusterhead which controls and manages the cluster, and mobile nodes which are handheld by the clusterhead. In a cluster, authorized nodes can communicate with the clusterhead directly as illustrated in Figure 1. In the figure, nodes A, B and C are in the same cluster. We assume A and B are within one hop and can communicate with each other directly. If A or B wants to communicate with C, which is not within one hop to A and B, they must communicate through the clusterhead. This case is the so-called intra-cluster. The other case of communication is that nodes are not in the same cluster as illustrated in Figure 2. In Figure 2, we assume nodes D and E are in different clusters. If they want to communicate to the other party, they each needs to transmit messages through clusterhead.

Besides, a NTDR network has the following two advantages: (1) it can use limited network resources efficiently, due to the necessity of communicating via their clusterhead when nodes are not within one hop, and (2) a clusterhead can monitor the nodes in the cluster when they transmit message through the clusterhead.

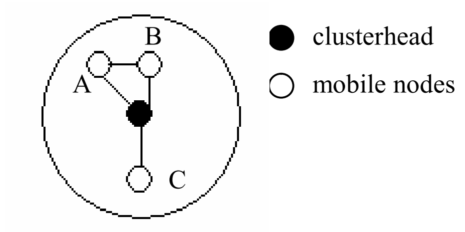


Figure1. Nodes communicate in the same cluster in a NTDR ad hoc network

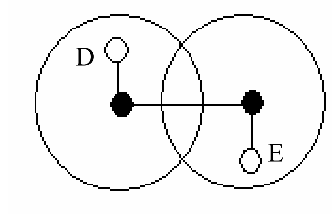


Figure2. Nodes in a NTDR ad hoc network communicate with each other in different clusters

2.3 Security requirements in a NTDR network

In this subsection, we will review the requirements of a secure communication which are not only for MANETs but also for traditional wired or infrastructure-based wireless networks. We delineate them as follows.

(1) Authentication: Only authorized and intended users can communicate to each other.
(2) Confidentiality: Only authorized users can access the correct message.
(3) Data-integrity: When messages transmitted in the network, it must be kept intact.
(4) Non-repudiation: A user can not deny the message sent by him before.
(5) Non-impersonation: Malicious users can not impersonate other authorized users to send or obtain valid information.
(6) Against key-compromise impersonation (KCI) attacks: The KCI attack means if the private key of user A is compromised, then an adversary can impersonate the other user to communicate with user A. Thus, a secure protocol needs to resist such an attack.
(7) Against man-in-the-middle attack: The man-in- the-middle attack means that an adversary E intercepts the transmitted messages between A and B and then modifies the intercepted messages to make two session keys to impersonate A to B and impersonate B to A, respectively.
(8) The forward secrecy: When a user is revoked by the group manager or leaves the group, he can not learn any future messages of the group.
(9) The backward secrecy: When a user becomes a new member of a group, he can not get any valid messages transmitted in the group.

3. Review of Chang and Lee et al. and Liaw et al.'s methods

In this section, we will first show the definitions of the notations used in the two authentication phases proposed by Chang and Lee et al. in 2005 [5] and 2006 [6], respectively. Then, we briefly review the two authentication phases in subsection 3.2

and secured key exchange protocol proposed by Liaw et al. in Section 3.3.

3.1 Definitions of used notations of Chang and Lee et al.s' protocols

In this subsection, we depict the notations used in Chang et al.'s protocols as follows:

Mh_i/MID_i: the identity of mobile node i

$CERT_X$: the public key certificate of node X

CID_j: the identity of cluster j

$CHID_j$: the identity of clusterhead j that dominates cluster j

$E_K/D_K[M]$: the encryption/decryption result of the message m encrypted/decrypted by the key K

T: timestamp

AUC: the authentication token

K_{MH}/K_{ij}: the session key shared by mobile node i and clusterhead j

3.2 Review of the two authentication phases in [5] and [6]

(a) Review of Chang and Lee et al.'s protocol [5]

In 2005, Chang et al. proposed the DH-based communication method for cluster-based ad hoc networks, but we find there is a mistake in their authentication phase. Since when a mobile host mh_i enters the radio range of a cluster CID_j and is detected by the clusterhead $CHID_j$, then $CHID_j$ and mh_i both will transmit their corresponding certificate to each other for authentication. Both of them want to authenticate each other by way of PKI. The authentication phase is as shown Figure 3. But they each does not check the validity of the certificate of the other party. Thus, an adversary E can easily impersonate one party to the other. In one words, in fact, their scheme can not achieve the goal of mutual authentication as claimed.

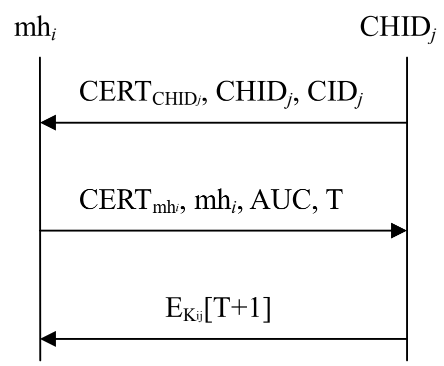


Figure 3. The authentication phase in [5]

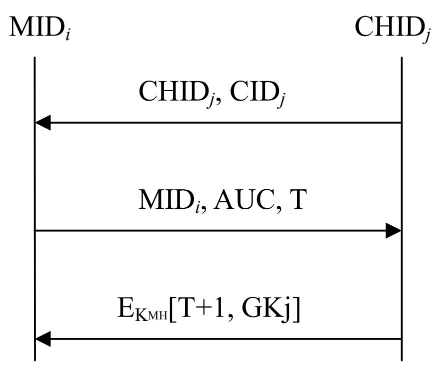


Figure 4. The authentication phase in [6]

(b) Review of Chang and Lee et al.'s protocol [6]

In 2006, Chang and Lee proposed the secure communication for cluster-based ad hoc networks using node identities. Their authentication phase is as illustrated in Figure 4. In the figure, MID_i and CHID_j each will compute an authentication token AUC after they have received the identity of the other party. But we find that the authentication tokens they calculate are not equal for $\mathrm{K}_{\mathrm{MH}} \neq \mathrm{K}_{\mathrm{MH}}'$. Thus, their authentication phase fails. The calculation of authentication tokens, AUC and AUC', are listed as follows:

$$\mathrm{AUC} = H(\mathrm{K}_{\mathrm{MH}}) = H((\mathrm{CHID}_j^2)^{H(\mathrm{T})*\mathrm{K}_i}) = H((\mathrm{CHID}_j^2)^{H(T)*e(\log_g(\mathrm{MID}_i)^2}$$

$$\mathrm{AUC'} = H(\mathrm{K}_{\mathrm{MH}}') = H((\mathrm{MID}_i^2)^{H(\mathrm{T})*C\mathrm{K}_j}) = H((\mathrm{MID}_i^2)^{H(T)*e(\log_g(\mathrm{CHID}_j)^2}$$

Here, the parameter g is the primitive element and e is TA's public key selected from $Z_{\phi(N)}^*$.

3.3 Review of Liaw et al.'s protocol [20]

Liaw et al. proposed a secured key exchange protocol for securing nodes communication of the network in 2005. But after we analysis, we find an adversary can easily obtain the session key shared with two nodes. We review and lunch an attack as follows.

3.3.1 Definition of used notations in Liaw et al.'s protocol

In this subsection we depict the notations used in Liaw et al.'s protocol as follows:

KGC: the key generation center

ID_i: identification number of user i

p, q: large and strong primes

n: the product of p and q; $n = pq$

$\phi(n) = (p-1)(q-1)$

e: a large prime is also a public key of KGC

d: a private key of KGC; $d = e^{-1} \bmod \phi(n)$

α : primitive element of $GF(p)$ and $GF(q)$

f( ): one-way hash function

g_i: a signature of user i computed by KGC

3.3.2 Four phase of Liaw et al.'s protocol

In this subsection, we describe four phases of their protocol as follows:

(a) Initialization phase

In this phase, the KGC calculates public key (n, e) and private key $(p, q, d, \phi(n))$.

(b) Registration phase

In this phase, user *i* needs to register to the KGC. First, he sends his identification number (ID_i) to the KGC, then he can obtain unique signature $g_i = ID_i^d \bmod n$ computed by the KGC. When user *i* receives the signature g_i from the KGC means registration complete, then the KGC can be closed or off-line, but we think this assumption is not pactice. Because nodes in MANETs change very frequently, due to this reason, the KGC needs to keep on-line for nodes registration.

(c) user verification phase

In this phase, the registered user *i* and *j* wants to communicate to each other, before generating the session key, they need to verify each other. We describe it using the following steps.

Step1: User *i* chooses a random number r_i and calculates two public keys as $y_i = g_i \cdot \alpha^{r_i} \bmod n$ and $t_i = r_i^e \bmod n$. Then user *i* uses ID_j and timestamp T_i for generating $f(y_i, t_i, T_i, ID_j)$ and computes $s_i = g_i \cdot r_i^{f(y_i, t_i, T_i, ID_j)} \bmod n$. Finally, user *i* sends ID_i, y_i, t_i, s_i, and T_i to user *j*.

Step2: Similarly, user *j* sends ID_j, $y_j = g_j \cdot \alpha^{r_{ji}} \bmod n$, $t_j = r_j^e \bmod n$, $s_j = g_j \cdot r_j^{f(y_{ji}, t_{ji}, T_{ji}, ID_i)} \bmod n$ and timestamp T_j to user *i*.

Step3: After receiving messages from each other, user *i* checks $s_j^e \stackrel{?}{=} ID_j \cdot t_j^{f(y_j, t_j, T_j, ID_i)} \bmod n$ for verifying user *j*. If $s_j^e = ID_j \cdot t_j^{f(y_j, t_j, T_j, ID_i)} \bmod n$ then user *j* is valid. Similarly, user j verifies user *i* by checking $s_i^e \stackrel{?}{=} ID_i \cdot t_i^{f(y_i, t_i, T_i, ID_j)} \bmod n$.

(d) key exchange protocol

After completing the user verification phase, user *i* and *j* can compute the session key as $SK_i = \left(\frac{y_j^e}{ID_j}\right)^{r_i} \bmod n = \left(\frac{y_i^e}{ID_i}\right)^{r_j} \bmod n = SK_j = \alpha^{e r_i r_j} \bmod n$.

3.3.3 Our attack for Liaw et al.'s protocol

In this subsection, we will launch an attack to obtain the session key $SK_i = SK_j$ shared with user *i* and *j* described as follows:

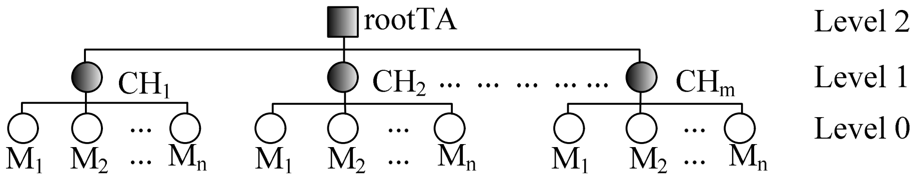
We assume an adversary E intercepts the public information t_i and t_j, then he sets his identification number $\mathrm{ID_E} = t_i \cdot t_j \bmod n$ and sends it to the KGC. After receiving the $\mathrm{ID_E}$ from E, the KGC computes $\mathrm{g_E} = \mathrm{ID_E^d} \bmod n = (t_i \cdot t_j)^{e \cdot d} \bmod n = (r_i \cdot r_j)^{e \cdot d} \bmod n = (r_i \cdot r_j)^{e \cdot e^{-1}} \bmod n = r_i \cdot r_j \bmod n$ and sends $\mathrm{g_E}$ to E. When E obtains $\mathrm{g_E}$, then he can construct the session key as $SK_\mathrm{E} = \alpha^{e r_i r_j} \bmod n = SK_i = SK_j$ by using the public parameter α and *e*. Thus, E can decrypt the messages shared with user *i* and *j*.

4. Our proposed protocol

In this section, we will first describe our protocol's environment in section 4.1 and then the definitions of used notations in section 4.2. Finally, we present our scheme in section 4.3. Our protocol bases on the NTDR network model without using PKI and includes three phases as follows: (1) session key generation phase for nodes in a cluster (2) group key generation phase for a cluster and all clusters. (3) session key generation phase for nodes in different clusters. In phase (1), we describe how a valid node can get the session key to achieve the following goals: (a) communicates with his clusterhead, (b) communicates with another node within one hop in the same cluster (c) communicates with another node which is not within one hop but in the same cluster. In phase (2), we will depict how to generate both of a cluster group key for all nodes in the same cluster to communicate and a clusterhead group key for all clusterheads communicating in different clusters. In phase (3), we will show how two nodes in different clusters can get their session keys.

4.1 Two-level hierarchy environment

In our protocol, we assume that each clusterhead is the trust third party (TA) for each node in the same cluster, and all clusterheads in different clusters are managed by the rootTA. In other words, there is not only a clusterhead in a cluster but also a clusterhead for all clusterheads in the clusterhead group. That is, our protocol is a 2-level structure in hierarchy. TAs in different clusters are at level 1 and rootTA that manages all TAs is at level 2. The TAs each computes a private key and the corresponding public key for each of his registered member in his cluster. Similarly, rootTA will do the same thing for each of the clusterhead in the group of clusterheads managed by him as illustrated in Figure 4.



The public/private key pair Q_{M_i}/S_{M_i} for mobile node *i* at level 0

The public/private key pair P_{CH_j}/s_j for clusterhead *j* (TAs for M_i) at level 0

The public/private key pair Q_{CH_j}/S_{CH_j} for clusterhead *j* at level 1

The public/private key pair P_{root}/s_{root} for rootTA (TAs for clusterhead *j*) at level 1

Figure 4. Two-level environment in our proposed protocol

The public/private key pair for node M_i at level 0 is Q_{M_i}/S_{M_i} and for his corresponding clusterhead (TA) is P_{CH_j}/s_j. Similarly, the key pair for the clusterhead CH_j at level 1 managed by rootTA is Q_{CH_j}/S_{CH_j} and the key pair for rootTA at level 2 is P_{root}/s_{root}.

4.2 Definitions of used notations

In this subsection, we define the notations used in our protocol as follows:

M_i: the identity of mobile node *i*

CH_j: the identity of clusterhead *j* which manages cluster *j*

CID_j: the identity of cluster *j*

s_j: the private key of clusterhead *j* which is also a TA of cluster *j*

P_{CH_j}: the public key of clusterhead *j*

$H(.)$: an one way hash function which maps a point in G_1 to a bit string

$Q_{M_i} = H(M_i)$: the long-term public key of M_i, and Q_{M_i} belongs to G_1

$S_{M_i} = s_j Q_{M_i}$: the long-term private key of M_i issued by CH_j.

i: the short-term private key of M_i which is a random number chosen by M_i

$P_{M_i} = iP$: the short-term public key of M_i

$K_{M_iH_j}$: the session key shared between M*i* and CH_j

SK_{AB}: the session key shared between mobile nodes A and B

CGK_j: the group key of cluster *j*

CHGK: the group key of all clusterheads in the clusterhead group managed by rootTA

r_i: the random number chosen by M*i*

T: timestamp

$E_K/D_K[M]$: the encryption/decryption result of message M en/decrypted by key K

4.3 Our Proposed Scheme

In this Section, we describe the three phases in our scheme as follows.

4.3.1 Session key generation phase

In this phase, we describe the following three cases:

(a) For a node entering a cluster to communicate with his clusterhead

In this case, when a mobile node M_i enters the radio range of cluster j and is detected by clusterhead CH_j, CH_j will generate the session key with his cooperation as illustrated in Figure 5 which is also described using the following steps.

M_i	CH_j
(1) CH_j, CID_j, P_{CHj}, T_1, r_cP_{CHj}, Msg_{CHj}	
(2) M_i, r_iQ_{Mi}, CH_j, CID_j, T_2, Msg_{Mi}	

$$\text{Msg}_{\text{CH}_j} = H(e(h_{ij} \times r_i Q_{\text{M}_i}, r_c s_j P_{\text{CH}_j}))$$
$$\text{Msg}_{\text{M}_i} = H(e(s_j Q_{\text{M}_i}, h_{ij} \times r_i r_c P_{\text{CH}_j}) + e(s_j Q_{\text{M}_i}, P_{\text{CH}_j}))$$

Figure 5. Session key generation phase for a node with his clusterhead

Step1: CH_j chooses a random number r_c, computes r_cP_{CHj} and Msg_{CHj}. The computation of Msg_{CHj} is shown in the figure, where the value $h_{ji} = H(e(s_j P_{\text{CH}_j}, Q_{\text{M}_i}))(= h_{ij})$. Then, CH_j sends the beacon message composed of CH_j, CID_j, P_{CHj}, r_cP_{CHj}, Msg_{CHj} and timestamp T_1 to M_i.

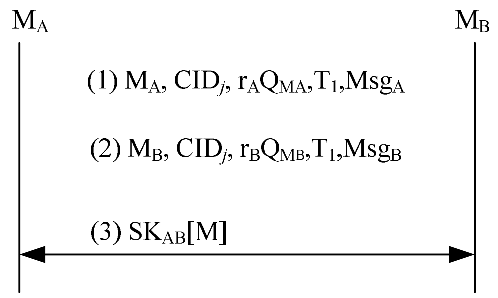
Step2: When receiving the beacon message from CH_j, M_i checks the validity of timestamp T_1 and computes $\text{Msg}_{\text{CH}_j}' = H(e(r_i s_j Q_{\text{M}_i}, h_{ij} \times r_c \text{P}_{\text{CH}_j}))$. If T_1 is not valid or Msg_{CHj}' is not equal to Msg_{CHj}, M_i interrupts the communication. Otherwise, M_i sends r_iQ_{Mi}, CH_j, CID_j, $\text{Msg}_{\text{M}_i} = H(e(s_j Q_{\text{M}_i}, h_{ij} \times r_i r_c P_{CH_j}) + e(s_j Q_{\text{M}_i}, P_{\text{CH}_j}))$, timestamp T_2 together with its identity M_i to CH_j. The value h_{ij} in Msg_{Mi} is equal to $H(e(s_j Q_{\text{M}_i}, P_{\text{CH}_j}))$. Then, M_i uses P_{CHj} to compute the pre-session key $K_{Mi\text{-}CHj}$ as.

$\text{K}_{\text{M}i-\text{CH}j} = e(S_{\text{M}i}, r_i P_{\text{CH}j}) = e(s_j Q_{\text{M}i}, r_i s_j P) = e(Q_{\text{M}i}, P)^{r_i s_j^2}$. After obtaining this pre-session key, M_i computes the session key shared with CH_j as $\text{K}_{\text{M}_i\text{H}_j} = H(\text{K}_{\text{M}_i-\text{CH}_j} \| \text{M}_i \| \text{CH}_j)$

Step3: After receiving the message from M_i, CH_j checks to see if the timestamp T_2 is valid, if it is not valid, he terminates the communication, else he computes $\text{Msg}_{\text{M}_i}' = H(e(h_{ji} \times r_i Q_{M_i}, r_c s_j P_{CH_j}) + e(s_j P_{\text{CH}_j}, Q_{\text{M}_i}))$. If $\text{Msg}_{\text{M}_i}' = \text{Msg}_{\text{M}_i}$, CH_j then computes the pre-session key $\text{K}_{\text{CH}j\text{-M}i}$ as $\text{K}_{\text{CH}j-\text{M}i} = e(r_i Q_{\text{M}i}, s_j P_{\text{CH}j}) = e(Q_{\text{M}i}, P)^{r_i s_j^2}$, which is equal to $\text{K}_{\text{M}i\text{-CH}j}$, else he terminates the communication. After obtaining this pre-session key, CH_j computes the session key shared with M_i as $\text{K}_{\text{M}_i\text{H}_j} = H(\text{K}_{\text{CH}_j-\text{M}_i} \| \text{M}_i \| \text{CH}_j)$.

(b) For nodes to communicate within one hop in the same cluster

When nodes are within one hop in the same cluster, they can communicate to each other directly. We delineate the session key generation under this situation in Figure 6 and describe it using the following steps.



$$\text{Msg}_\text{A} = H(e(S_{\text{M}_\text{A}}, h_{\text{AB}} \times r_A r_B Q_{\text{M}_\text{B}}))$$
$$\text{Msg}_\text{B} = H(e(S_{\text{M}_\text{B}}, h_{\text{BA}} \times r_B r_A Q_{\text{M}_\text{A}}) + e(S_{\text{M}_\text{B}}, Q_{\text{M}_\text{A}}))$$

Figure 6. Session key generation for nodes to communicate within one hop in the cluster

Step1: M_A chooses a random number r_A, computes $r_\text{A}Q_{\text{M}_\text{A}}$ and $\text{Msg}_\text{A} = H(e(S_{\text{M}_\text{A}}, h_{\text{AB}} \times r_A r_B Q_{\text{M}_\text{B}}))$ where value h_{AB} in Msg_A is equal to $H(e(S_{\text{M}_\text{A}}, Q_{\text{M}_\text{B}}))$, then he sends $r_\text{A}Q_{\text{M}_\text{A}}$, Msg_A, CID_j and timestamp T_1 to M_B together with its identity M_A.

Step2: After receiving the message from M_A, M_B checks the validity of T_1. If it is valid, M_B computes $\text{Msg}_\text{A}' = H(e(h_{\text{AB}} \times r_B r_A Q_{\text{M}_\text{A}}, S_{\text{M}_\text{B}}))$, if Msg_A' is not equal Msg_A. M_B terminates the communication, else M_B selects a random number r_B and computes $r_\text{B}Q_{\text{M}_\text{B}}$ and Msg_B where h_{BA} in Msg_B is equal to $H(e(S_{M_B}, Q_{M_A}))$, then he sends $r_B Q_{M_B}$, timestamp T_2 and Msg_B to M_A together with M_B and CID_j. After that, he computes the pre-session key as

$$K_{BA} = e(r_A Q_{M_A}, r_B S_{M_B}) = e(r_A Q_{M_A}, r_B s_1 Q_{M_B}) = e(Q_{M_A}, Q_{M_B})^{r_A r_B s_1}$$ and then computes the session key as $SK_{AB} = H(K_{BA} \| M_A \| M_B)$.

Step3: When obtaining the message sent from M_B, M_A checks the validity of T_2, if T_2 is invalid, M_A interrupts the communication. Else, M_A computes $Msg_B' = H(e(h_{AB} \times r_A r_B Q_{M_B}, S_{M_A}) + e(S_{M_A}, Q_{M_B}))$. If the $Msg_B' = Msg_B$ then he computes the pre-session key as

$$K_{AB} = e(r_A S_{M_A}, r_B Q_{M_B}) = e(r_A s_1 Q_{M_A}, r_B Q_{M_B}) = e(Q_{M_A}, Q_{M_B})^{r_A r_B s_1}$$ and then computes the session key as $SK_{AB} = H(K_{AB} \| M_A \| M_B)$.

After completing the above steps, M_A and M_B both can obtain their session key SK_{AB}.

Similarly, we can use the same method to generate the session key between two clusterheads CH_1 and CH_2 in different clusters by replacing M_A with CH_1 and M_B with CH_2, respectively. We show the computation of the session key $SK_{H_1H_2}$ for CH_1 and CH_2 as follows:

For CH1, he computes $SK_{H_1H_2} = H(e(r_{CH_1} S_{CH_1}, r_{CH_2} Q_{CH_2}) \| CH_1 \| CH_2)$. For CH2, he computes $SK_{H_1H_2} = H(e(r_{CH_1} Q_{CH_1}, r_{CH_2} S_{CH_2}) \| CH_1 \| CH_2)$, where Q_{CH_1} and Q_{CH_2} are the corresponding public key of CH_1 and CH_2, r_{CH_1} and r_{CH_2} are random numbers chosen by CH_1 and CH_2, respectively and s_{root} used in CH_1 and CH_2 is the private key of rootTA who is a TA of all clusterheads. Besides, CH1 and CH2 each also needs to compute $Msg_{CH_1} = H(e(S_{CH_1}, h_{12} \times r_{CH_1} Q_{CH_2}))$ and $Msg_{CH_2} = H(e(S_{CH_2}, h_{21} \times r_{CH_2} Q_{CH_1}))$ respectively for authenticating the random number chosen by the other party. Finally, CH_1 and CH_2 each can obtain the session key $SK_{H_1H_2}$.

(c) For nodes, beyond one-hop apart in the same cluster, to communicate with each other through clusterhead

We assume that there are two nodes, M_A and M_B, in the same cluster but not within one hop want to transmit messages through the clusterhead CH_1. We delineate how they can get their session key in Figure 8 and also describe it using the following steps.

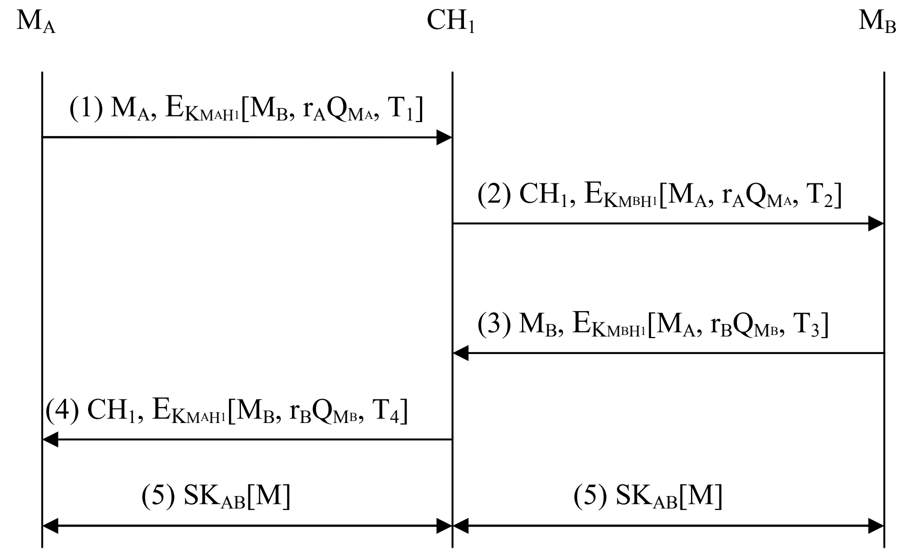


Figure 7. Session key generation for nodes communicating through the clusterhead in the same cluster

Step1: M_A selects a random number r_A to computes $r_AQ_{M_A}$. He then transmits the encryption of the message composed of M_B, $r_AQ_{M_A}$ and timestamp T_1 by using the session key $K_{M_AH_1}$ shared between M_A and CH_1, together with his identity M_A to CH_1.

Step2: When receiving the message sent by M_A, CH_1 uses the session key $K_{M_AH_1}$ to decrypt the message and obtain M_B, $r_AQ_{M_A}$, and T_1. He then checks the validity of T_1. If the message is in time then CH_1 uses the session key $K_{M_BH_1}$ shared with M_B to encrypt the M_A, $r_AQ_{M_A}$, and timestamp T_2 and then sends this encrypted message together with its identity CH_1 to M_B.

Step3: After obtaining the encrypted message from CH_1, M_B decrypts it using the session key shared with CH_1 to get M_A, $r_AQ_{M_A}$, and T_2. M_B then checks the validity of timestamp T_2, if T_2 is overdue, he rejects the communication, else he chooses a random number r_B and computes $r_BQ_{M_B}$. He then encrypts the message consisting of M_A, $r_BQ_{M_B}$, and timestamp T_3 using the session key $K_{M_BH_1}$ and sends this encryption together with his identity M_B to CH_1. After this, he can compute the session key (shared with M_A) to be $SK_{AB} = H(K_{BA} \| M_A \| M_B)$, which is the same value as computed in step2 of case (b) in this section.

Step4: After receiving the encrypted message sent by M_B, CH_1 uses the session key $K_{M_BH_1}$ to decrypt the message and obtains M_A, $r_BQ_{M_B}$ and timestamp T_3. Then CH_1 checks the validity of T_3, if T_3 is not valid, CH_1 stops communicating with M_B; otherwise, he uses the session key $K_{M_AH_1}$ shared with M_A to encrypt the message including M_B, $r_BQ_{M_B}$ and timestamp T_4, then CH_1 sends this encrypted message along with his identity CH_1 to M_A.

Step5: When receiving the encrypted message sent by CH_1, M_A uses the session key $K_{M_AH_1}$ to decrypt this encrypted message and obtains M_B, $r_BQ_{M_B}$ and timestamp T_4. Then M_A checks the validity of T_4. If T_4 is valid, he computes the session key SK_{AB} to en/decrypt messages for communicating with M_B through clusterhead CH_1. The computation of the session key is $SK_{AB} = H(K_{AB} \| M_A \| M_B)$, which is the same value as computed in step3 of case (b) in this section.

4.3.2 Group key generation phase for a cluster and for the group of clusterheads

In this phase, we describe the group key generation phase in two cases: (a) group key generation for a cluster and (b) group key generation for the group of clusterheads.

(a) Group key generation for a cluster

We delineate the group key generation phase for a cluster in Figure 8 and describe it using the following steps.

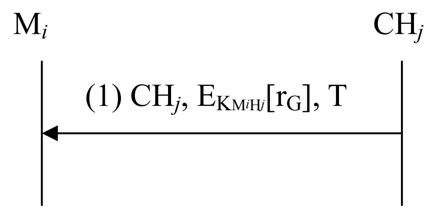


Figure 8. CH_j sends r_G to M_i for generating group key

Step1: After generating session key $K_{M_iH_j}$ with each node M_i, i = A, B,…, and N in phase 1 as described in Section 4.3.1. Here, we assume that there are n mobile nodes in the cluster. They are node A, B, C,…, and N, CH_j sends his identity CH_j, the encrypted r_G and timestamp T to each M_i for creating cluster group key.

Step2: When each M_i obtains the message from CH_j. He checks the validity of timestamp T. If it is not correct, M_i will interrupt the communication. Else, he broadcasts the message consisting of $H(K_{M_iH_j}, P_{M_i}, M_i)$ and P_{M_i}

to all nodes in the cluster, where P_{M_i} is his short-term public key. (This message also can let the clusterhead to detect out which one is the cheater when there exists a malicious node broadcasting the wrong P_{M_i}.)

Step3: Each node in the cluster decrypts the encrypted r_G sent by CH_j and uses P_{M_i}, in each node's broadcast message to calculate the same cluster group key CGK using the following equation.

$$\begin{aligned}\text{CGK} &= e(P_{M_A}, r_G P)\cdot e(P_{M_B}, r_G P)\cdot\cdots\cdot e(P_{M_N}, r_G P)\\ &= e(aP, r_G P)\cdot e(bP, r_G P)\cdot\cdots\cdot e(nP, r_G P)\\ &= e(P,P)^{ar_G}\cdot e(P,P)^{br_G}\cdot\cdots\cdot e(P,P)^{nr_G}\\ &= e(P,P)^{r_G(a+b+\cdots+n)}\end{aligned}$$

(b) Group key generation for the group of clusterheads

The computation of clusterhead group key (CHGK) for the group of all clusterheads is similar to the computation of the cluster group key (CGK) in a cluster in mentioned as the above steps in case (a) just by replacing CH_j with rootTA and M_i with CH_i. We list the calculation of the CHGK by the following equation.

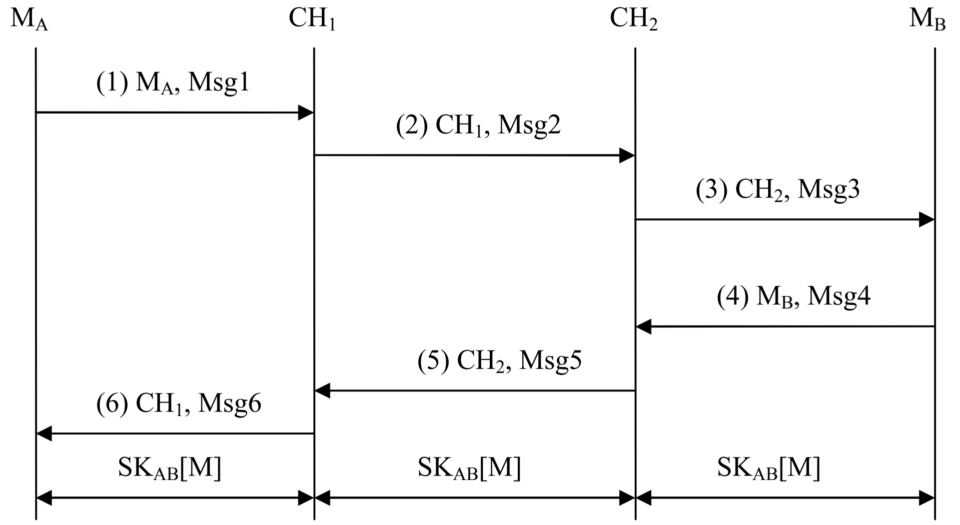
$$\text{CHGK} = e(P_{CH_1}, r_{CH}P)\cdot e(P_{CH_2}, r_{CH}P)\cdot\cdots\cdot e(P_{CH_n}, r_{CH}P) = e(P,P)^{r_{CH}(c_1+c_2+\cdots+c_n)}$$,where

$P_{CH_1}, \ldots, P_{CH_n}$ are the short-term public keys of clusterhead 1 to clusterhead n and r_{CH} is a random number chosen by rootTA.

In the cluster-based ad hoc network, nodes change frequently, thus the computation of cluster group key for a cluster in (a) and clusterhead group key for group of clusterheads in (b) needs to be recalculated once the members have changed in (a) or (b) for the consideration of the forward and backward secrecy.

4.3.3 Session key generation phase for nodes in different clusters

After completing phase 2 (Section 4.3.2), in this section, we describe how nodes in different clusters can compute their session keys. Here, we assume that mobile nodes M_A and M_B are in different clusters, we depict the process in Figure 9 and describe it using the following steps.



Msg1 = $E_{K_{M_AH_1}}$[M_B, r_A, T_1]	Msg2 = E_{CHGK}[M_A, CID_1,M_B, r_A, r_x, T_2]
Msg3 = $E_{K_{M_BH_2}}$[$M_{A,}$$CID_1$, r_A, r_x, r_y, T_3]	Msg4 = $E_{K_{M_BH_2}}$[$M_{A,}$ CID_1 , r_B, T_4]
Msg5 = $E_{K_{H_1H_2}}$[$M_{B,}$$CID_2$, M_A, r_B, r_y, T_5]	Msg6 = $E_{K_{M_AH_1}}$[M_B, $CID_{2,}$$r_B$, r_y, r_x, T_6]
r_A: random number is chosen by M_A	r_B: random number is chosen by M_B
r_X: random number is chosen by CH_1	r_y: random number is chosen by CH_2

Figure 9. Nodes communicate in the different cluster

Step1: First, M_A in clusterhead 1 chooses a random number r_A and uses the session key $K_{M_AH_1}$ shared with CH_1 to encrypt M_B, r_A and timestamp T_1, to form message Msg1, then M_A sends Msg1 to CH_1 together with its identity M_A.

Step2: After receiving Msg1, CH_1 uses the session key $K_{M_AH_1}$ to decrypt Msg1 and then checks the validity of timestamp T_1, if T_1 is not valid, CH_1 interrupts the communication with M_A, else he chooses a random number r_x and uses the clusterhead group key CHGK shared with each clusterhead to encrypt M_A, CID_1, M_B, r_A, r_x and timestamp T_2 to form Msg2, then CH_1 broadcasts Msg2 together with his identity CH_1 to all clusterheads of the network.

Step3: When all clusterheads in the network receiving Msg2 from CH_1, they can use the clusterhead group key CHGK to decrypt Msg2 and then check the validity of the timestamp T_2. If T_2 is valid, they check their database to see if the location information and identity of mobile node B (M_B) belongs to him. Here, we assume M_B belongs to cluster 2 and is managed by CH_2. CH_2 selects a random number r_y and uses session key $K_{M_BH_2}$ shared with M_B to encrypt $M_{A,}$ CID_1, r_A, r_x, r_y, and timestamp T_3

to form Msg3, then CH_2 sends Msg3 to M_B together with its identity CH_2.

Step4: After obtaining Msg3 from CH_2, M_B uses the session key $K_{M_BH_2}$ to decrypt it and get M_A, CID_1, r_A, r_x, r_y, and timestamp T_3. He then checks the validity of T_3, if T_3 is not correct, he terminates the communication with CH_2; otherwise, he randomly chooses a number r_B and then encrypts M_A, CID_1, r_B and timestamp T_4 by using session key $K_{M_BH_2}$ to form Msg4. Then M_B sends Msg4 to CH_2 together with his identity M_B. After this, M_B can compute the pre-session key $K_{BA} = e(r_A P, r_x P)e(r_B P, r_y P) = e(P,P)^{r_A r_x + r_B r_y}$. He then computes the session key as $SK_{AB} = H(K_{BA} \| M_A \| M_B)$.

Step5: After receiving Msg4 from M_B, CH_2 decrypts it using session key $K_{M_BH_2}$. Then, he checks the validity of timestamp T_4. If T_4 is overdue, he terminates the communication with M_B; otherwise, he uses session key $K_{H_1H_2}$ to encrypt M_B, CID_2, M_A, r_B, r_y, and timestamp T_5 to form Msg5, then CH_2 sends Msg5 together with his identity CH_2 to CH_1.

Step6: After receiving Msg5, CH_1 uses session key $K_{H_1H_2}$ to decrypt it, obtaining M_B, CID_2, M_A, r_B, r_x, r_y and T_5. Then he checks the validity of T_5. If T_5 is valid, he uses session key $K_{M_AH_1}$ to encrypt M_B, CID_2, r_B, r_y, r_x and timestamp T_6 to form Msg6 and then sends Msg6 to M_A together with his identity CH_1.

Step7: When receiving Msg6 from CH_1, M_A uses session key $K_{M_AH_1}$ to decrypt it and checks the validity of the timestamp T_6. If T_6 is valid, then he computes session key SK_{AB} as follows.

First, he computes the pre-session key K_{AB} as.

$$K_{AB} = e(r_A P, r_x P)e(r_B P, r_y P) = e(P,P)^{r_A r_x} \cdot e(P,P)^{r_B r_y} = e(P,P)^{r_A r_x + r_B r_y} = K_{BA},$$

then computes SK_{AB} as $SK_{AB} = H(K_{AB} \| M_A \| M_B) = H(K_{BA} \| M_A \| M_B)$.

5. Security analysis

In this section, we discuss the security of our protocol, we prove that our protocol can satisfy all the security requirements in the session key establishment including: (1) against DoS attacks (2) non-repudiation (3) against KCI attacks (4) against man in the middle attacks (5) authentication. (6) the forward/backward secrecy We describe them as follows.

(1) Against DoS attacks

In case (a) of Section 4.3.1, the values h_{ij}, $e(s_j Q_{M_i}, P_{CH_j}))$ and h_{ji} in Msg_{M_i} and Msg_{CH_j} is generated by identification information and pre-computed by M_i and CH_j. Thus, an adversary can not obtain the correct messages Msg_{M_i} and Msg_{CH_j} for mutual authentication and impersonate any users to authenticate to CH_j. Similarly, in case (b) of Section 4.3.1, an adversary still can not compromise other users to communicate with M_B. By this reason, our protocol can against DoS attacks.

(2) Non-repudiation

For the clusterhead can monitor all the messages sent by his members and can authenticate his members, we can say that nobody can deny the message he sent before since only he and the clusterhead have the session key $K_{M_iH_j}$.

(3) Against KCI attack

Here, we assume that the private key S_{M_A} of M_A had been compromised to an adversary E. We want to show that E still can not impersonate M_B to communicate with M_A. In other words, E can not obtain the session key SK_{AB} shared between M_A and M_B. Due to E can not know the random numbers, r_A chosen by M_A and r_B chosen by M_B and the private key S_{M_B} of M_B. Therefore, E can not obtain the session key SK_{AB}. For the computation of SK_{AB} equals $H(e(r_A S_{M_A}, r_B Q_{M_B}) \| M_A \| M_B)$. By this reason, our protocol can against KCI attacks.

(4) Against man-in-the-middle attack (MIMA)

Since in our scheme, phases (2) and (3) base on phase (1). If phase (1) is secure, then our scheme is secure. Hence, we only discuss MIMA in the two cases: (a) and (b) in Section 4.3.1, respectively. Assume that an adversary E wants to launch a MIMA to impersonate M_A to M_B, he can not succeed. Due to $h_{AB} = H(e(S_{M_A}, Q_{M_B}))$ is pre-computed by M_A and $h_{BA} = H(e(S_{M_B}, Q_{M_A}))$ is pre-computed by M_B, E can not know the content of h_{AB} and h_{BA}. Thus, when E wants to impersonate M_A by modifying Msg_A to M_B, he will fail. Because M_B needs to compute Msg_A' (illustrated in step 2 of case (b) in Figure 6 of Section 4.3.1.). When Msg_A' is not equal to Msg_A, then M_B interrupt the communication. Hence, E can not obtain the session key SK_{AB}.

However, if our scheme lacks the pre-computation of values h_{AB}, then E can launch a MIMA to impersonate M_A to M_B. For E can intercept the message from

$\mathrm{M_A}$ and replaces $\mathrm{Msg_A} = H(e(S_\mathrm{A}, r_A Q_{\mathrm{M_B}}))$ with $\mathrm{Msg_A} = H(e(S_\mathrm{E}, r_E Q_{\mathrm{M_B}}))$.

Then $\mathrm{M_B}$ computes $\mathrm{Msg_A}' = H(e(r_E Q_{\mathrm{M_E}}, S_\mathrm{B}))$ which is equal to $\mathrm{Msg_A}$. Consequently, E can impersonate $\mathrm{M_A}$ to communicate with $\mathrm{M_B}$.

Similarly, E can not lunch MIMA in (a) (illustrated in Figure 5.). Hence, our proposed protocol can resist against MIMA.

(5) Authentication

Here, we claim that only the intended members can communicate to each other in our protocol. Before the authentication, TA provides each member M_i a private key through a secure channel. If M_i wants to become a new member of the cluster j then M_i must register himself to CH_j (depicted in case (a) of subsection 4.3.1).

After entering the radio range of CH_j and receiving the beacon message from CH_j, M_i first computes the pre-session key $\mathrm{K}_{\mathrm{M}_i\text{-}\mathrm{CH}_j}$ (for obtaining the session key $\mathrm{K}_{\mathrm{M}_i\mathrm{H}_j}$) as $\mathrm{K}_{\mathrm{M}_i-\mathrm{CH}_j} = e(S_{\mathrm{M}_i}, r_i P_{\mathrm{CH}_j}) = e(r_i Q_{\mathrm{M}_i}, s_j P_{\mathrm{CH}_j}) = e(Q_{\mathrm{M}_i}, P)^{r_i s_j^2} = \mathrm{K}_{\mathrm{CH}_j-\mathrm{M}_i}$. Then M_i can compute the session key as $\mathrm{K}_{\mathrm{M}_i\mathrm{H}_j} = H(\mathrm{K}_{\mathrm{M}_i-\mathrm{CH}_j} \| \mathrm{M}_i \| \mathrm{CH}_j)$. He then sends the message that consists of his identity M_i, $\mathrm{r}_i\mathrm{Q}_{\mathrm{M}_i}$, identity of clusterhead CH_j, identity of cluster j CID_j, T_2 and $\mathrm{Msg}_{\mathrm{M}_i}$ to CH_j. After obtaining the message from M_i, CH_j computes the pre-session key $\mathrm{K}_{\mathrm{CH}_j\text{-}\mathrm{M}_i}$ that is equal to $\mathrm{K}_{\mathrm{M}_i\text{-}\mathrm{CH}_j}$. Then, CH_j can compute the session key $\mathrm{K}_{\mathrm{M}_i\mathrm{H}_j}$ as follows:

$$\mathrm{K}_{\mathrm{M}_i\mathrm{H}_j} = H(\mathrm{K}_{\mathrm{CH}_j-\mathrm{M}_i} \| \mathrm{M}_i \| \mathrm{CH}_j)$$

In our protocol, if the value $\mathrm{K}_{\mathrm{M}_i\text{-}\mathrm{CH}_j}$ is not equal to $\mathrm{K}_{\mathrm{CH}_j\text{-}\mathrm{M}_i}$ then we can say M_i is not authorized by CH_j, because only the authorized and intended member can generate the same session key. Therefore, our protocol can achieve the goal of authentication.

(6) The backward secrecy

Backward secrecy means that when a node becomes a new member of a cluster, it can not learn any past messages. In this subsection, we assume that a new node $\mathrm{M}_{\mathrm{N+1}}$ wants to become a member of cluster j. When he joins into cluster j, he must broadcast the message consisting of his ID $\mathrm{M}_{\mathrm{N+1}}$, his short-term public key $\mathrm{P}_{\mathrm{M}_{\mathrm{N+1}}}$and the verification message $H(\mathrm{K}_{\mathrm{M}_{\mathrm{N+1}}\mathrm{H}_j}, \mathrm{P}_{\mathrm{M}_{\mathrm{N+1}}}, \mathrm{M}_{\mathrm{N+1}})$ to all members. Meanwhile, each of the members needs to update his own broadcasted information by replacing his short-term public key $\mathrm{P}_{\mathrm{M}_i}$ with $\mathrm{P}_{\mathrm{M}_i}$'' for generating the new cluster group key including $P_{M_{N+1}}$. Then, every node of cluster *j* can reconstruct the new cluster group key CGK'' by computing as follows.

$$\begin{aligned} CGK'' &= e(P_{M_A}'', r_G P_{CHj}) \cdot e(P_{M_B}'', r_G P_{CHj}) \cdots\cdot e(P_{M_N}'', r_G P_{CHj}) \cdot e(P_{M_{N+1}}, r_G P_{CHj}) \\ &= e(P,P)^{r_G s_j (a''+b''+\cdots+n''+(n+1))} \end{aligned}$$

Apparently, CGK'' is not equal to the original group key CGK. In other hands, the new member M_{N+1} can not use this new cluster group key CGK'' to decrypt any messages encrypted by the old group CGK. Therefore, our proposed protocol can achieve the backward secrecy property.

(7) The forward secrecy

Forward secrecy means that when M_A leaves cluster *j*, all other left members in the cluster, (M_B,…, M_N), each needs to broadcast his ID M_i, i = B to N, his new short-term public key P_{M_i}' and $H(K_{M_iH_j}, P_{M_i}', M_i)$. The clusterhead then verifies the correctness of the information $H(K_{M_iH_j}, P_{M_i}', M_i)$ to authenticate M_i. These legal members then can reconstruct the new cluster group key CGK' after M_A leaves the cluster. (We denote the original cluster group key as CGK and the new cluster group key as CGK'.) We list both of their computations as follows.

$$\begin{aligned} CGK &= e(P_{M_A}, r_G P) \cdot e(P_{M_B}, r_G P) \cdots\cdot e(P_{M_N}, r_G P) \\ &= e(P,P)^{r_G (a+b+\cdots+n)} \end{aligned}$$

and

$$\begin{aligned} CGK' &= e(P_{M_B}', r_G P) \cdot e(P_{M_C}', r_G P) \cdots\cdot e(P_{M_N}', r_G P) \\ &= e(P,P)^{r_G (b'+c'+\cdots+n')} \end{aligned}$$

Apparently, the new group key CGK' is not equal to the old one, CGK, due to $P_{M_i} \neq P_{M_i}'$, for i = B, C,…, N, and the lack of short-term public key of M_A. Hence, M_A can not access any future messages encrypted by CGK' in the cluster. Thus, our proposed protocol also can satisfy the forward secrecy requirement.

6. Conclusions

Due to nodes transmiting message through the clusterhead, the architecture of the NTDR ad hoc network is especially suitable for an ad hoc network in a large communication area. For it can greatly reduce the power consumption and the clusterhead can monitor the communication messages to ensure its safety. But, there still does not exist a secure protocol which can really satisfy the security requirements when nodes communicate in a NTDR network. In this paper, we propose a novel two-level architecture for securing session key generation using ID-based bilinear pairing. We have described and proved the correctness of our protocol. Up to now, this is the first scheme which can really be implemented securely and efficiently.

References

[1] C. E. Perkins and E. M. Royer. "Ad hoc on-demand distance vector routing, "in Proc. WMCSA, New Orleans, LA, Feb. 1999, pp. 90-100.

[2] D. Johnson, D. Maltz, and Y.-C. Hu. "The dynamic source routing protocol for mobile ad hoc networks (DSR), "IEEE Internet Draft, Apr. 2003.

[3] Sanzgiri, K., LaFlamme, D., Dahill, B.. Levine, B.N.; Shields, C.; Belding-Royer, E.M.;"Authenticated routing for ad hoc networks Selected Areas in Communications, "IEEE Journal on Volume 23, Issue 3, March 2005 Page(s):598 - 610 Digital Object Identifier 10.1109/JSAC.2004.842547

[4] Vijay Varadharajan, Rajan Shankaran and Michael Hitchens. "Security for cluster based ad hoc networks, "*Computer Communications*, *Volume 27, Issue 5*, *20 March 2004*, *Pages 488-501*

[5] Chin-Chen Chang, Keng-Chu Lin, Jung-San Lee. "DH-based communication method for cluster-based ad hoc networks Mobile Technology, Applications and Systems, " the 2nd International Conference on 15-17 Nov. 2005 Page(s):8 pp.

[6] Jung-San Lee and Chin-Chen Chang. "Secure communications for cluster-based ad hoc networks using node identities, "*Journal of Network and Computer Applications*, *In Press, Corrected Proof*, *Available online 1 December 2006* Jung-San Lee and Chin-Chen Chang

[7] Hung-Yu Chien, Ru-Yu Lin. "Identity-based Key Agreement Protocol for Mobile Ad-hoc Networks Using Bilinear Pairing, "Sensor Networks, Ubiquitous, and Trustworthy Computing, IEEE International Conference on Volume 1, 05-07 June 2006 Page(s):520 - 529

[8] Fangguo Zhang and Xiaofeng Chen. "Attack on an ID-based authenticated group key agreement scheme from PKC 2004, "*Information Processing Letters*,*Volume 91, Issue 4*,*31 August 2004*,*Pages 191-193*

[9] Kyungah Shim and Sungsik Woo. "Weakness in ID-based one round authenticated tripartite multiple-key agreement protocol with pairings, "*Applied Mathematics and Computation*, *Volume 166, Issue 3*, *26 July 2005*, *Pages 523-530*

[10] Claude Castelluccia, Nitesh Saxena and Jeong Hyun Yi. "Robust self-keying mobile ad hoc networks, "*Computer Networks*, *Volume 51, Issue 4*, *14 March 2007*, *Pages 1169-1182*

[11] Popescu, C. "A secure authenticated key agreement protocol, "Electrotechnical Conference, MELECON 2004. Proceedings of the 12th IEEE Mediterranean Volume 2, 12-15 May 2004 Page(s):783 - 786 Vol.2

[12] Youn-Ho Lee, Heeyoul Kim, Byungchun Chung, Jaewon Lee, Hyunsoo Yoon. “On-demand secure routing protocol for ad hoc network using ID based cryptosystem, “Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference on 27-29 Aug. 2003 Page(s):211 - 215 Digital Object Identifier 10.1109/PDCAT.2003.1236290

[13] Arjan Durresi, Vijay Bulusu, Vamsi Paruchuri and Leonard Barolli. “Secure emergency communication of cellular phones in ad hoc mode, ”*Ad Hoc Networks*, *Volume 5, Issue 1*, *January 2007*, *Pages 126-133*

[14] Hung-Yu Chien and Ru-Yu Lin. **“**Improved ID-based security framework for ad hoc network, ”*Ad Hoc Networks*, *In Press, Corrected Proof*, *Available online 31 August 2006*

[15] Capkun, S.; Buttyan, L.; Hubaux, J.-P.. “Self-organized public-key management for mobile ad hoc networks, ”Mobile Computing, IEEE Transactions on Volume 2, Issue 1, Jan.-March 2003 Page(s):52 - 64 Digital Object Identifier 10.1109/TMC.2003.1195151

[16] Hongmei Deng, Wei Li, Agrawal, D.P.. “Routing security in wireless ad hoc networks, ”Communications Magazine, IEEE Volume 40, Issue 10, Oct. 2002 Page(s):70 – 75 Digital Object Identifier 10.1109/MCOM.2002.1039859

[17] Ruppe R. Grisward S, Walsh P, Martin R. “Near term digital radio (NTDR) system, ”proceedings of the IEEE military communication conference, California, USA: 1997

[18] Zavgren J. “NTDR mobility management protocols and procedures, ”Proceeding of the IEEE military conferences, California, USA: 1997

[19] Chandra, J.; Singh, L.L. “**A** cluster based security model for mobile ad hoc networks, ” Personal_Wireless Communications, 2005. ICPWC 2005. 2005 IEEE International Conference on 23-25 Jan. 2005 Page(s):413 – 416 Digital Object Identifier 10.1109/ICPWC.2005.1431377

[20] Shu-Hwang Liaw, Pin-Chang Su, Henry Ker-Chang Chang, Erl-Huei Lu, Shun-Fu Pon, “Secured key exchange protocol in wireless mobile ad hoc networks,” Security Technology, 2005. CCST '05. 39th Annual 2005 International Carnahan Conference on 11-14 Oct. 2005 Page(s):171 - 173 Digital Object Identifier 10.1109/CCST.2005.1594839

[21] Shamir, “Identity based cryptosystems & signature schemes, ”Advances in Cryptology, CRYPTO’84, Lecture Notes-Computer Science, 1984, pp. 47–53