Paper

An identity-based broadcast encryption scheme for mobile ad hoc networks

Ching Yu Ng, Yi Mu, and Willy Susilo

Abstract— Dynamic ad hoc networks facilitate interconnections between mobile devices without the support of any network infrastructure. In this paper, we propose a secure identity-based ad hoc protocol for mobile devices to construct a group key for a setup of a secure communication network in an efficient way and support dynamic changing of network topology. Unlike group key management protocols proposed previously in the literature, mobile devices can use our protocol to construct the group key by observing the others' identity, like the MAC address, which distinguishes the device from the others. In contrast to other interactive protocols, we only need one broadcast to setup the group key and member removal is also highly efficient. Finally, we discuss the security issues and provide security proofs for our protocol.

Keywords— dynamic mobile ad hoc network, identity-based, non-interactive, secure communication protocol, group key management.

1. Introduction

Many modern computing environments involve dynamic ad hoc networks. Ad hoc networks facilitate interconnections between mobile devices without the need of support for any network infrastructure. When a mobile ad hoc network is formed in an open network environment, all intended and unintended devices can listen and observe the broadcasted communication since wireless signal cannot be hidden underground like wired networks. Security is becoming crucial in this environment. Therefore, the content of the communication must be protected so that only group members in the ad hoc group can obtain the information. Hence, a secure communication protocol and a robust group key management scheme are required to provide strong protection for group communication.

A naive approach to provide a secure communication in this environment is to share a common key, $\mathscr{K}$, among the group members, and this key will be used to encrypt and decrypt each message sent among them. The drawbacks of this approach are as follows:

- This protocol requires prior distribution of $\mathscr{K}$ before the network can be formed, which turns out to be inefficient when the key needs to be updated.
- This protocol does not support the dynamics of the group. When a group member decides to leave the group, the key $\mathscr{K}' \neq \mathscr{K}$ needs to be redistributed among the rest of the group members, which is inefficient.
- It is not possible to create a subgroup within the group, since everyone holds the same key.

Another important issue that needs to be considered in an ad hoc network is the trusted authority (TA). Group members should be able to form their network at anytime because of the mobility of ad hoc network. Hence, we cannot expect an online TA who can always redistribute a key $\mathscr{K}$ whenever needed. A common solution to avoid the need of TA is to employ Diffie-Hellman (DH) key exchange protocol where two parties can come up with the same key $\mathscr{K}$ by exchanging their own random secret interactively and use them to construct the key $\mathscr{K}$ [1]. Although this protocol can only supports two-party, some recent researches have shown that the extension to multiple-party protocol is possible [2–5]. The drawbacks of this approach are as follows:

- The group members must engage in an extensive protocol during the key setup phase. Usually, a leader or a root in the protocol is required to initialize the protocol.
- Depending on the number of group members, the total number of message exchanges can be large when a new key is required (e.g., when a new member joins).
- Due to the large number of message exchanges and the need of leader role, some of the group members may perform more calculations than others (the fairness problem) depending on the key management hierarchy (message exchange order of group members for setting up a new key) being adopted.

This is not encouraged in mobile ad hoc networks, since normally each group member is equipped with a device that has a very limited battery life. Having to perform a huge computation will simply mean that it will drain the battery of the device.

Conceptually, the idea proposed in [6, 7] by incorporating multilinear map may provide a good solution to this key setup problem. In their setting, each group member supplies their own random secret and broadcast it to other group members. Then they can construct a new group key in one round by using the multilinear map computation method. Unfortunately, at this stage, research has not successfully shown that the concrete construction of multilinear map exists. The existing map is the bilinear map in which Joux showed how to extend the DH key exchange protocol into a tripartite one round version using this map [8]. Barua, Dutta and Sharkar combine the bilinear map with the traditional DH key exchange protocol to construct a tree-based group key management protocol in [9]. Nevertheless, these protocols have not solved the fairness issue mentioned earlier, since some group members still need to perform more computations compared with others.

Having considered the main disadvantages of using key management protocols to setup the group key for mobile ad hoc group, we propose a new protocol which does not require the group members to perform any message exchanges during the generation process of group key. To achieve this goal, we incorporate the identity-based cryptosystem [10] with a bilinear map and pairing computation [11] to replace the contributory setup of a group key as seen in other literature [1–9]. Each group member is treated as a *broadcaster* in which he can select the designated receiver(s)(the whole ad hoc group or part of it) by himself and encrypt the message(key) that is only decipherable by them. Unlike previous protocols, our protocol avoids massive message exchanges for key setup that are sent between group members. Each group member is only required to broadcast one message to setup the group key, and hence, it is most efficient in terms of message exchanges and it provides fairness to every group members. They can also assure that only the designated receiver(s) can decrypt the message(key). We shall note that our protocol is perfect for a small group of people who would like to form a mobile ad hoc network. We would also like to point out that in a mobile ad hoc network, it is not common to have a very large group.

The rest of this paper is organized as follows. In the next section, we will provide some mathematical backgrounds that will be used to construct our scheme. In Section 3, we will provide our proposed scheme follow by a security analysis. Section 4 will conclude the paper.

2. Preliminaries

In this section, we describe the mathematical tools that will be used in our scheme.

2.1. Bilinear map and pairing

Let $\mathbb{G}_1$ be an additive group of points on an elliptic curve and $\mathbb{G}_2$ be a multiplicative group of a finite field. The order of both groups, $|\mathbb{G}_1| = |\mathbb{G}_2| = q$, where q is a large prime and the discrete logarithm problem in $\mathbb{Z}_q^*$ is intractable.

In the following, let P_1, P_2, P, $Q \in \mathbb{G}_1$ be the generators, and $a, b \in \mathbb{Z}_q^*$. A bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a function that:

- is *bilinear*:
  - $\hat{e}(aP, bQ) = \hat{e}(P,Q)^{ab}$,
  - $\hat{e}(P_1 + P_2, Q) = \hat{e}(P_1, Q)\hat{e}(P_2, Q)$;
- is *non-degenerate*:
  - for any generator $P \in \mathbb{G}_1$, $\hat{e}(P,P) \neq 1$;
- is *computable*:
  - there exists an efficient algorithm that can compute the map in polynomial time.

A pairing is an efficient algorithm to compute the mapping between $\mathbb{G}_1$ and $\mathbb{G}_2$ for all generators in $\mathbb{G}_1$. Modified Weil pairing is one of the pairings that has been used frequently in recent cryptographic applications [8, 11–13].

2.2. Identity-based cryptosystem

In an identity-based cryptosystem (or ID-based, for short), users are not bound to certificates and no online trusted authorities are required to verify the validity of their certificate. They are bound to their unique identifier (ID) and their private key is obtained from a key generation center (KGC) while their public key is determined with their ID. The center, KGC, can go off-line after the setup of common system parameters and the distribution of keys to users. Later on, one of the two users *Alice* and *Bob*, say *Alice*, wants to send a message to *Bob*, she can encrypt the message using the public key computed from the ID (name, e-mail address, etc., as long as it can be used to uniquely identify the user) of *Bob*. The encrypted message can only be decrypted by *Bob* using his private key previously obtained from the KGC.

Currently the well known ID-based encryption scheme [11] that incorporates the bilinear map and pairing is as follows. The ID-based cryptosystem proposed by Boneh and Franklin:

- `Setup.` KGC generates two groups $(\mathbb{G}_1, +)$ the additive group and $(\mathbb{G}_2, \cdot)$ the multiplicative group both with prime order q together with a bilinear map $\hat{e} : (\mathbb{G}_1, +)^2 \to (\mathbb{G}_2, \cdot)$. It also selects an arbitrary generator $P \in \mathbb{G}_1$, then picks $s \in \mathbb{Z}_q^*$ randomly and sets $P_{pub} = sP$ as its public key, where s denotes the master secret key. Finally, two cryptographically strong hash functions are selected: $F : \{0,1\}^* \to \mathbb{G}_1$, $H : \mathbb{G}_2 \to \{0,1\}^n$, where n denotes the size of the plaintext message space. The system parameters and their descriptions are made public in a tuple $\{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, q, n, P, P_{pub}, F, H\}$ while the master secret key s is kept secret.
- `Extract.` After performing physical identification of a user, say *Alice*, and making sure the uniqueness of her ID_{Alice}, KGC generates her secret key as follows. It computes $Q_{\mathrm{ID}_{Alice}} = F(\mathrm{ID}_{Alice})$ and sets $S_{\mathrm{ID}_{Alice}} = sQ_{\mathrm{ID}_{Alice}}$. $S_{\mathrm{ID}_{Alice}}$ is given to Alice as her secret key. It is the same for *Bob* where his identity is ID_{Bob} and his secret key $S_{\mathrm{ID}_{Bob}} = sQ_{\mathrm{ID}_{Bob}}$.
- `Encrypt.` To send an encrypted message to *Bob*, *Alice* first obtains the system parameters and uses *Bob*'s identity to compute $Q_{\mathrm{ID}_{Bob}} = F(\mathrm{ID}_{Bob})$. Then,

to encrypt a message $m \in \{0,1\}^n$, *Alice* picks $r \in Z_q^*$ randomly and computes rP and $g_{\mathrm{ID}_{Bob}} = \hat{e}(Q_{\mathrm{ID}_{Bob}}, P_{pub})^r$. The ciphertext is $C = (rP, m \oplus H(g_{\mathrm{ID}_{Bob}}))$.

- `Decrypt.` Let $C = (U,V)$ be the ciphertext received by *Bob*. To decrypt C using his private key $S_{\mathrm{ID}_{Bob}}$, he computes $g_{\mathrm{ID}_{Bob}} = \hat{e}(S_{\mathrm{ID}_{Bob}}, U) = \hat{e}(sQ_{\mathrm{ID}_{Bob}}, rP) = \hat{e}(Q_{\mathrm{ID}_{Bob}}, sP)^r = \hat{e}(Q_{\mathrm{ID}_{Bob}}, P_{pub})^r$. The message is $m = V \oplus H(g_{\mathrm{ID}_{Bob}})$.

2.3. Single encryption and multiple decryptions

In [14], a new public key based cryptosystem was proposed where there is one public encryption key and multiple decryption keys. It works by considering the polynomial function:

$$f(x) = \prod_{i=1}^{n}(x - x_i) \equiv \sum_{i=0}^{n} a_i x^i ,$$

where a_i denotes the coefficient corresponding to x^i after the expansion of $f(x)$, i.e., $a_0 = \prod_{i=1}^{n}(-x_i)$, $a_1 = \sum_{i=1}^{n}\prod_{j \neq i}^{n}(-x_j)$, …, $a_{n-1} = \sum_{i=1}^{n}(-x_i)$, $a_n = 1$ (note that $f(x_i), 1 \leq i \leq n$ is equal to 0).

Under this construction, any generator $g \in \mathbb{Z}_q^*$ rises to power $f(x)$, i.e., $g^{f(x)}$ *mod* q (q is a large prime) will give the result equals to 1 for $x = x_i$, $i = 1 \ldots n$. (We assume the calculations in this paper are under modulo q and will omit the (mod q) notation in the rest of the paper where it is obvious from the context).

With this property, we let $x_1, x_2, \ldots, x_n$ be the private decryption keys of user $\mathscr{U}_1, \mathscr{U}_2, \ldots, \mathscr{U}_n$, respectively, and $\{g_0, g_1, g_2, \ldots, g_n\} = \{g^{a_0}, g^{a_1}, g^{a_2}, \ldots, g^{a_n}\}$ be the public encryption key tuple. Then a message m can be encrypted as $m \cdot g_0^r$ by choosing a random number $r \in \mathbb{Z}_q^*$ and sending $C = \{m \cdot g_0^r, g_1^r, g_2^r, \ldots, g_n^r\}$ as the ciphertext. The encrypted message can be decrypted by any one of the users by using his own private key x_i to calculate:

$$\begin{aligned} m \cdot g_0^r \cdot \prod_{j=1}^{n} g_j^{rx_i^j} &= m \cdot \prod_{j=0}^{n} g_j^{rx_i^j} \\ &= m \cdot g^{\sum_{j=0}^{n} a_j x_i^j \cdot r} \\ &= m \cdot g^{f(x_i) \cdot r} \\ &= m \cdot 1^r = m\,. \end{aligned}$$

3. Our proposed scheme

3.1. Security model

3.1.1. System model

In our paper, we consider the situation where a group of users are selected as a subset from the user set $\mathscr{U} = \{\mathscr{U}_1, \mathscr{U}_2, \ldots, \mathscr{U}_k\}$ who would like to form a mobile ad hoc network by using their wireless devices. There exists a key generation center that sets up system parameters, generates and distributes private keys as described in Subsection 2.2. The KGC will accept any person's ID. Upon successful verification of the ID, KGC generates the private key associated with the ID provided. The n users in set $\mathscr{U}$ are those who have contacted the KGC to obtain their private key and have their ID being known by each user within the set. We note that the KGC's role is only to provide the necessary system parameters and distribute each user his private key, hence the KGC is not necessary to keep online after the completion of these procedures and is not required anymore by the users who want to setup a mobile ad hoc network, which fulfill the infrastructureless requirement of dynamic ad hoc networks.

3.1.2. Adversary model

We assume there exists an adversary $\mathscr{A} \notin \mathscr{U}$. All messages available in the network are also available to $\mathscr{A}$. This includes all the messages sent by any set of users $\subset \mathscr{U}$ that wishes to create a mobile ad hoc network. The main goal of $\mathscr{A}$ is to deviate the protocol by decrypting any messages sent within the network intended to any set of users $\subset \mathscr{U}$ but not him. $\mathscr{A}$ is considered to be successful if he wins in the following experiment.

Indistinguishability of encryptions under adaptive chosen plaintext attack (IND-CPA):

1. $\mathscr{A}$ picks a group of user IDs to be attacked and tells the challenger $\mathscr{C}$.
2. $\mathscr{C}$ runs the KGC's `Setup` algorithm to generate the necessary system parameters and his private key. The parameters are given to $\mathscr{A}$ while $\mathscr{C}$ keeps his private key secret.
3. $\mathscr{A}$ can query $\mathscr{C}$ up to q_H hash queries on any ID he wants and up to q_E extraction queries on any ID not equal to the IDs he picked in Step 1. $\mathscr{C}$ will reply with proper hash results on those IDs and runs the `Extract` algorithm to reply $\mathscr{A}$ the private keys he needs.
4. Meanwhile, $\mathscr{A}$ will select two messages $\{m_0, m_1\}$ and gives them to $\mathscr{C}$. $\mathscr{C}$ will then pick one of them randomly by flipping a fair coin to obtain $b \in \{0,1\}$. $\mathscr{C}$ runs the `Encrypt` algorithm on m_b using the IDs picked by $\mathscr{A}$ in Step 1 to get the ciphertext C and gives it back to $\mathscr{A}$ without letting him knows which message is being picked.
5. $\mathscr{A}$ can keep on querying $\mathscr{C}$ the hash or extract values if the total numbers of queries have not exceeded q_H and q_E.
6. Eventually $\mathscr{A}$ will make a guess $b' \in \{0,1\}$ on which message was being picked by $\mathscr{C}$.

If $\mathscr{A}$ somehow managed to guess the correct answer (i.e., $b' = b$) in the experiment on the protocol above then $\mathscr{A}$ wins the experiment and the protocol is not secure. We say that $\mathscr{A}$ has a guessing advantage ε that the probability of $\mathscr{A}$ winning the experiment is $P[b' = b] = \frac{1}{2} + \varepsilon$.

A protocol is said to be secure against IND-CPA if there exist no adversaries with advantage ε that can win the experiment within $q_H + q_E$ queries, in other words ε is negligible.

3.1.3. Security properties

Our protocol is secure against IND-CPA, which means no adversaries can decrypt the messages sent within the network not intended to them. If we consider the messages as some group keys in different sessions, we obtain a secure group key management method with the following properties:

1. *Group key secrecy*. The group key is computationally infeasible to compute.
2. *Known session key secrecy*. Even if one or more previous group session keys are exposed, the current or future session keys are still secure.
3. *Forward secrecy*. If one or more group members' private key are exposed, only the previous session keys are revealed, the current or future session keys are still secure.
4. *Key control secrecy*. The group key is randomly constructed and can not be predicted.

3.2. System construction

Our protocol incorporates the ID-based cryptosystem [10] and its construction using a bilinear map and pairing [11] together with the single encryption and multiple decryption method [14] to create a secure and efficient communication protocol for mobile ad hoc network.

For simplicity, we assume that each of the users $\mathcal{U}_i \in \mathcal{U}$ has contacted the KGC to obtain their ID-based private key $S_{\mathrm{ID}_i} = sF(\mathrm{ID}_i)$. The system parameters $\{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, q, n, P, P_{pub}, F, H\}$ are publicly known and each user's ID is known within the user group $\mathcal{U}$. These procedures can be done at anytime before the network is formed.

Let there be a set of users $\mathcal{U}$ and a subset $\mathcal{U}' \subset \mathcal{U}$ of size n wanting to form a mobile ad hoc group. Let $\mathcal{U}_s$ denote a group member who joins $\mathcal{U}'$ and wants to broadcast a message (or session key) to the rest of group. We refer to $\mathcal{U}' \cup \{\mathcal{U}_s\}$ as the current group. Our protocol works as follows:

- `Setup.` Given the system parameters as described above, each of the group members in the current group will perform the following calculations:
  - Select a random number $r \in \mathbb{Z}_q^*$, set $R = rP$.
  - For n other group members in the current group, calculate $e_i = H(\hat{e}(P_{pub}, rF(\mathrm{ID}_i))), i = 1 \ldots n$.
  - Use the e_i values to construct the polynomial function $f(e) = \prod_{i=1}^{n}(e - e_i) = \sum_{i=0}^{n} a_i e^i$.
  - Compute $\{g_0, g_1, \ldots, g_n\} = \{g^{a_0}, g^{a_1}, ..., g^{a_n}\}$.

  After this phase, each group member is equipped with a different encryption key tuple $\{g_0, g_1, \ldots, g_n, R\}$. This tuple will not change throughout the whole session as long as the group topology does not change and none of the private keys of current group members has been exposed.
- `Encrypt.` Let m be the message (or new session key). $\mathcal{U}_s$ will perform the following calculations to encrypt m and broadcast it to the rest of current group members:
  - Select two random numbers $k_1, k_2 \in \mathbb{Z}_q^*$.
  - Raise each component in the encryption tuple to power k_2, i.e., calculate $\{g_0^{k_2}, g_1^{k_2}, \ldots, g_n^{k_2}\}$.
  - Encrypt the message m as $Z = m \oplus k_1$ and compute $A = k_1 \cdot g_0^{k_2}$.
  - Broadcast $C = \{Z, A, g_1^{k_2}, \ldots, g_n^{k_2}, R\}$.
- `Decrypt.` Upon receiving the broadcast message from $\mathcal{U}_s$, each user in current group can decrypt the message with the following calculations:
  - Compute $e_i = H(\hat{e}(R, S_{\mathrm{ID}_i}))$ using his private key S_{ID_i}.
  - Compute $k = A \cdot \prod_{j=1}^{n} g_j^{k_2 \cdot e_i^j}$.
  - $m = Z \oplus k$.

  Note that the computation $H(\hat{e}(R, S_{\mathrm{ID}_i})) = H(\hat{e}(rP, sF(\mathrm{ID}_i))) = H(\hat{e}(sP, rF(\mathrm{ID}_i))) = H(\hat{e}(P_{pub}, rF(\mathrm{ID}_i)))$ and $A \cdot \prod_{j=1}^{n} g_j^{k_2 \cdot e_i^j} = k_1 \cdot g_0^{k_2} \cdot \prod_{j=1}^{n} g_j^{k_2 \cdot e_i^j} = k_1 \cdot \prod_{j=0}^{n} g_j^{k_2 \cdot e_i^j} = k_1 \cdot g^{k_2 \cdot \sum_{j=0}^{n} a_j e_i^j} = k_1 \cdot g^{f(e_i) \cdot k_2} = k_1 \cdot 1^{k_2} = k_1$ and hence message m can be decrypted correctly.

As the mobile ad hoc user group is dynamic, whenever there is a join or leave of group member, simply add or exclude that member's ID during execution of `Setup` to obtain a new encryption key tuple. Note that the pairing computation for the e_i values can be reused if the new join member is a returning old member, only the encryption key tuple is needed to recalculate. This can save a lot of computation as pairing computations are expensive.

3.3. Security analysis

To prove our protocol is secure against IND-CPA, we first assume that there exists an adversary $\mathcal{A}$ that wins in the indistinguishability experiment described in Subsection 3.1. Then we create a simulator $\mathcal{B}$ that intercepts all the communication between $\mathcal{A}$ and the challenger $\mathcal{C}$, $\mathcal{B}$ is able to modify and forward the communication contents and is transparent to $\mathcal{A}$ and $\mathcal{C}$ making $\mathcal{A}$ see no difference between the simulator $\mathcal{B}$ or the real challenger $\mathcal{C}$. The goal of $\mathcal{B}$ is to make use of $\mathcal{A}$ to solve a cryptographic hard problem. Since the hard problem is known to be unsolvable in polynomial time, the assumption that $\mathcal{A}$ exists leads to a contradiction and hence our protocol is secure. We first review the cryptographic hard problem that we will use in the proof:

Bilinear decisional Diffie-Hellman problem (BDDHP): given an instance (P, aP, bP, cP, θ), where P is a generator $\in \mathbb{G}_1$, $a, b, c \in \mathbb{Z}_q^*$ are chosen uniformly at random and $\theta \in \mathbb{G}_2$. The goal for an attacker is to decide whether $\theta = \hat{e}(P,P)^{abc}$ within polynomial time. BDDHP is hard with an assumption that there exists no polynomial time algorithm for any attacker to solve BDDHP, such that the probability of success is non-negligible.
We now construct the simulator $\mathscr{B}$ as follows (note that $\mathscr{C}$ can be omitted here as $\mathscr{B}$ has simulated it):

1. $\mathscr{B}$ is given an instance (P, aP, bP, cP, θ) of BDDHP as described above.
2. $\mathscr{A}$ picks a group of user IDs to be attacked and tells $\mathscr{B}$.
3. $\mathscr{B}$ runs the KGC's `Setup` algorithm to generate the necessary system parameters. The parameters $\{\mathbb{G}_1, \mathbb{G}_2, \hat{e}, q, n, P, P_{pub}, F, H\}$ are modified by $\mathscr{B}$ by setting P_{pub} to cP before giving to $\mathscr{A}$.
4. Whenever $\mathscr{A}$ issues a hash query on ID_i, $\mathscr{B}$ replies with his modified hash function $F^{'}$ using the following method:
   - $\mathscr{B}$ maintains a query list $F_{list} : \{\mathrm{ID}_i, r_i, F^{'}(\mathrm{ID}_i)\}$. When the query on ID_i has been asked before, $\mathscr{B}$ looks up F_{list} to find the matching ID_i and replies with $F^{'}(\mathrm{ID}_i)$.
   - If the query on ID_i has not been asked before, $\mathscr{B}$ first selects a random number $r_i \in \mathbb{Z}_q^*$ and further checks that if ID_i is one of the IDs picked by $\mathscr{A}$ in Step 2. If it is, $\mathscr{B}$ sets $F^{'}(\mathrm{ID}_i) = r_iP + bP$, else $\mathscr{B}$ sets $F^{'}(\mathrm{ID}_i) = r_iP$.
   - $\mathscr{B}$ updates F_{list} with the new entry and replies $\mathscr{A}$ $F^{'}(\mathrm{ID}_i)$.
5. Whenever $\mathscr{A}$ issues an extraction query on ID_i, $\mathscr{B}$ replies with his modified `Extract` algorithm using the following method:
   - If the query on ID_i exists on F_{list}, $\mathscr{B}$ takes the $F^{'}(\mathrm{ID}_i)$ value and replies with $S_{\mathrm{ID}_i} = r_icP$.
   - Otherwise $\mathscr{B}$ follows the hash query replying method to create a new entry for ID_i first then replies with $S_{\mathrm{ID}_i} = r_icP$.
   - Note that $\mathscr{A}$ is not allowed to query on the IDs picked in Step 2. For extraction values, hence $F^{'}(\mathrm{ID}_i)$ is always in the form r_iP in F_{list} and $r_icP = cr_iP = cF^{'}(\mathrm{ID}_i)$, which is a perfect simulation of extraction value (since P_{pub} has been replaced by cP).
6. At the time $\mathscr{A}$ provides two messages $\{m_0, m_1\}$, $\mathscr{B}$ picks one of them randomly to obtain $b \in \{0,1\}$ and looks up F_{list} for the r_i values on the IDs picked by $\mathscr{A}$ in Step 2. $\mathscr{B}$ runs the `Setup` algorithm of our protocol to calculate the e_i values for these IDs by setting $R = aP$ and $e_i = H(\theta \cdot \hat{e}(R, P_{pub})^{r_i})$. With these e_i values, $\mathscr{B}$ runs the `Encrypt` algorithm of our protocol to encrypt the selected message m_b and sends $\mathscr{A}$ the ciphertext.
7. $\mathscr{A}$ can keep on querying if the total numbers of queries have not exceeded q_H and q_E.
8. Eventually $\mathscr{A}$ will make a guess $b^{'} \in \{0,1\}$ on which message was being picked by $\mathscr{B}$.

If the guess from $\mathscr{A}$ is correct (i.e., $b^{'} = b$), then $\mathscr{B}$ knows that $\theta = \hat{e}(P,P)^{abc}$, otherwise $\mathscr{B}$ knows that $\theta \neq \hat{e}(P,P)^{abc}$. This is because the e_i values computed by $\mathscr{B}$ are able to construct a valid ciphertext on m_b.
Note that if $\mathscr{A}$ guesses it correctly, then $e_i = H(\theta \cdot \hat{e}(R, P_{pub})^{r_i}) = H(\hat{e}(P,P)^{abc} \cdot \hat{e}(aP, cP)^{r_i}) = H(\hat{e}(aP, bcP + r_icP)) = H(\hat{e}(R, cF^{'}(\mathrm{ID}_i))) = H(\hat{e}(R, S_{\mathrm{ID}_i}))$. For the above construction of simulator $\mathscr{B}$, we successfully show that $\mathscr{B}$ can solve the BDDHP using the guess provided by $\mathscr{A}$, which leads to a contradiction that BDDHP is unsolvable. Hence the assumption that $\mathscr{A}$ exists is invalid and our protocol is secure. The other security properties mentioned in Subsection 3.1 are straight froward: our IND-CPA protocol implies the *group key secrecy*. With two random values k_1, k_2 selected every time the new session key is broadcasted, we ensure the *known session key secrecy* and *key control secrecy*. *Forward secrecy* can be provided if the group member who has lost his private key is promptly informed to the group and the other group members can simply exclude his ID from the `Setup` phase.

4. Conclusion

We proposed a new secure communication protocol for mobile ad hoc networks. The protocol offers an efficient setup algorithm, together with an efficient protocol for encrypting and decrypting the message among the ad hoc group. Member joining or removal is also simple and quick. With only one broadcast message, each member in the ad hoc group can obtain a new group session key. The use of ID-based cryptosystem provides an easy way to setup our protocol and to include or exclude designated receivers without interrupting the other group members, which can be an advantage for greater flexibility.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography", *IEEE Trans. Inform. Theory*, vol. IT-22, no. 6, pp. 644–654, 1976.

[2] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman key distribution extended to group communication", in *ACM Conf. Comput. Commun. Secur.*, New Delhi, India, 1996, pp. 31–37.

[3] G. Ateniese, M. Steiner, and G. Tsudik, "New multiparty authentication services and key agreement protocols", *IEEE J. Selec. Areas Commun.*, vol. 18, no. 4, pp. 628–639, 2000.

[4] E. R. Anton and O. C. M. B. Duarte, "Group key establishment in wireless ad hoc networks", in *Worksh. QoS Mob.*, Angra dos Reis, Brazil, 2002.

[5] N. Asokan and P. Ginzboorg, “Key-agreement in ad hoc networks”, *Comput. Commun.*, vol. 23, no. 17, pp. 1627–1637, 2000.

[6] D. Boneh and A. Silverberg, “Applications of multilinear forms to cryptography”, *Cryptol. ePrint Arch.*, Rep. 2002/080, 2002.

[7] H. K. Lee, H. S. Lee, and Y. R. Lee, “Multi-party authenticated key agreement protocols from multilinear forms”, *Cryptol. ePrint Arch.*, Rep. 2002/166, 2002.

[8] A. Joux, “A one round protocol for tripartite Diffie-Hellman”, in *Algorithmic Number Theory, 4th International Symposium ANTS-IV, Lecture Notes in Computer Science*. Leiden: Springer, 2000, vol. 1838, pp. 385–394.

[9] R. Barua, R. Dutta, and P. Sarkar, “An *n*-party key agreement scheme using bilinear map”, *Cryptol. ePrint Arch.*, Rep. 2003/062, 2003.

[10] A. Shamir, “Identity-based cryptosystems and signature schemes”, in *Advances in Cryptology: Proceedings of CRYPTO 84, Lecture Notes in Computer Science*. Santa Barbara: Springer, 1984, vol. 196, pp. 47–53.

[11] D. Boneh and M. Franklin, “Identity based encryption from the weil pairing”, in *Advances in Cryptology: Proceedings of CRYPTO’01, Lecture Notes in Computer Science*. Santa Barbara: Springer, 2001, vol. 2139, pp. 213–229.

[12] N. P. Smart, “An identity based authenticated key agreement protocol based on the weil pairing”, *Cryptol. ePrint Arch.*, Rep. 2001/111, 2001.

[13] D. Nalla, “ID-based tripartite key agreement with signatures”, *Cryptol. ePrint Arch.*, Rep. 2003/144, 2003.

[14] Y. Mu, V. Varadharajan, and K. Q. Nguyen, “Delegated decryption”, in *Proceedings of Cryptography and Coding, Lecture Notes in Computer Science*. Cirencester: Springer, 1999, vol. 1746, pp. 258–269.

Ching Yu Ng graduated with the M.Sc. degree in computer science from the School of Engineering at the Hong Kong University of Science and Technology in 2003. He continues his postgraduate studies in computer security as a research student in the Center for Information Security at the University of Wollongong in Australia. His research topics include wireless security, group key agreement protocols for dynamic ad hoc network, broadcast encryption schemes and digital signatures.
e-mail: cyn27@uow.edu.au
School of Information Technology and Computer Science
Faculty of Informatics
University of Wollongong
Wollongong, NSW 2522, Australia

Yi Mu received his Ph.D. from the Australian National University in 1994. He was a Lecturer in the School of Computing and IT at the University of Western Sydney and a Senior Lecturer in the Department of Computing at Macquarie University. He currently is an Associate Professor in the Information Technology and Computer Science, University of Wollongong. His current research interests include network security, computer security, and cryptography. He is a Member of the Editorial Board of “Journal of Universal Computer Science”, a Senior Member of the IEEE, and a Member of the IACR.
e-mail: ymu@uow.edu.au
School of Information Technology and Computer Science
Faculty of Informatics
University of Wollongong
Wollongong, NSW 2522, Australia

Willy Susilo received a Ph.D. in computer science from University of Wollongong, Australia. He is currently a Senior Lecturer at the School of Information Technology and Computer Science of the University of Wollongong. He is the Coordinator of Network Security Research Laboratory at the University of Wollongong. His research interests include cryptography, information security, computer security and network security. His main contribution is in the area of digital signature schemes, in particular fail-stop signature schemes and short signature schemes. He has served as a program committee member in a number of international conferences. He is a Member of the IACR.
e-mail: wsusilo@uow.edu.au
School of Information Technology and Computer Science
Faculty of Informatics
University of Wollongong
Wollongong, NSW 2522, Australia