ID-Based Key Agreement with Anonymity for Ad Hoc Networks

Hung-Yu Chien

Department of Information Management, National Chi Nan University, Taiwan, R.O.C.
redfish6@ms45.hinet.net

Abstract. Security support is a must for ad hoc networks. However, existing key agreement schemes for ad hoc networks ignore the issue of entity anonymity. Without anonymity, the adversary can easily identify and track specific entities in the communications. Not only entities' movement information is valuable to the adversary but also the adversary can launch heavy attacks on those important nodes, based on the information. This paper proposes an ID-based n-party ($n \geq 2$) key agreement scheme that preserves entity anonymity from outsiders. The scheme is efficient and very suitable for the structure-free mobile ad hoc networks. The security of the schemes is proved in a modified Bellare-Rogaway model.

Keywords: ad hoc networks, bilinear pairing, identity-based cryptosystem, key agreement, anonymity.

1 Introduction

Ad hoc networks that support self-configurable and autonomous communications are regarded as ideal technologies for creating instant communication networks for civilian and military applications. Depending on the applications and the environments, different ad hoc networks may require different degree of support infrastructure. Asokan and Ginzboorg 0 classified three types of support infrastructures for ad hoc networks. The first type is the routing infrastructure in the form of fixed routers and stable links. The second type is the server infrastructure of on-line servers that provide various services such as name service, directory services, certificate look-up services, etc. The third type is the organizational and administrative support such as registration of users, issuing of certificates, and cross-certification agreements between different user domains. Regarding ad hoc networks, some other features are worth further discussions. First, ad hoc networks are dynamic. It means that nodes in an ad hoc network will move dynamically and some nodes might own poor connectivity with neighbors (or might own rich connectivity for only a short time). So the algorithms designed for ad hoc networks should take these features into account. Secondly, the locations and movements of specific nodes could be valuable information to the adversary. For example, in military applications or some commercial applications, some nodes might play important (or even vital) roles in the communications. One example is the commander in military operations or in crisis management operations, and another example is the server in a business meeting outside. Therefore, exposure of the identities and the locations of nodes could endanger the whole system.

Regarding the security requirements of ad hoc networks, secure key agreement schemes and efficient group key management are two of the most important mechanisms to build a secure network [15]. However, existing key agreement schemes or key management schemes like [10, 12-14, 16-18] for ad hoc networks all ignore the anonymity issue, and many of them assume the on-line certificate servers to support Public Key Infra-structure (PKI) service. Even though it is feasible to support on-line PKI services via distributed mechanism [16-17], the cost to pay is very high which limits their applications, when we consider the dynamic property, the poor connectivity property and the possible resource limitation on these mobile nodes.

Conventionally, the certificate-based public key infrastructure requires an entity to access and verify certificates before using the public keys. It is costly. To get rid of the weaknesses of certificate-based public key infrastructure, Shamir 0 first proposed the first IDentity-based (ID-based) cryptosystem, where an entity's identification is taken as its public key, and, therefore, there is no requirement to securely maintain and verify the public key before using it. An essential requirement of ID-based schemes is that entities' identifications are well known. Fortunately, many ad hoc applications meet this requirement. For example, in military, campus, emergency operations and commercial environments, some kind of identification mechanisms (like social security number, e-mail address, IP address, or codes) have been widely used to uniquely identify the entities. These features make the ID-based cryptosystems very suitable for many ad hoc networks.

In this paper, we focus on the issues of key agreement schemes with anonymity for ad hoc networks; and the issue of secure routing and secure channels for multi-hop link are beyond the scope of this paper; of course, the mechanism proposed in this paper can be used as building blocks for secure routing and secure channel over multi-hop links. We also assume that it is not easy to compromise the entities; therefore, those compromise-prone devices like sensors and RFIDs are excluded in this work. We propose ID-based key agreement schemes with anonymity for ad hoc networks with single Key Generator Center (KGC), and the possible extension for multiple KGCs [21] or for compromise-prone devices is our future work. The benefits of the schemes include: (1) there is no requirement of on-line server support; (2) the schemes preserve anonymity so that an outsider cannot identify or track the communicating parties; (3) they are efficient and adaptive so that they meet the poor connectivity property and the resource-limited property. Regarding the security, we consider the indistinguishability of the session key and the anonymity of the communicating parties in our modified Bellare-Rogaway model that considers the anonymity property and the tripartite case. The rest of this paper is organized as follows. Section 2 discusses the related works. Section 3 introduces some definitions useful to understand the design of the schemes. Section 4 presents our two-party key agreement, tripartite key agreement and group key agreement with anonymity. Finally, our conclusions are drawn in Section 5.

Related Works. Related works includes the key agreement schemes for ad hoc networks, the key management schemes for ad hoc networks and the pairing-based key agreement schemes. They are briefly discussed as follows.

One interesting key agreement scheme for ad hoc networks is Asokan-Ginzboorg' *location*-based key agreement scheme [13], where the people physically present in some place know and trust one another physically, but they do not have any a priori means of digitally identifying and authenticating one another, such as shared keys, public key certificate, or on-line trusted servers. Their scheme is so called "location-based key agreement", because only the people locating at the same room (or place) who can see each other can set up a shared password physically and establish the secure communication accordingly.

Contrary to the above special type of ad hoc networks, most ad hoc networks are like those cases where the entities (could be people, devices, and mobile nodes) knowing the identities of other entities instead of "location" want to set up secure communications. Kaya et al.'s multicast scheme [14] attaches joining nodes to the best closest neighbor therefore reducing the cost of request broadcast and reducing the communication and computation cost incurred by the source. The protocol strongly requires the support of on-line certificate authorities, which makes it not suitable for most structure-free ad hoc networks and resource constrained nodes. Rhee et al.'s group key management architecture [10] for MANETs uses the Implicitly Certified Public Keys (ICPK) to eliminate the requirement of on-line server. However, the ICPK exchange for computing a pair-wise key is costly, and the cost of re-keying the group key is $O(\log_2 n)$. Instead of ICPK, Bohio-Miri [12] and Chien-Lin [18], based on ID-based cryptosystem, had proposed the security frameworks for ad hoc networks to get rid of the requirement of on-line servers. However, none of the above schemes considered t the anonymity issue.

Our proposed anonymous key agreement schemes are based on ID-based cryptosystems from pairing. However, none of the previous pairing-based key agreement schemes like [1, 9, 11, 18, 20-23] considered the anonymity property, and it seems difficult to achieve the anonymity property by simply extending the previous works, because all the previous key agreement schemes need to exchange entities' identities when they try to establish session keys.

The key management schemes like [16-17], instead of the key agreement issues, focused on the key management issue: how to build the Certificate Authority (CA) [16] service for conventional PKI or the Key Generator Center (KGC) service [17] for ID-based cryptosystem in ad hoc networks. The key management scheme [17] is complementary to our work, and the idea of bootstrapping the KGC can be applied on our schemes for those environments where the entities do not get the public parameters and their private keys from the KGC before the ad hoc network is formed.

2 Preliminaries

We propose our ID-based key agreement schemes from bilinear pairings [6, 8]. In this section, we briefly describe the basic definitions and properties of the bilinear pairing and the assumptions.

2.1 Bilinear Pairing

Let G_1 and G_2 denote two groups of prime order q, where G_1 is an additive group that consists of points on an elliptic curve, and G_2 is a multiplicative group of a finite field. A bilinear pairing is a computable bilinear map between two groups. Two pairings have been studied for cryptographic use. They are the (modified) Weil pairing $\hat{e}:G_1 \times G_1 \rightarrow G_2$ [6] and the (modified) Tate pairing $\hat{t}:G_1 \times G_1 \rightarrow G_2$ [8]. For the purposes of this paper, we let e denote a general bilinear map, i.e., $e:G_1 \times G_1 \rightarrow G_2$, which can be either the modified Weil pairing or the modified Tate pairing, and has the following three properties:

(1) Bilinear: if $P,Q,R \in G_1$ and $a \in Z_q^*$, $e(P+Q,R)=e(P,R)e(Q,R)$, $e(P,Q+R)=$ $e(P,Q)e(P,R)$, and $e(aP,Q)=e(P,aQ)=e(P,Q)^a$.

(2) Non-degenerate: There exists $P,Q \in G_1$ such that $e(P,Q) \neq 1$.

(3) Computable: There exist efficient algorithms to compute $e(P,Q)$ for all $P,Q \in G_1$.

Definition 1. The bilinear Diffie-Hellman problem (BDHP) for a bilinear pairing $e:G_1 \times G_1 \rightarrow G_2$ is defined as follows: Given $P,aP,bP,cP \in G_1$, where a,b,c are random numbers from Z_q^*, compute $e(P,P)^{abc} \in G_2$.

Definition 2. The computational Diffie-Hellman problem (CDHP) is defined as follows: Given $P,aP,bP \in G_1$, where a and b are random numbers form Z_q^*, compute $abP \in G_1$.

Definition 3. The decision bilinear Diffie-Hellman problem (DBDH) for a bilinear pairing $e:G_1 \times G_1 \rightarrow G_2$ is defined as follows: Define two probability distributions of tuples of seven elements, $Q_0=\{\left\langle G_1,G_2,P,aP,bP,cP,e(P,P)^{abc}\right\rangle : a,b,c \in_R Z_q\}$ and $Q_1=\{\left\langle G_1,G_2,P,aP,bP,cP,e(P,P)^{d}\right\rangle : a,b,c,d \in_R Z_q\}$. Then, given the tuple $\left\langle G_1,G_2,P,P_{\mathrm{A}},P_B,P_C,K\right\rangle$, decide whether the tuple is from Q_0 or from Q_1.

Definition 4. The Inverse Computational Diffie-Hellman Problem (Inv-CDHP): given P, aP, to compute $a^{-1}P$.

In order to prove the security of our schemes, we define a new problem and prove it is equivalent to other hard problems as follows. To our best knowledge, we do not know there is any formulation of the BoIDHP before, and we refer it the name BoIDHP to differentiate it from the conventional BDHP.

Definition 5. The Bilinear one Inverse Diffie-Hellman Problem (BoIDHP): given bilinear pairing $e:G_1 \times G_1 \rightarrow G_2$, $P,aP,bP,cP \in G_1$, where a,b,c are random numbers from Z_q^*, compute $e(P,P)^{abc^{-1}} \in G_2$.

CDHP, BDHP, DBDH, Inv-CDHP assumptions: It is commonly believed that there is no polynomial time algorithm to solve BDHP, CDHP, Inv-CDHP or DBDH with non-negligible probability [1, 6, 7, 19].

Theorem 1. BoICDHP and BDHP are polynomial time equivalent.

Proof: we give a simple proof as follows.

(i) we first prove BDHP $\Rightarrow$ BoIDHP. Given P, aP, bP, cP, we set the input of BDHP as follows. $Q = cP, Q_1 = aP = ac^{-1}Q, Q_2 = bP = bc^{-1}Q, Q_3 = P = c^{-1}Q$, then BDHP outputs $e(Q,Q)^{ac^{-1}bc^{-1}c^{-1}} = e(P,P)^{abc^{-1}}$.
(ii) BoIDHP $\Rightarrow$ BDHP. Given P, aP, bP, cP, we set the input of BoIDHP as follows. $Q = cP, Q_1 = aP = ac^{-1}Q, Q_2 = bP = bc^{-1}Q, Q_3 = P = c^{-1}Q$, then BoIDHP outputs $e(Q,Q)^{ac^{-1}bc^{-1}c}$ = $e(P,P)^{abc}$. □

2.2 Parameters for ID-Based Cryptosystems from Pairing

Let G_1 and G_2 denote two groups of prime order q, where G_1 is a group on the elliptic curves. q is a prime which is large enough to make solving discrete logarithm problem in G_1 and G_2 infeasible. Let *P* is a generator of G_1, and the *MapToPoint* function 0 encodes the identity of a user to a point in the group G_1. Let us denote such a function as H_1 which takes an input *ID* of any length and outputs a point in the group G_1. The output point is taken as the entity's public key. That is, $Q_A = H_1(ID_A)$ is the public key of entity *A* with identity ID_A. Let e be a bilinear paring as defined above.

Initially, the key generation center (KGC), which is also a Trusted authority (TA), selects the system parameters $\{G_1, G_2, e, q, P, H_1\}$, chooses a random secret $s \in_R Z_q^*$ as its secret key, computes his public key $P_{KGC} = s \cdot P$ and finally publishes $\{G_1, G_2, e, q, P, H_1, P_{KGC}\}$. For each registered user *A* with his identity ID_A, his public key is given by $Q_A = H_1(ID_A)$ and the private key is $S_A = s \cdot Q_A$ which is sent by the KGC to the user via a secure channel.

3 Anonymous Key Agreement Schemes

Now we describe our key agreement schemes that consists of two-party key agreement, tripartite key agreement, and group key agreement. In the following, we assume that all entities are properly set up before the ad hoc network is formed. If this assumption does not stand for some applications, the idea of bootstrap the KGC [17] can be applied. In an ID-based scheme, all entities being properly set up mean that a unique identification mechanism is well known among the entities, and these entities get the public parameters and their private keys from the KGC before the ad hoc network is formed. That is, an entity *A* has got the public parameters and his private key $S_A = s \cdot Q_A$ from the KGC. In addition, in our schemes, all the registered entities get one additional secret from the KGC, the group secret $S_G = \frac{1}{s}P$. In the rest of this paper, both the term node and the term entity denote one mobile node. We also differentiate the entities in ad hoc networks into two kinds: *group members* denote those entities that have shared a well known identification mechanism and are authorized to join the ad hoc networks, and *group outsiders* denote those entities that may be eavesdroppers or adversaries and are not allowed to join the ad hoc networks. Our proposed schemes satisfy the anonymity against any *active group outsider* and against *passive group members* (who are not the partners of the sessions).

Now we summarize notations used in this paper as follows:

U_i / ID_i : U_i is the *i*th node with identity ID_i .

s / P_{KGC} : *s* is the secret key of the KGC, and P_{KGC} =*sP* is KGC's public key.

$S_G = \frac{1}{s}P$: the group secret that is shared among all the registered entities.

Q_{ID_i} / S_i : The public key of *node* i is $Q_{ID_i} = H_1(ID_i)$, and the private key is $S_i = sQ_{ID_i}$.

$Sig_A(m)$: node *A*'s signature on message *m*. Here, we suggest the use of Hess's ID-based signature [7], because it is efficient (it requires only one pairing operation) and has been proven secure in the random oracle model.

$E_k(m)$: the symmetric key encryption using key k . The scheme should satisfy the indistinguishability under chosen plain text attack (IND-CPA) property.

D_{AB} :The pair-wise secret of *node A* and node *B*.

$H_1() / H_2() / H_3()$: $H_1 : \{0,1\}^* \to G_1$ is the *MaptoPoint* function [6]; a one-way hash function $H_2 : G_2 \to \{0,1\}^t$, where t is the bit length of the key for symmetric encryption; a hash function $H_3 : \{0,1\}^* \to \{0,1\}^q$. The hash functions are modeled as random oracles in the security proofs.

3.1 Static Pair-Wise Key Agreement

Initially, each registered node *A* receives its private key $S_A = sQ_A = sH_1(ID_A)$ **,** where $Q_A = H_1(ID_A)$ is the public key. Now *A* computes the shared secret $D_{AB} = e(S_A, Q_B) = e(Q_A, Q_B)^s$ with *B*. *B* computes the shared secret $D_{BA} = e(Q_A, S_B) = e(Q_A, Q_B)^s$. Finally, the shared symmetric secret key is $K = H_2(D_{AB}) = H_2(D_{BA})$ which will be used to encrypt the communications between *A* and *B*. Note that any two nodes can generate this static key without any interaction.

3.2 Dynamic pair-wise key agreement

To further provide dynamic pair-wise session key, we propose a new two-party key agreement with anonymity as follows. Assume *A* and *B* are close to one another, and they can detect the existence of each other (for example, by broadcasting a special format beacon like that in Aloha network) and want to establish an authenticated session key without disclosing their identities to outsiders. In the following, *sid* denotes the session identifier that can uniquely identify one session from others, and $A \Rightarrow$ all denotes A broadcasts its messages to its neighbors.

1. $A \Rightarrow$ all: sid, $P_A = xP_{KGC}$

   A first chooses a random integer $x \in Z_q^*$, computes and sends $P_A = xP_{KGC}$ to B.

2. $B \Rightarrow$ all: sid, $P_B = yP_{KGC}$, $E_{k_1}(ID_B \| Sig_B(H_3(sid \| ID_B \| P_A \| P_B)))$

   B chooses a random integer $y \in Z_q^*$, computes $D_{AB} = e(P_A, S_G)^y = e(xP_{KGC}, 1/sP)^y = e(P,P)^{xy}$, $k_1 = H_2(D_{AB})$ and $Sig_B(H_3(m))$, where $m = sid \| ID_B \| P_A \| P_B$. Then B use k_1 as the encrypting key to encrypt the data $ID_B \| Sig_B(H_3(m)$.

3. $A \Rightarrow$ all: sid, $E_{k_1}(ID_A \| ID_B Sig_A(H_3(sid \| ID_A \| ID_B \| P_A \| P_B)))$

   Upon receiving the data in Step 2, A first computes $D_{AB} = e(P_B, S_G)^x = e(P,P)^{xy}$ and $k_1 = H_2(D_{AB})$, and then uses k_1 to decrypt the second part of the data to derive $ID_B \| Sig_B(...)$. Now A learns the identity ID_B of its communicating party, and verifies whether the signature $Sig_B(...)$ is valid. If the verification succeeds, then it generates its signature on the data $m = sid \| ID_A \| ID_B \| P_A \| P_B$ as $Sig_A(H_3(m))$, and sends $E_{k_1}(ID_A \| ID_B \| Sig_A(H_3(m)))$ to B. The final session key K_{sess} is computed as $K_{sess} = H_2(k_1 \| sid \| ID_A \| ID_B)$.

   Upon receiving the response $E_{K_1}(ID_A \| ID_B \| Sig_A(H_3(m)))$ from A, B first uses k_1 to decrypt the data and gets $ID_A \| ID_B \| Sig_A(...)$. Now B learns the identity, ID_A, and can verify whether the signature is valid. If the verification succeeds, then B accepts the message, and computes the final session key $K_{sess} = H_2(k_1 \| sid \| ID_A \| ID_B)$.

3.3 Tripartite Key Agreement with Anonymity

We now describe our tripartite key agreement which can be used to set up secure communication among three entities and can be used as a primitive for set up the group key for group broadcasting.

Assume *A, B, and C* are three nodes that detect the existence of each other, and want to establish session keys among them. They can perform the following tripartite key agreement protocol to establish the session key without disclosing their identities to outsiders. Our protocol consists of two rounds where the entities broadcast their ephemeral public keys in the first run and the entities broadcast their encryption on signatures and the identity in the second round. The protocol is described as follows.

Round 1:

1.1. $A \Rightarrow$ all: sid, $P_A = aP_{KGC}$

1.2. $B \Rightarrow$ all: sid, $P_B = bP_{KGC}$

1.3. $C \Rightarrow$ all: sid, $P_C = cP_{KGC}$

A computes $P_A = aP_{KGC}$, where a is a random number chosen by A. A broadcasts (sid, P_A). Likewise, B/C respectively chooses a random integer b/c, computes and broadcasts the ephemeral public keys P_B / P_C respectively.

Round 2:

2.1. $A \Rightarrow$ all: sid, $E_{k_1}(ID_A \| Sig_A(H_3(m_A)))$

2.2. $B \Rightarrow$ all: sid, $E_{k_1}(ID_B \| Sig_B(H_3(m_B)))$

2.3. $C \Rightarrow$ all: sid, $E_{k_1}(ID_C \| Sig_C(H_3(m_C))$

Upon receiving the broadcast data in Step 1, A first computes $k_1 = H_2(sid \| e(P_B, S_G) \cdot e(P_C, S_G) \cdot e(P,P)^a \cdot e(P_B, P_C)^a) = H_2(sid \| e(P,P)^{a+b+c+abcs^2})$ and generates its signature $Sig_A(H_3(m_A))$, where $m_A = sid \| ID_A \| P_A \| P_B \| P_C$. Likewise, B/C respectively computes $k_1 = H_2(sid \| e(P_C, S_G) \cdot e(P_A, S_G) \cdot e(P,P)^b \cdot e(P_C, P_A)^b) = H_2(e(P,P)^{a+b+c+abcs^2})$ / $k_1 = H_2(sid \| e(P_A, S_G) \cdot e(P_B, S_G) \cdot e(P,P)^c \cdot e(P_B, P_A)^c) = H_2(e(P,P)^{a+b+c+abcs^2})$ and generates the signature $Sig_B(H_3(m_B))$ / $Sig_C(H_3(m_C))$, where $m_B = sid \| ID_B \| P_A \| P_B \| P_C$ and $m_C = sid \| ID_C \| P_A \| P_B \| P_C$. The final session key $K_{sess} = H_2(sid \| e(P,P)^{a+b+c+abcs^2} \| ID_A \| ID_B \| ID_C)$.

The proposed tripartite scheme is secure in terms of in-distinguishability and resistance to both the key-compromise impersonation attack and the insider attack against an actively attacker (except the TA) in a modified Bellare-Rogaway model.

3.4 Group Key Management

To derive the group key, we propose to build up the group key by dividing the group into a ternary tree with all the entities at the leaves, and iteratively run the tripartite key agreement protocol or the two-party key agreement, depending on the down-degree of the current parent node, from bottom to top to get the group key. For each derived secret k after applying the key agreement protocol at level i, the value kP_{KGC} will be used as the ephemeral public value for the key agreement protocol at the (i-1)th level. Also the node with the smallest identity in each subgroup will represent the subgroup to participate the (i-1)th level key agreement. The final derived key for the root node is the final group key for the whole group.

Take Figure 1 as an example. Entities 1~8 are arranged in the leaves, and the intermediate nodes represent the sub-groups covering the entities under the nodes. The root node represents the final group key. Initially, all leaves at level 3 respectively involve the protocol instances of their subgroups. Nodes 1, 2, 3 launch the tripartite key agreement to derive the subgroup key, say $k_{1,2,3}$. Nodes 4, 5, 6 involve in another instance to derive the subgroup key, say $k_{4,5,6}$. Node 7 and 8 initiate an instance of two-party key agreement protocol to derive the subgroup key, say $k_{7,8}$. At level 2, Node 1, 4, 7 respectively represents their subgroups to initiate the tripartite key agreement protocol for level 2. In this protocol instance, Node 1 uses

$k_{1,2,3}P_{KGC}$ as its ephemeral public value, Node 4 uses $k_{4,5,6}P_{KGC}$ as its ephemeral public value, and Node 7 uses $k_{7,8}P_{KGC}$ as its ephemeral public value. After this protocol instance, the group key corresponding to Node 12 is $K_{1\sim 8} = H_2(sid \| e(P,P)^{k_{1,2,3}+k_{4,5,6}+k_{7,8}+k_{1,2,3}\cdot k_{4,5,6}\cdot k_{7,8}\cdot s^2} \| ID_9 \| ID_{10} \| ID_{11})$. Since each leaf in the tree knows exactly one secret of ($k_{1,2,3}$, $k_{4,5,6}$, $k_{7,8}$), all the leaves can derive the group key $K_{1\sim 8}$.

To dynamically adapt to the membership change in ad hoc networks, the ternary tree is updated accordingly and the keys on the path from the lowest updated node to the root are refreshed, using the key agreement protocols. The computational complexity of this approach is O($\log_3 n$), which is more efficient than its counterparts [10] whose complexity is O($\log_2 n$). The security of the group key management is directly based on that of the two-party key agreement and that of the tripartite key agreement. Since both the two-party key agreement and the tripartite key agreement are secure, the group key agreement is secure.

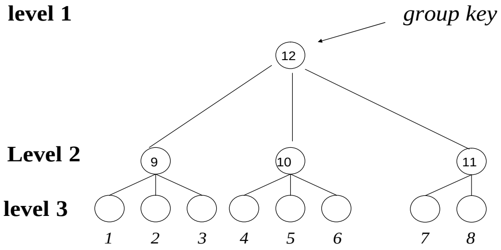

Figure 1. Bottom-up, divide-and-conquer to derive the group key

Table 1 summarizes the comparisons of our proposed schemes with its counterparts. Asokan-Ginzboorg’s key agreement scheme, and the key management schemes [16, 17] and Kaya et al.’s multicast scheme [14] are not listed in the comparisons, because Asokan-Ginzboorg’s location-based key agreement schemes are for special ad hoc networks, and Kaya et al. scheme focused only on group management that attaches joining node to the closest neighbor. The proposed schemes, Chien-Lin’s scheme [18] and Rhee et al.’s scheme require no on-line server support, which makes them more suitable for ad hoc networks. Also the three schemes provide formal proofs of the protocols, but Bohio-Miri’s scheme has security flaws. Our scheme and Chien-Lin’s scheme [18] provide efficient static pair-wise key agreement, efficient dynamic two-party key agreement and efficient tripartite key agreement, while Rhee et al.’s scheme only supports their costly two-party key agreement protocol. While Rhee et al.’s two-party key agreement protocol requires 5 message runs, our scheme requires only two message runs. Finally, only the proposed scheme here provides entity anonymity.

4 Conclusions and Future Work

This paper has discussed the infra-structure support property, the poor connectivity property, the anonymity property and the possible resource-limited property of mobile ad hoc networks. Based on ID-based cryptosystem from pairings, we have proposed our key agreement protocols with anonymity, and have proved the security in our model. The benefits of our proposed schemes include: (1) there is no requirement of on-line server support, (2) the protocols are efficient, and (3) the protocols preserve the entities' anonymity. These features make them very attractive to mobile ad hoc networks. As low-cost mobile devices become more and more popular, it is interesting to extend the results to those compromise-prone devices and to those resource-limited devices where public key cryptography is not feasible.

Table 1 Comparisons among secure schemes for ad hoc networks

	Rhee [10]	Bohio-Miri [12]	Chien-Lin [18]	The proposed scheme
Types of cryptosystems	ICPK*	ID-based, certificate-based	ID-based	ID-based
On-line server support	No	Yes	No	No
Security property	Formal proof	Security flaws (forgery problem)	Formal proof	Formal proof
Static pair-wise key	No	Yes	Yes	Yes
Cost of dynamic two-party key agreement	5 message runs, 5 T_E for one entity	No dynamic key agreement provided	2 runs, $2T_P+1T_M+1T_{Scalar}$ for one entity	** 3 runs, $2T_P+1T_M+1T_{Scalar}+2T_{ENC}$ for one entity
Efficient tripartite key agreement	No	No	Yes	Yes
Complexity of group key management	Group key agreement in $O(\log_2 n)$	The group key is chosen by the group leader.	Group key agreement in $O(\log_3 n)$	Group key agreement in $O(\log_3 n)$
Entity anonymity	No	No	No	Yes

* ICPK: Implicitly Certified Public keys.

** T_E denotes the cost of one modular exponentiation, T_{ENC} denote the cost of one symmetric encryption, T_P denotes that of one pairing operation, T_M denotes that of one modular multiplication, T_{scalar} denotes that of one scalar multiplication in G_1. Here assume that Hess's signature scheme is used to generate the signature.

Acknowledgements. We sincerely appreciate the anonymous reviewers for providing us the valuable comments. This research is partially supported by the National Science Council, Taiwan, R.O.C., under contract no NSC 95-2221-E-260 -050 -MY2.

References

1. Joux, A.: A One Round Protocol for Tripartite Diffie-Hellman. ANTS-IV 2000. LNCS, vol. 1838, pp.385—394. Springer, Heidelberg (2000)
2. Shamir, A..: Identity Based on Cryptosystems and Signature Schemes. Crypto'84. LNCS 196, pp. 47—53. Springer-Verlag (1984)
3. Bellare, M., Rogaway, P.: Provably Secure Session Key Distribution: The Three Party Case. In: 27th ACM Symposium on the Theory of Computing, pp. 57—66. ACM press (1995)
4. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated Key Exchange Secure against Dictionary Attacks. Eurocrypt 2000, LNCS, vol. 1807, pp. 139--155, Springer (2000)
5. Canetti, R., Krawczyk, H.: Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. Eurocrypt 2001, LNCS, vol. 2045, pp. 451--472. Springer (2001)
6. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. CRYPTO 2001, LNCS, vol. 2139, pp. 213—229. Springer (2001)
7. Hess, F.: Efficient Identity Based Signature Schemes Based on Pairings. SAC 2002, LNCS, vol. 2595, pp. 310—324. Springer-Verlag (2002)
8. Frey, G., Muller, M., Ruck, H..: The Tate Pairing and the Discrete Logarithm Applied to Elliptic Curve Cryptosystem. IEEE Trans. on I.T. 45(5), 1717--1719 (1999)
9. Chien, H.Y.: ID-Based Tripartite Multiple Key Agreement Protocol Facilitating Computer Auditing and Transaction Refereeing. Journal of Information Management 13(4), 185--204 (2006)
10. Rhee, K.H., Park,Y.H., Tsudik., G.: A Group Key Management Architecture for Mobile Ad-hoc Wireless Networks. Journal of Information Science and Engineering 21, 415—428 (2005)
11. Chen, L., Kudla, C.: Identity Based Authenticated Key Agreement Protocols from Pairings. Cryptology ePrint Archive, Report 2002/184 (2002)
12. Bohio, M., Miri, A.: Efficient Identity-Based Security Schemes for Ad Hoc Network Routing Protocols. Ad Hoc Networks 3, 309--317 (2004)
13. Asokan, N., Ginzboorg, P.: Key Agreement in Ad Hoc Networks. Computer Communications 23, 1627--1637 (2000)
14. Kaya, T., Lin, G., Noubir, G., Yilmaz, A.: Secure Multicast Groups on Ad Hoc Networks. Proc. of the 1st ACM Workshop Security of Ad Hoc and Sensor Networks, pp. 94--102 (2003)
15. Varadharajan, V., Shankaran, R., Hitchens, M.: Security for Cluster Based Ad Hoc Networks. Computer Communications 27, 488--501 (2004)
16. Zhu, B., Bao, F., Deng, R. H., Kankanhalli, M. S., Wang, G.: Efficient and Robust Key Management for Large Mobile Ad Hoc Networks. Computer Networks 48, 657--682 (2005)
17. Khalili, A., Katz, J., Arbaugh, W.A.: Toward Secure Key Distribution in Truly Ad-hoc Networks. Proc. of the 2003 Symp. on Applications and the Internet Workshop (2003)
18. Chien, H.Y., Lin, R.Y.: Improved ID-Based Security Framework for Ad-hoc Networks. Ad Hoc Networks, doi:10.1016/j.adhoc.2006.07.006 (2006)
19. Sadeghi, A.R., Steiner, M.: Assumptions Related to Discrete Logarithms: Why Subtleties make a Difference. EuroCrypt'01, LNCS, vol. 2045, pp. 243—260. Springer (2001)
20. Smart, N.P.: An Identity Based Authenticated Key Agreement Protocol Based on the Weil Pairing. Electronics Letters 38, 630--632 (2002)

21. Chen, L., Harrison, K., Soldera, D., Smart, N.: Applications of Multiple Trust Authorities in Pairing Based Cryptosystems. HP Journal, Feburary (2003)
22. McCullagh, N., Barreto, P.: A New Two-Party Identity-Based Authenticated Key Agreement. CT-RSA 2005, LNCS, vol. 3376, pp. 262--274. Springer-Verlag (2005)
23. Sakai, R., Ohgishi, K., Kasahara, M.: Cryptosystems Based on Pairing. In: the 2000 Symp. On Cryptography and Security (SCIS2000), Japan, 26-28 (2000)

Appendix. Security Notations and Proofs

The security of the proposed schemes concerns both the privacy of the authenticated session key and the privacy of the identities of the communicating parties. To capture the security of the tripartite key agreement scheme, we consider the in-distinguishability property [3-5], and the resistance to key-compromise impersonation, *known-key attack, forward secrecy* and the insider attack. We, therefore, prove the in-distinguishability in a modified model. Regarding the in-distinguishability, we adopt the BPR2000 model with some modifications- (1) extension to the tripartite case, (2) extension for the Corrupt query, and (3) adaptations to the identity anonymity.

The in-distinguishability of the proposed tripartite key agreement
Since the protocols hide the identities in the communications, in the model, the adversary AD cannot *fully* control the communications (*fully controls the partnership relation*) and, instead, *partially* controls the communications that take place between parties by interacting with a set of $\Pi^i_{U_1,*,*}$ oracles ($\Pi^i_{U_1,*,*}$ denotes that U_1 does not know its partners so far). In our protocol with anonymity, the adversary does not know the identities and the possible matching among oracles. We, therefore, try to model this situation by the following adaptations: (1) the adversary is allowed to send queries to specific oracle instances, but it does not know the partners of the oracle; (2) the challenger (or the simulator) randomly determines the matching among the instantiated oracles, and keeps the matching consistent through all the sessions but keeps the information secret from the adversary. The pre-defined oracle queries include - Hash query**,** Send(U_1, *, *, *i*, *m*), Reveal(U_1, *, *, *i*) , Corrupt(U_1), Test(U_1, *, *, *i*)**,** Sign (U_1, *i*, *m*). Note that, after an oracle has accepted, it knows the identities of its partners.

Security in the model is defined using the game *G*, played between the adversary and a collections of $\Pi^i_{U_x,*,*}$ oracles for players $U_x \in \{U_1, U_2, ..., U_{N_P}\}$ and instances $i \in \{1,...,l\}$. The adversary AD runs the game simulation *G* with setting as follows (we let the simulation *G* randomly determines the matching relationship among oracles, keeps it consistent through the simulation and keeps it secret from AD).

Stage 1: AD is able to send *Hash*, *Sign*, *Send*, *Reveal*, and *Corrupt* queries in the simulation.

Stage 2: At some point during *G*, AD will choose a fresh session and send a *Test* query to the fresh oracle associated with the test session. Depending on the randomly chosen bit *b*, AD is given either the actual session key or a session key drawn from the session key distribution.

Stage 3: AD continues making any *Hash, Sign, Send, Reveal* and *Corrupt* oracle queries to its choice.

Stage 4: Eventually, AD terminates the game simulation and output its guess bit *b'*.

Success of AD in *G* is measured in terms of AD's advantage in distinguishing whether AD receives the real key or a random value. Let the advantage function of AD be denoted by $Adv^{AD}(k)$, where k is the security parameter and $Adv^{AD}(k)$=2Pr[*b*=*b'*]-1.

Definition 6 (Secure tripartite key agreement protocol). A tripartite key agreement protocol is secure in our model if the following thee requirements are satisfied:

Validity: When the protocol is run among three oracles in the absence of an active adversary, the three oracles accept the same key.

Indistinguishability: For all probabilistic, polynomial-time adversaries AD, $Adv^{AD}(k)$ is negligible.

Security against insider impersonation and key-compromise impersonation: Even an insider (and a key-compromise impersonator) cannot impersonate another entity to the third entity and complete the session run with the third one.

Theorem 2. The proposed tripartite key agreement protocol is secure in the sense of Definition 6 if the underlying digital signature scheme is secure against the adaptively chosen message attack and the DBDH is hard.

Definition 7. We say that a tripartite key agreement scheme satisfies the entities anonymity if no probability polynomial time (PPT) distinguisher has a non-negligible advantage in the following game.

1. The challenger sets the system parameters (which might includes the group secret), and determines the private key/ public key pair, S_{ID_i}/Q_{ID_i}, for each $U_i \in \{U_1,...,U_{N_P}\}$. It hands the public parameters to the distinguisher *D*.

2. *D* adaptively queries the oracles defined in Appendix.

3. Once stage 2 is over, the challenger randomly chooses $b_1, b_2, b_3 \in \{1,...,N_P\}$ such that b_1, b_2, and b_3 are all different. The challenger lets U_{b_1}, U_{b_2}, and U_{b_3} be the three entities running a matching session, faithfully follows the protocol specification to generate the communication transcripts $trans*$ among the three oracles such that they follows the order (U_{b_1}, U_{b_2}, U_{b_3}) in generating their first round messages. It finally hands $trans*$ to *D*.

4. *D* adaptively queries the oracles as in stage 2 with the restriction that, this time, it is disallowed to send Reveal queries to the three target oracles in stage 3.

5. At the end of the game, *D* outputs $\overline{b}_1, \overline{b}_2, \overline{b}_3 \in \{1,...,N_P\}$ and wins if $\overline{b}_1 = b_1$ or $\overline{b}_2 = b_2$ or $\overline{b}_3 = b_3$. Its advantage is defined to be

$$Adv_{tripartite}^{anonymity}(D) := \Pr[\overline{b}_1 = b_1 \text{ or } \overline{b}_2 = b_2 \text{ or } \overline{b}_3 = b_3] - 3/N_P$$

Likewise, similar definitions can be defined and similar theorems can de derived for the two-party case. Due to page limitation, the detailed definitions and the proofs are omitted in this version.