ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2013, Vol.42, No.1

An Enhanced ID-based Authenticated Multiple Key Agreement Protocol

Zuowen Tan

Department of Computer Science, School of Information Technology,
Jiangxi University of Finance & Economics,Nanchang 330032, Jiangxi, China
e-mail: tanzyw@gmail.com

crossref http://dx.doi.org/10.5755/j01.itc.42.1.1381

Abstract. Authenticated multiple key agreement protocols provide secure communication between the participants via multiple session keys within one run of the protocol in an authentic way. Recently, Dehkordi and Alimoradi proposed an identity-based authenticated multiple key agreement protocol. Subsequently, Cheng presented ephemeral key compromise attack and impersonation attack against Dehkordi and Alimoradi's protocol. In order to overcome their security flaws, Cheng proposed an improvement on Dehkordi and Alimoradi's identity-based authenticated multiple key agreement protocol. In this paper, we demonstrate that Cheng's protocol is also insecure. Then we propose an identity-based multiple key agreement protocol which removes their weaknesses of the two protocols. A detailed analysis demonstrates that the proposed protocol can satisfy the strong security requirements.

Keywords: Multiple key agreement; Pairing; Identity; Ephemeral key.

1. Introduction

Before two parties transmit message over the pulic channel, they always run key agreement protocols [7,18,15,11,21] to generate a session key. The key is subsequently applied to achieve some cryptographic goals such as confidentiality or data integrity. In 1976, Diffie and Hellman [7] proposed the first key agreement protocol for two parties to establish a session key. Unfortunately, the original Diffie-Hellman protocol suffers from the main-in-the-middle attack because of lack of authentication between two communication parties. In order to avoid such an attack, Menezes et al. [19] proposed the MQV key agreement protocol, which is the first authenticated key agreement (AKA) protocol that used a signature without using a one-way hash function. Based on the MQV protocol, Harn and Lin [9] proposed an authenticated multiple-key agreement (AMKA) protocol to enable two communication parties to establish multiple session keys in one run of the protocol. Many AMKA protocol [8-10,12,13,16,17,27,29] have been presented after the Harn and Lin's works.

In 1984, Shamir [22] introduces identity-based cryptography in which an arbitrary string such an email address can be used as a user's public key. Identity-based cryptosystem can greatly simplify the public key management in the certificate-based public key infrastructures. A trusted authority (private key generator, PKG) is required to derive private keys from arbitrary public keys. Some identity-based key agreement (ID-KA) protocols [25,3,28] have been presented. But these ID-KA protocols cannot provide authentication function in which the users confirm their session keys by the succedent communication. Moreover, one run of these protocols [25,3,28] only can produce one session key at a time.

In 2002, Smart proposed the first identity-based authenticated key agreement (ID-AKA) protocol using bilinear pairings. The ID-AKA protocol not only allows participants to agree on session key but also ensures the authenticity of the other party. Many ID-AKA protocols subsequently using bilinear pairings have been developed [28,25,5,3,2] but they are not all secure. In 2004, Kim et al. presented an ID-based authenticated multiple-key (ID-AMKA) agreement protocol using pairing [14]. Recently, Dehkordi *et al*. proposed an efficient ID-AMKA protocol [6]. Dehkordi *et al*. claim that their protocol has stronger security. However, Cheng [4] pointed that Dehkordi *et al*.'s protocol is insecure against ephemeral key compromise attack and impersonation attack. In [4], an improvement on Dehkordi *et al*.'s protocol is presented .

A secure ID-AKA protocol should withstand the potential attacks. Based on the security attributes as

claimed [3,1,26,14,23,4], we highlight security requirements of ID-AMKA protocols as follows.

C1 (Mutual Authentication) ID-AMKA protocols not only allow participants to agree on the session keys but also ensure the authenticity of the other party. Thus, ID-AMKA protocols with mutual authentication can provide unknown key-share resilience. That is, one entity with ID_1 believes that she shares a key with an entity with ID_2, while the entity with ID_2 also believes that the key is shared with the entity with ID_1. In addition, mutual authentication ensures that ID-AMKA protocols hold key-compromise impersonation resilience. In other words, even if an adversary has corrupted one entity, e.g. *Alice*, and obtained *Alice*'s secret key, the adversary still can not impersonate the other entity, e.g. *Bob*, to the entity *Alice*.

C2 (Known-Key Secrecy) Session keys in one run of the protocol are independent of those ones generated during other executions of the protocol. Even though an adversary has obtained the participants' secret keys and some of session keys, the adversary cannot obtain other session keys in the other run of the protocol.

C3 (PKG Forward Secrecy) Even if an adversary has obtained the master secret key of the PKG, the previously established session keys will not be compromised. Since the secret keys of all the participants can be derived of the master secret key, PKG Forward Secrecy is a stronger security than Perfect Forward Security for ID-AMKA protocols.

C4 (No Key Control) Session keys are determined jointly by both the participants.

C5 (Mutual Security) Since one run of an ID-AMKA protocol will produce more than one session keys instead of only one session key, one has to consider whether other session keys will be recoverd when one or more session keys are disclosed. Suppose that an adversary either can obtain the master key of the PKG or can get the ephemeral private keys, but the adversary cannot do both. If the adversary has further obtained session keys, none of other session keys which are produced in the same run of the protocol can be recovered by the adversary. This is called *Multual Security* of ID-AMKA protocols. In essence, multual security is also necessary for the certificate-based MKA protocols. To the best of my knowledge,multual security of ID-AMKA protocols has not been referred to yet.

Most of the AMKA protocols [8,23] do not hold all the security requirements C1-C5. In this paper, we will demonstrate that the ID-AMKA protocols [6,4] fail to provide PKG Forward Secrecy and Multual Security. Then we will propose an enhanced ID-AMKA protocol which removes their weaknesses.

The remainder of this paper is organized as follows. In Section 2, we briefly review bilinear pairings, the cryptographic computational problems and some cryptographic assumptions. In Section 3, Cheng' protocol is reviewed and analyzed. We propose an enhanced ID-AMKA protocol in Section 4. Its security and performance analysis is given in Section 5. Section 6 concludes.

2. Preliminaries

In this section, we briefly review the properties of bilinear pairings and some related cryptographic security assumptions.

2.1. Bilinear pairings

Let $\{G_1, G_2\}$ be two cyclic groups of a large prime order q, and P be a generator of G_1. A bilinear pairing is defined by $\hat{e}: G_1 \times G_1 \to G_2$ and satisfies the following properties:

(1) Bilinear. For all $U, V \in G_1$ G$_1$ and $a, b \in Z_p^*$, $\hat{e}(aU, bV) = \hat{e}(U, V)^b$.

(2) Non-degenerate. $\hat{e}(P, P) \neq 1_{G_2}$.

(3) Computability. For all $U, V \in G_1$, there exists a probabilistic polynomial time algorithm to compute $\hat{e}(U, V)$.

2.2. Security assumptions

Definition 1 (*Computational Diffie–Hellman (CDH) Problem*) Given $\{P, aP, bP\}$ in G_1 for some unknown $a, b \in Z_q^*$, the CDH problem is to compute abP.

The advantage of an algorithm F within probabilistic polynomial time t in solving the CDH problem is defined as

$$Adv_F^{CDH}(t) = \Pr\left[F(aP, bP) = abP \middle| a, b \in Z_q^*\right].$$

Definition 2 (*CDH Assumption*) Given (P, aP, bP) in G_1 for unknown $a, b \in Z_q^*$, there does not exist any adversary F with non-negligible advantage $Adv_F^{CDH}(t)$ within probabilistic polynomial time t to compute abP.

Definition 3 (*Bilinear Diffie–Hellman (BDH) Problem*) Given the elements (P, aP, bP, cP) in an additive cyclic group G_1 for some unknown $a, b, c \in Z_q^*$, the BDH problem is to compute $\hat{e}(P, P)^{abc}$.

The advantage of an algorithm F within probabilistic polynomial time t in solving the BDH problem is defined as

$$Adv_F^{BDH}(t) = \Pr\left[F(P, aP, bP, cP) = \hat{e}(P, P)^{abc} \middle| a, b, c \in Z_q^*\right].$$

Definition 4 (*BDH Assumption*) Given (P, aP, bP, cP) for some unknown $a, b, c \in Z_q^*$, there does not exist any adversary F with non-negligible advantage $Adv_F^{BDH}(t)$ within probabilistic polynomial time t to compute $\hat{e}(P, P)^{abc}$.

3. Review and cryptanalysis of Cheng's Protocol

In this section, we first review Cheng's protocol [4] and then analyze its security.

3.1. Review of Cheng's Protocol

Cheng's protocol is composed of three phases: system initialization, key-extract and key-agreement. It is involved with three participants: a private key generator (PKG), an initiator Bob with identity ID_1 set and a responder Alice with identity ID_2.

(1) System Initialization Phase

Let G_1 and G_2 be two cyclic groups of a large prime order q. P is a generator of G_1. Let $\hat{e}$ be a bilinear pairing $\hat{e}: G_1 \times G_1 \to G_2$. PKG chooses a random value $s \in Z_q^*$ as the master secret key and computes $P_{pub} = sP$ as the public key.

PKG selects three cryptographic hash functions $H: \{0,1\}^* \to G_1, H_1: \{0,1\}^* \to Z_q^*$ and $H_2: \{0,1\}^* \to Z_q^*$. The system's public parameters are $\{G_1, G_2, \hat{e}, H_1(), H_2(), H(), q, P, P_{pub}\}$.

(2) Key-extract Phase

PKG runs the key extract algorithm and issues a public/secret key pair (Q_{ID}, S_{ID}) through a secure channel to every participant with identity *ID*, where $Q_{ID} = H(ID)$ and $S_{ID} = sQ_{ID}$. The participant can verify the key pair by checking if the following equation holds: $\hat{e}(P_{pub}, Q_{ID}) = \hat{e}(P, S_{ID})$.

(3) Key-agreement phase

Suppose that Alice with ID_1 and Bob with ID_2 attempt to agree on multiple key over open network. Throughout the paper, assume that Alice and Bob have their public/secret key pair (Q_1, S_1) and (Q_2, S_2), respectively. Alice and Bob can establish four keys in one run of the protocol.

Step 1:Bob selects a random value $r_B \in Z_q^*$, computes $C = r_B Q_2$ and sends $\{C, ID_2\}$ to Alice.

Step 2:Alice first selects a random value $r_A \in Z_q^*$, computes

$$T = r_A Q_1,$$

$$f_A = \hat{e}(r_A S_1 + H_2(T, ID_1, ID_2) S_1, C + H_2(C, ID_2, ID_1) Q_2),$$

$$\bar{X} = H_1(f_A, ID_1, ID_2),$$

and sends $\{T, \bar{X}, ID_1\}$ to Bob.

Step 3: Bob computes

$$f_B = \hat{e}(T + H_2(T, ID_1, ID_2) Q_1, r_B S_2 + H_2(C, ID_2, ID_1) S_2),$$

$$\bar{Y} = H_1(f_B, ID_2, ID_1),$$

$$X_B = H_1(f_B, ID_1, ID_2),$$

and checks if $\bar{X} = X_B$. If the verificaion equation holds, Bob sends $\{\bar{Y}, ID_2\}$ to Alice. Finally, Bob computes the session keys

$$K_{B1} = \hat{e}(T, S_2)^{r_B}, K_{B2} = \hat{e}(Q_1, S_2) K_{B1},$$

$$K_{B3} = \hat{e}(Q_1, S_2)^{r_B} K_{B1}, K_{B4} = \hat{e}(T, S_2) K_{B1}.$$

Step 4: Alice calculates

$$Y_A = H_1(f_A, ID_2, ID_1)$$

and checks if the following equation holds: $\bar{Y} = Y_A$. If the equation holds, Alice computes the session keys

$$K_{A1} = \hat{e}(S_1, C)^{r_A}, K_{A2} = \hat{e}(S_1, Q_2) K_{A1},$$

$$K_{A3} = \hat{e}(S_1, C) K_{A1}, K_{A4} = \hat{e}(S_1, Q_2) K_{A1}.$$

3.2. Security analysis of Cheng's Protocol

Cheng's protocol [4] has overcome the weaknesses of Dehkordi *et al.*'s ID-AMKA protocol [6]. The improved protocol can resist against the impersonation attack and the ephemeral key compromise attack. However, Cheng's protocol is still insecure. In this subsection, we will demonstrate that Cheng's protocol cannot provide PKG Forward Security. Moreover, Cheng's protocol lacks Mutual Security. Dehkordi *et al.*'s ID-AMKA protocol suffers from the same security vulnerability as Cheng's protocol.

(1) Cheng's protocol cannot provide PKG forward secrecy.

Cheng's protocol achieves perfect forward secrecy. If private keys of the participants are disclosed, the secrecy of previous session keys is not affected. But, if the master key *s* of PKG is disclosed, the previous keys can be dervied from the transmitted message over the public channel. This is because the four session keys in Cheng's protocol can be calculated as follows:

$$K_{B1} = K_{A1} = \hat{e}(T, S_2)^{r_B} = \hat{e}(T, C)^s,$$

$$K_{B2} = K_{A2} = \hat{e}(Q_1, S_2) K_{B1} = \hat{e}(Q_1, Q_2)^s \hat{e}(T, C)^s,$$

$$K_{B3} = K_{A3} = \hat{e}(Q_1, S_2)^{r_B} K_{B1} = \hat{e}(Q_1, C)^s \hat{e}(T, C)^s,$$

$$K_{B4} = K_{A4} = \hat{e}(T, S_2) K_{B1} = \hat{e}(T, Q_2)^s \hat{e}(T, C)^s.$$

Therefore, Cheng's protocol cannot provide PKG forward security.

(2) Cheng's protocol cannot provide mutual security.

Mutual security is necessary for the security of mulitple key agreement protocols. It refers to the security that compromise of both long-term private keys and one or more session keys will not lead to the compromise of the other session keys in the same run of the protocol. Some mulitple key agreement protocols in the literature cannot provide mutual security [23,8]. We will show that Cheng's protocol also lacks mutual security.

Suppose that an adversary F has intercepted the transmitted messages {*C*,*T*} between Alice and Bob. If F has obtained Alice's private key S_1 and one session key $K_{B1}(K_{A1})$, F can recover $K_{B2}(K_{A2})$ and $K_{B3}(K_{A3})$:

$$K_{B2} = K_{A2} = \hat{e}(S_1, Q_2) K_{B1},$$

$$K_{B3} = K_{A3} = \hat{e}(S_1, C) K_{B1}$$

If F has obtained Bob's private key S_2 and one session key $K_{B1}(K_{A1})$, F can recover $K_{B2}(K_{A2})$ and $K_{B4}(K_{A4})$:

$$K_{B2} = K_{A2} = \hat{e}(Q_1, S_2) K_{B1},$$

$$K_{B4} = K_{A4} = \hat{e}(T, S_2) K_{B1}.$$

If F has obtained Alice's private key S_1 and one session key $K_{B2}(K_{A2})$, F can recover $K_{B1}(K_{A1})$ and $K_{B3}(K_{A3})$:

$$K_{B1} = K_{A1} = \hat{e}(S_1, Q_2)^{-1} K_{B2},$$

$K_{B3} = K_{A3} = \hat{e}(S_1, C)\hat{e}(S_1, Q_2)^{-1}K_{B2}$.

If F has obtained Bob's private key S_2 and one session key $K_{B2}(K_{A2})$, F can recover $K_{B1}(K_{A1})$ and $K_{B4}(K_{A4})$

$K_{B1} = K_{A1} = \hat{e}(Q_1, S_2)^{-1}K_{B2}$,

$K_{B4} = K_{A4} = \hat{e}(T, S_2)\hat{e}(Q_1, S_2)^{-1}K_{B2}$.

Likewise, if F has obtained Alice's private key S_1 and one session key $K_{B3}(K_{A3})$ or Bob's private key S_2 and one session key $K_{B4}(K_{A4})$, F can recover $K_{B1}(K_{A1})$ and $K_{B2}(K_{A2})$.

From the above analysis, one easily knows that if the adversary *F* obtains both Alice 's private key and Bob's private key, F can recover all the other three session keys with knowledge of any session key.

Therefore, Cheng's protocol cannot provide mutual security. Similar analysis can be applied to Dehkordi *et al*.'s ID-AMKA protocol [6].

4. The enhanced ID-AMKA protocol

In the section, we propose an enhanced ID-AMKA protocol. The new ID-AMKA scheme is composed of three phases: setup, key-extract and key-agreement. Assume that two participants Alice with identity ID_1 and Bob with identity ID_2 attempt to agree on session keys.

4.1. Setup phase

The proposed ID-AMKA protocol has the same system parameters as in Cheng's protocol [4]. Let G_1 and G_2 be two cyclic groups of a large prime order q. is a generator of G_1. Let $\hat{e}$ be a bilinear pairing $\hat{e}: G_1 \times G_1 \to G_2$. PKG takes $s \in Z_q^*$ and $P_{pub} = sP$ as the master secret and public key. PKG selects cryptographic hash functions $H: \{0,1\}^* \to G_1, H_1: \{0,1\}^* \to Z_q^*$ and $H_2: \{0,1\}^* \to Z_q^*$. The system's public parameters are $\{ G_1, G_2, \hat{e}, H_1(), H_2(), H(), q, P, P_{pub}\}$.

4.2. Key-extract phase

PKG computes $Q_1 = H(ID_1)$, $S_1 = sQ_1$, $Q_2 = H(ID_2)$ and $S_2 = sQ_2$. And PKG issues a public/secret key pair (Q_1, S_1) and (Q_2, S_2) to Alice and Bob through a secure channel, respectively.

4.3. Key-agreement phase

Alice and Bob generate eight session keys in one run of the protocol by executing the following procedures.

Step 1: Alice selects random values $r_1, t_1, v_1 \in Z_q^*$ and computes

$R_1 = \hat{e}(P, P)^{r_1}, T_1 = t_1Q_1, U_1 = v_1P$

$Y = H_1(R_1 \| T_1 \| U_1)S_1 + r_1P$.

Next, Alice sends $\{R_1, T_1, U_1, Y, ID_1\}$ to Bob.

Step 2: Bob first validates the message $\{ R_1, T_1, U_1, Y, ID_1 \}$ by checking if the verification equation holds: $\hat{e}(Y, P) = \hat{e}(Q_1, P_{pub})^{H_1(R_1\|T_1\|U_1)}R_1$.

If the above equation does not hold, Bob aborts. Otherwise, Bob randomly selects $r_2, t_2, v_2 \in Z_q^*$ and computes

$R_2 = \hat{e}(P, P)^{r_2}, T_2 = t_2Q_2, U_2 = v_2P$

$Z = H_1(U_1 \| T_1 \| U_2 \| T_2)S_2 + r_2P$.

Next, Bob sends $\{ R_2, T_2, U_2, Z, ID_2 \}$ to Alice. Finally, Bob computes the shared secrets

$\sigma_B = v_2U_2, \sigma_{B0} = \hat{e}(\mathrm{T}_1, \mathrm{t}_2, \mathrm{S}_2), \sigma_{B1} = \hat{e}(U_1, v_2, P_{pub}),$

$\sigma_{B2} = \hat{e}(T_1, v_2, P_{pub}),$

$\sigma_{B3} = \hat{e}(Q_1, v_2, P_{pub}),\ \sigma_{B4} = \hat{e}(U_1, t_2S_2),$

$\sigma_{B5} = \hat{e}(U_1, S_2),\ \sigma_{B6} = \hat{e}(Q_1, S_2), \sigma_{B7} = \hat{e}(Q_1, t_2S_2).$

and the session keys:

$K_{B1} = H_2(\sigma_B, \sigma_{B0}, ID_1, ID_2, T_1, T_2, U_1, U_2)$,

$K_{B2} = H_2(\sigma_B, \sigma_{B0}, \sigma_{B1}, ID_1, ID_2, T_1, T_2, U_1, U_2)$,

…

$K_{Bi} = H_2(\sigma_B, \sigma_{B0}, \sigma_{B,i-1}, ID_1, ID_2, T_1, T_2, U_1, U_2)$,

…

$K_{B8} = H_2(\sigma_B, \sigma_{B0}, \sigma_{B7}, ID_1, ID_2, T_1, T_2, U_1, U_2)$,

where $i \in \{1,2,\ldots,7\}$.

Step 3: Upon receiving the message $\{R_2, T_2, U_2, Z, ID_2\}$, Alice first validates the message by checking if the verification equation holds: $\hat{e}(Z, P) = \hat{e}(Q_2, P_{pub})^{H_1(U_1\|T_1\|U_2\|T_2\|Y)}R_2$. If the above equation does not hold, Alice refuses the response. Otherwise, Alice computes the shared secrets

$\sigma_A = v_1U_2,\ \sigma_{A0} = \hat{e}(T_2, t_1S_1),\ \sigma_{A1} = \hat{e}(U_2, v_1P_{pub}),$

$\sigma_{A2} = \hat{e}(U_2, t_1S_1),\ \sigma_{A3} = \hat{e}(U_2, S_1),$

$\sigma_{A4} = \hat{e}(P_{pub}, v_1T_2),\ \sigma_{A5} = \hat{e}(P_{pub}, v_1Q_2),$

$\sigma_{A6} = \hat{e}(Q_2, S_1), \sigma_{A7} = \hat{e}(S_1, T_2).$

and the session keys:

$K_{A1} = H_2(\sigma_A, \sigma_{A0}, ID_1, ID_2, T_1, T_2, U_1, U_2)$,

$K_{A2} = H_2(\sigma_A, \sigma_{A0}, \sigma_{A1}, ID_1, ID_2, T_1, T_2, U_1, U_2)$,

…

$K_{Ai} = H_2(\sigma_A, \sigma_{A0}, \sigma_{A,i-1}, ID_1, ID_2, T_1, T_2, U_1, U_2)$,

…

$K_{A8} = H_2(\sigma_A, \sigma_{A0}, \sigma_{A7}, ID_1, ID_2, T_1, T_2, U_1, U_2)$,

where $i \in \{1,2,\ldots,7\}$.

5. Analysis on the proposed ID-AMKA protocol

In this section, we analyze the security and performance of the proposed ID-AMKA protocol.

5.1. Correctness

It is easy to check the correctness of the proposed ID-AMKA protocol. The shared secrets sastify

$\sigma_A = v_1U_2 = v_2U_1 = \sigma_B$,

$\sigma_{A0} = \hat{e}(T_2, t_1S_1) = \hat{e}(T_1, t_2S_2) = \sigma_{B0}$,

$\sigma_{A1} = \hat{e}(U_2, v_1P_{pub}) = \hat{e}(U_1, v_2P_{pub}) = \sigma_{B1}$,

$\sigma_{A2} = \hat{e}(U_2, t_1S_1) = \hat{e}(T_1, v_2P_{pub}) = \sigma_{B2}$,

$\sigma_{A3} = \hat{e}(U_2, S_1) = \hat{e}(Q_1, v_2P_{pub}) = \sigma_{B3}$,

$\sigma_{A4} = \hat{e}(P_{pub}, v_1T_2) = \hat{e}(U_1, t_2S_2) = \sigma_{B4}$,

$\sigma_{A5} = \hat{e}(P_{pub}, v_1Q_2) = \hat{e}(U_1, S_2) = \sigma_{B5}$,

$\sigma_{A6} = \hat{e}(Q_2, S_1) = \hat{e}(Q_1, S_2) = \sigma_{B6}$,

$\sigma_{A7} = \hat{e}(S_1, T_2) = \hat{e}(Q_1, t_2S_2) = \sigma_{B7}$.

Therefore, we have $K_{Ai} = K_{Bi}$ for all $i \in \{1,2,\ldots,8\}$.

5.2. Security Analysis

Theorem 1. *The proposed ID-AMKA scheme provides the mutual authentication (C1).*

▼**Proof:** In the proposed protocol, when Alice sends the request message, Alice produces an ID-based signature Y about the message $\{R_1, T_1, U_1, ID_1\}$. Similiary, Bob sends the response message $\{R_2, T_2, U_2, ID_2\}$ with a signature Z. The two ID-based signatures are secure against existential forgery-adaptively chosen message and ID attack in the random oracle model [24] via Forking Lemma [20], assuming the hardness of CDH problem. Thus, when Alice and Bob have obtained the session keys, they can authenticate each other. The proposed ID-AMKA protocol holds mutual authentication. So, the protocol also can provide unknown key-share resilience and key-compromise impersonation resilience.▲

Theorem 2. *The proposed ID-AMKA protocol provides Known-Key Secrecy (C2).*

▼**Proof:** Suppose that an adversary F has obtained Alice's and Bob's private keys $\{S_1,S_2\}$ and some session keys $\{K_i\}$. The adversary F attempts to recover the previous session keys in the other run of the protocol. From the key-agreement phase described in Session 4.3, F has to recover the secret values

$\sigma_B = \sigma_A = v_2v_1P$,

$\sigma_{A0} = \hat{e}(T_2, t_1S_1) = \hat{e}(T_1, t_2S_2) = \sigma_{B0}$,

$\sigma_{A1} = \hat{e}(U_2, v_1P_{pub}) = \hat{e}(U_1, v_2P_{pub}) = \sigma_{B1}$, etc. The secret values $\{\sigma_B, \sigma_{B0}, \sigma_{B1}, \sigma_{B2}, \sigma_{B4}\}$ or $\{\sigma_A, \sigma_{A0}, \sigma_{A1}, \sigma_{A2}, \sigma_{A4}\}$ are derived both from the private keys and from the ephemeral private keys. Since the ephemeral private keys $\{v_1, v_2, t_1, t_2\}$ are random values, the session key in one run of the protocol cannot help the adversary to compute new shared secret values in a different run of the protocol. Especially, when the adversary wants to compute σ_B or σ_A from $\{U_1, U_2\}$, he/she will have to be faced with an instance of the CDH problems. Therefore, the proposed ID-AMKA protocol holds Known-Key Secrecy.▲

Theorem 3. *The proposed ID-AMKA scheme provides PKG Forward Secrecy (C3).*

▼**Proof:** Suppose that the master secret key s is compromised. Then the adversary F can obtain Alice's and Bob's secret keys $\{S_1, S_2\}$. Assume that the adversary F has intercepted all the ephemeral keys $\{T_1, T_2, U_1, U_2\}$ transmitted between Alice and Bob. However, the adversary who knows the values $\{S_1, S_2\}$ and $\{T_1, T_2, U_1, U_2\}$ cannot compute the shared secret value $\sigma_B = v_1v_2P = \sigma_A$, since F has to be faced with a CDH problem: to compute v_1v_2P from $\{U_1, U_2\}$. If the adversary F has only obtain Alice's and Bob's secret keys $\{S_1, S_2\}$ without knowledge of the master secret key s, the adversary cannot compute the shared secret value $\sigma_{B1} = \hat{e}(U_1, v_2P_{pub}) = \sigma_{A1}$, since F has to be faced with a BDH problem: to compute $\hat{e}(P,P)^{t_1t_2s}$ from the triple $\{U_1, U_2, P_{pub}\}$. The adversary also cannot compute the shared secret value σ_{A0} or σ_{B0}, since F has to be faced with one CDH problem: to compute t_1S_1 from $\{T_1, S_1\}$ or to compute t_2S_2 from $\{T_2, S_2\}$. Likewise, The adversary cannot compute the shared secret value σ_{A2} or σ_{B2}, since F has to solve one CDH problem: to compute t_1S_1 from $\{T_1, S_1\}$ or to compute v_2P_{pub} from $\{U_2, P_{pub}\}$. Similarly, the adversary cannot obtain the shared secret value σ_{A4} or σ_{B4}, since F has to solve one CDH problem: to compute v_1P_{pub} from $\{U_1, P_{pub}\}$ or to compute t_2S_2 from $\{T_2, S_2\}$.

From the above analysis, to learn the previous session keys, even if PKG's master key is disclosed, the adversary still has to get the corresponding ephemeral private keys. Therefore, under the CDH assumption and BDH assumption, the proposed ID-AMKA protocol achieves perfect forward secrecy.▲

Theorem 4. *The proposed ID-AMKA protocol provides No Key Control (C4).*

▼**Proof:** Session key K_i (i=1,2,3,…,8) is computed from the shared secret values $\sigma_A, \sigma_{A0}, \sigma_{Ai}$or $\sigma_B, \sigma_{B0}, \sigma_{Bi}$ These shared secret values $\sigma_A, \sigma_{A0}, \sigma_{Ai}$ or $\sigma_B, \sigma_{B0}, \sigma_{Bi}$ are derived of the ephemeral keys $\{U_1, U_2, T_1, T_2\}$. The keys $\{U_1, T_1\}$ and $\{U_2, T_2\}$ are chosen by Alice and Bob, respectively. Since the verification parts are secure signatures, neither of Alice and Bob can preselect the ephermeral keys of the other participant or predetermine the session keys. In other words, session keys are determined cooperatively by Alice and Bob.▲

Theorem 5. *The proposed ID-AMKA protocol provides Mutual Security(C5).*

▼**Proof:** Suppose that an adversary F either can obtain the master key s or can get the ephemeral private keys, but F cannot have both the master key and the ephemeral private keys. The assumption is reasonable. If both the keys are compromised, F can reveal all the shared secrets from which F can further compute all the session keys. Here, we discuss it in the two cases.

Case 1: Some of the session keys $K_i \in [1,8]$ and all the ephemeral private keys are compromised.

Every session key $K_i \in [1,8]$ is a hash value of secure hash function $H_2()$. Even if some of session keys are comprised, due to the onewayness of the hash function, F is still unable to recover their pre-image $(\sigma_B, \sigma_{B0}, \sigma_{Bi})$ or $(\sigma_A, \sigma_{A0}, \sigma_{Ai})$. If F wants to compute other session keys, F must compute at least the shared secret values $\sigma_B = v_1 v_2 P = \sigma_A$ and σ_{A0} or σ_{B0}. Since F has the knowledge of the ephemeral private keys $\{v_1, v_2\}$, F can obtain σ_{B0} or σ_{A0}. However, since $\sigma_{A0} = \hat{e}(T_2, t_1 S_1)$, $\hat{e}(T_1, t_2 S_2) = \sigma_{B0}$, it is infeasible to compute σ_{A0} or σ_{B0} without the knowledge of master secret s or the participants's secret key S_1 or S_2.

Case 2: Some of the session keys $K_i \in [1,8]$ and the master secret key s are compromised.

By similar analysis to Case 1, we know that even if F has obtained some of session key, F is still unable to recover their pre-image $(\sigma_B, \sigma_{B0}, \sigma_{Bi})$ or $(\sigma_A, \sigma_{A0}, \sigma_{Ai})$ from those session keys K_i. This is because that the shared secret values $(\sigma_B, \sigma_{B0}, \sigma_{Bi})$ are protected by the secure hash function $H_2()$. If F wants to compute other session keys, F must compute the shared secret values, say, $\sigma_B(\sigma_A)$ and $\sigma_{A0}(\sigma_{B0})$. Since F has the knowledge of the master key s, F can compute the shared secret value $\sigma_{A0}(\sigma_{B0})$ from the emphemeral public keys:

$\sigma_{A0} = \hat{e}(T_2, t_1)^s$ or $\sigma_{B0} = \hat{e}(T_2, T_1)^s$.

However, when F computes the shared secret value $\sigma_B(\sigma_A)$ from the emphemeral public key, F has to be faced with a CDH problem: to compute $v_1 v_2 P$ from $\{U_1, U_2\}$.

Without the knowledge of emphemeral private keys $\{v_1 v_2\}$, it is infeasible to compute $\sigma_B(\sigma_A)$ under the CDH assumption.▲

5.3. Performance Analysis and comparisons

We analyze the performance of the proposed *ID-AMKA* protocol in terms of the security property and efficiency. We make the security property and performance comparison with the previous AMKE protocols.

The AMKE protocols [10, 13, 17,27] adopt certificate to provide the authentication of the public key. The participants must verify the certificate of the other participant before using his/her public key. As a consequence, the protocols require a high computation cost and a large amount of storage. In the following, we compare the proposed ID-AMKA protocol with the previous ID-AMKA protocols. The *ID-AMKA* protocols consist of phases: setup, key-extraction and key-agreement. Since the setup and key-extraction phases are executed once before the protocol runs, we don't include their time cost in the comparison result. In contrast with four session keys produced in one run of the ID-AMKA protocols in [14,6,4], the proposed ID-AMKA protocol can establish eight session keys.

To evaluate the computational complexity, we define T_S,T_A,T_p,T_M and T_E as one scalar multiplication in G_1, one point addition in G_1, one bilinear pairing computation in G_2, one multiplication computation in G_2 and one exponent computation in G_2, respectively. Since the addition operation and hash operation cost very little, we omit them. We give the efficiency comparisons in **Table 1**. As shown in **Table 1**, the protocols in [6,4] need three passes of message. The proposed protocol requres lower computation cost than the ID-AMKA protocols in [6]. Compared with the ID-AMKA protocols in [14], the proposed protocol requres a little more computation cost. However, Section 3 shows that the ID-AMKA protocol in [4] fails to provide PKG forward security and mutual security (for details,also see Table 2).

In **Table 2**, we compare the security attributes of the proposed protocol and the ID-AMKA protocols [14,6,4] including the certificate-based AMKA protocols [10,13,17,27]. Section 5.2 shows that the proposed protocol satisfies all the strong security requirements C1-C5. From Table 2, the AMKA protocols in [10,13,17,27] and the ID-AMKA protocols in [4,14,6] cannot provide PKG Forward Secrecy. For example, in the ID-AMKA protocol [14], the session keys can be computed as follows:

$K_1 = \hat{e}(T_{A1}, T_{B1})^s$, $K_2 = \hat{e}(T_{A1}, T_{B2})^s$,

$K_3 = \hat{e}(T_{A2}, T_{B1})^s$, $K_4 = \hat{e}(T_{A2}, T_{B2})^s$,

where $\{T_{A1}, T_{B1}, T_{A2}, T_{B2}\}$ are the ephemeral public keys. Therefore, once the master key s is compromised, all the session keys can be recovered

Table 1. Efficiency comparisons

	[14]	[4]	[6]	Ours
E1	√	√	√	√
E2	4	4	4	8
E3	2	3	3	2
E4	$7T_S+4T_p+4T_E+3T_A$	$4T_S+3T_p+2T_E+3T_M+2T_A$	$3T_S+6T_p+2T_E+3T_M$	$5T_S+10T_p+5T_E+T_M$

E1: ID-AKA protocol; E2: Number of session keys; E3: Number of pass; E4: Total computation cost.

Table 2. Features comparisons

	[13]	[17]	[27]	[10]	[14]	[4]	[6]	Ours
C1	×	×	×	×	√	√	×	√
C2	√	√	√	√	√	√	√	√
C3	×	×	×	×	×	×	×	√
C4	√	√	√	√	√	√	√	√
C5	×	×	×	×	√	×	×	√
D1	×	×	×	×	√	√	×	√
D2	modification attack, forgery signature attack	impersonation attack	reflection attack, forgery attack	impersonation attack, forgery attack	√	√	impersonation attack	√

C1 Mutual Authentication; C2 Known-Key Secrecy; C3 PKG Forward Secrecy; C4 No Key Control; C5 Mutual Security;D1 Resistence against Ephemeral key compromise attack;D2 Resistence against other potential attacks

from the ephemeral public keys. The AMKA protocols in [10,13,17,27] cannot provide Mutual Security. The analysis in Section 3 shows that the ID-AMKA protocols in [4,6] cannot provide PKG Forward Secrecy and Mutual Security. In addition, Zhou et al. [29] showed that Harn and Lin's protocol [10] suffers from forgery attack and only three of these keys can provide perfect forward secrecy. Hwang *et al*.'s protocol [13] suffers from the modification attack [16] and forgery signature attack [12]. Lee *et al*.'s protocol [17] is vulnerable to impersonation attack [27]. Moreover, Farash *et al*. [8] demonstrated that Lee *et al*.'s protocol [17] cannot hold Mutual Security. M. S. Vo *et al*.'s protocol [27] is vulnerable to another kind of forgery attacks and a reflection attack [8].

6. Conclusion

In this paper, we define the two strong security properties PKG Forward Secrecy and Mutual Security of ID-AMAK protocols. We have found that the previous AMAK protocols fail to provide these security properties. Our analysis shows that the ID-AMAK protocols [4,14,6] fail to PKG Forward Secrecy and Mutual Security. To remove the security vulnerabilities, we have proposed an enhanced ID-AMKA protocol. Performance and security analysis demonstrates that the proposed protocol has better performance and stronger security as compared with the previous ID-AMAK protocols. To design an efficient ID-AMAK protocol with the security proof in a formal model is our future research.

Acknowledgement

This work is partially supported by the National Natural Science Foundation of China under grant No.61163053 and Natural Science Foundation of Jiangxi Province under grant No.20122BAB201035.

References

[1] **S. Blake-Wilson, A. Menezes.** Authenticated Diffie Hellman key agreement protocols. In: *Proceedings of the SAC' 98*, LNCS, Vol. 1556. Berlin: Springer Verlag, 1999, pp. 339-361. Available at: http://dx.doi.org/10.1007/3-540-48892-8_26.

[2] **C. Boyd, Y. Cliff, J. M. González Nieto, K. G. Paterson**. Efficient one-round key exchange in the standard model. In: *Information Security and Privacy—ACISP 2008*, LNCS, Vol. 5107, Springer, Berlin, 2008, pp. 69–83. Available at: http://dx.doi.org/10.1007/978-3-540-70500-0_6.

[3] **L. Chen, Z. Cheng, N. P. Smart**. Identity based key agreement protocols from pairings. In: *International Journal of Information Security*, 2007, Vol. 6, No. 4, pp. 213-241. Available at: http://dx.doi.org/10.1007/s10207-006-0011-9.

[4] **Q. F. Cheng.** Cryptanalysis of an Identity-Based Multiple Key Agreement Scheme, Available at: http://eprint.iacr.org/2012/410.pdf.

[5] **K. K. R. Choo, S. S. M. Chow.** Strongly-secure identity-based key agreement and anonymous extension. In: *Information Security—ISC 2008*, LNCS, Springer, Berlin, 2007, Vol. 4779, pp. 203–220. Available at: http://dx.doi.org/10.1007/978-3-540-75496-1_14.

[6] **M. H. Dehkordi, R. Alimoradi.** Identity-based Multiple Key Agreement Scheme. In: *KSII Transactions on Internet and Information Systems*, 2011, Vol. 5, No. 2, pp. 2392-2402. Available at: http://dx.doi.org/10.3837/tiis.2011.12.007

[7] **W. Diffie, M. E. Hellman.** New directions in cryptography. In: *IEEE Transactions on Information Theory*, Vol. 22, No. 6, 1976, pp. 644-654. Available at: http://dx.doi.org/10.1109/TIT.1976.1055638.

[8] **M. S. Farash, M. Bayat, M. A. Attari.** Vulnerability of two multiple-key agreement protocols. In: *Computers and Electrical Engineering*, 2011, Vol. 37, No. 2, pp. 199–204. Available at: http://dx.doi.org/10.1016/j.compeleceng.2011.02.007.

[9] **L. Harn, H. Y. Lin**. An authenticated key agreement protocol without using one-way functions. In: *Proceedings of the 8th National Conference on Information Security*, Kaohsiung, Taiwan, May 1998, pp. 155–160.

[10] **L. Harn, H. Y. Lin.** Authenticated key agreement without using one-way hash function. In: *Electronics Letter*, 2001, Vol. 37, No. 10, pp. 629-630. http://dx.doi.org/10.1049/el:20010441.

[11] **J. W. Hong, S. Y. Yoon, D. I. Park**. A new efficient key agreement scheme for VSAT satellite communications based on elliptic curve cryptosystem. In: *Information Technology and Control*, 2011, Vol. 40, No. 3, pp. 252-259. Available at: http://dx.doi.org/10.5755/j01.itc.40.3.634.

[12] **M. S. Hwang, T. Y. Chang, S. C. Lin, C. S. Tsai.** On the security of an enhanced authentication key exchange protocol. In: *18th International Conference on Advanced Information Networking and Applications*, 2004, Vol. 2, pp. 160-163. Available at: http://dx.doi.org/10.1109/AINA.2004.1283777.

[13] **R. J. Hwang, S. H. Shiau, C. H. Lai**. An enhanced authentication key exchange protocol. In: *Proceedings of the 17th international conference on AINA*, 27-29 March 2003, pp. 20-25. Available at: http://dx.doi.org/10.1109/AINA.2003.1192871.

[14] **K. W. Kim, E. K. Ryu, K. Y. Yoo**. ID-Based Authenticated Multiple-Key Agreement Protocol from Pairings. In: *ICCSA, LNCS* 2004, Vol. 3046, pp. 672-680. Available at: http://dx.doi.org/10.1007/978-3-540-24768-5_72.

[15] **C. C. Lee, T. C. Lin, M. S. Hwang.** A key agreement scheme for satellite communications. In: *Information Technology and Control*, 2010, Vol. 39, No. 1, pp. 43-47.

[16] **N. Y. Lee, C. N. Wu**. Improved authentication key exchange protocol without using one-way hash function. In: *ACM SIGOPS Operating Systems Review*, 2004, Vol. 38, No. 2, pp. 85-92. Available at: http://dx.doi.org/10.1145/991130.991139.

[17] **N. Y. Lee, C. N. Wu, C. C. Wang**. Authenticated multiple key exchange protocols based on elliptic curves and bilinear pairings. In: *Computers and Electrical Engineering*, 2008, Vol. 34, No. 1, pp. 12-20. Available at: http://dx.doi.org/10.1016/j.compe-leceng.2006.11.005.

[18] **J. W. Lo, S. C Lin, M. S. Hwang**. A parallel password-authenticated key exchange protocol for wireless environments. In: *Information Technology and Control*, 2010, Vol. 39, No. 2, pp. 146–151.

[19] **A. J. Menezes, M. Qu, S. A. Vanstone**. Some key agreement protocols providing implicit authentication. In: *Proceedings of 2nd Workshop Selected Areas in Cryptography*, 1995, pp. 22–32.

[20] **D. Pointcheval, J. Stern.** Security arguments for digital signatures and blind signatures. In: *Journal of Cryptology*, Vol. 13, No. 3, 2000, pp. 361–396. http://dx.doi.org/10.1007/s001450010003.

[21] **E. Sakalauskas, A. Katvickis, G. Dosinas**, "Key agreement protocol over the ring of multivariate polynomials,"*Information Technology and Control*, Vol. 39, No. 1, 2010, pp. 43-47.

[22] **A. Shamir.** Identity-based cryptosystems and signature schemes. In: *Advances in Cryptology- CRYPTO* '84, LNCS, 1984, Vol. 196, pp. 47-53. Available at: http://dx.doi.org/10.1007/3-540-39568-7_5.

[23] **K. A. Shim.** Vulnerabilities of generalized MQV key agreement protocol without using one-way hash functions. In: *Computer Standards & Interfaces*, 2007, Vol. 29, No. 4, pp. 467-470. Available at: http://dx.doi.org/10.1016/j.csi.2006.11.002.

[24] **H. Singh, G. K. Verma.** ID-based proxy signature scheme with message recovery. In: *The Journal of Systems and Software*, 2012, Vol. 85, No. 1, pp. 209-214. Available at: http://dx.doi.org/10.1016/j.jss.2011.08.018.

[25] **N. P. Smart.** An identity based authenticated key agreement protocol based on the Weil bilinear pairing. In: *Electronics Letters*, 2002, Vol. 38, No. 1, pp. 630-632. Available at: http://dx.doi.org/10.1049/el:20020387.

[26] **Z. W. Tan.** Efficient identity-based authenticated multiple key exchange protocol. In: *Computers and Electrical Engineering*, Vol. 37, 2011, pp. 191-198. http://dx.doi.org/10.1016/j.compeleceng.2011.02.006.

[27] **D. L. Vo, H. Lee, C. Y. Yeun, K. Kim.** Enhancements of authenticated multiple key agreement protocol based on bilinear pairings. In: *Computers and Electrical Engineering*, 2010, Vol. 36, No. 1, pp. 155-159. Available at: http://dx.doi.org/10.1016/j.compeleceng.2009.08.001.

[28] **S. B. Wang, Z. F. Cao, X. L. Dong.** Provably secure Identity-based authenticated key agreement protocols in the standard model. In: *Chinese Journal of Computer*, 2007, Vol. 30, No. 10, pp. 1842-1852.

[29] **H. S. Zhou, L. Fan, J. H. Li**. Remarks on unknown key-share attack on authenticated multiple-key agreement protocol. In: *Electronics Letter*, 2003, Vol. 39, No. 17, pp. 1248–1249. Available at: http://dx.doi.org/10.1049/el:20030804.

Received March 2012.