Identity-based Authenticated Multiple Key Agreement Protocol with PKG Forward Security

Zuowen Tan
School of Information Technology, Jiangxi University of Finance and Economics,
Nanchang 330032, Jiangxi Province, P.R. China
[e-mail: tanzyw@gmail.com]

Received March 18, 2012; revised May 29, 2012; accepted July 24, 2012; published August 30, 2012

Abstract

Authenticated multiple key agreement protocols not only allow participants to agree the multiple session keys within one run of the protocol but also ensure the authenticity of the other party. In 2011, Dehkordi et al. proposed an identity-based authenticated multiple key agreement protocol. In this paper, we demonstrate that Dehkordi *et al*.'s protocol is vulnerable to impersonation attacks. Furthermore, we have found that their protocol cannot provide perfect forward security or mutual security. Then we propose an identity-based authenticated multiple key agreement protocol which removes the weaknesses of the Dehkordi *et al*.'s protocol. Compared with the multiple key agreement protocols in the literature, the proposed protocol is more efficient and holds stronger security.

Keywords: Multiple key agreement, identity-based cryptography, forward security, mutual security

This work is partially supported by the National Natural Science Foundation of China under grant No.10961013 and No.61163053, Science Research Fund of Jiangxi Province Educational Department (GJJ11418) and Natural Science Foundation of Jiangxi Province (2110GZS0047).

DOI: DOI: 10.3837/tiis.2012.08.006

1. Introduction

Two or more entities always need to establish shared secret keys. The keys are subsequently used to achieve some cryptographic goals such as confidentiality or data integrity by the communication parties. Diffie and Hellman [1] first introduced a key agreement protocol. However, since two participants do not verify the identity of each other, the Diffie-Hellman protocol suffers from the man-in-the-middle attack. Authenticated key agreement (AKA) protocols [2][3] apply a typical approach to solve the problem. AKA protocols not only allow participants to agree on the session keys but also ensure the authenticity of the other party.

Yen and Joye [3] proposed an authenticated multiple key agreement (AMKA) protocol in which two entities generate four shared keys at a time. Wu *et al.* [4] pointed that the Yen–Joye's protocol [3] is insecure against forgery attack. In 2001, Harn and Lin [5] proposed an improved protocol. But their improved protocol still suffers from forgery attack which is shown in [6]. Only three of these keys can provide perfect forward secrecy. Hwang *et al.* proposed an AKA protocol [7]. However, their protocol [7] suffers from the modification attack [8] and forgery signature attack [9]. In 2008, Lee *et al.* [10] proposed an AMKA protocol based on elliptic curve cryptography using bilinear pairings. Unfortunately, Vo *et al.* [11] showed that Lee *et al.*'s protocol is vulnerable to impersonation attack. Recently, Farash *et al.* [12] demonstrated that Lee *et al.*'s protocol is insecure against forgery attack and, if long-term private keys of two entities and one session key are revealed, the other session keys will be exposed, too. Farash *et al.* [12] also showed Vo *et al.*'s protocol [11] is vulnerable to another kind of forgery attacks and reflection attacks.

In 1984, Shamir [13] introduced the identity-based cryptography. In the identity-based cryptography, an arbitrary string (typically an identity string) such an email address can be used as a user's public key. Compared with in the certificate-based public key cryptosystem, the identity-based cryptosystem can greatly simplify the public key management. A trusted authority (private key generator, PKG) is required to derive private keys from arbitrary public keys. In 2002, Smart proposed the first identity-based authenticated key agreement (IBAKA) protocol using bilinear pairings. Many IBAKA protocols using bilinear pairings have subsequently been developed [14][15][16][17][18][19] but they are not all secure.

Some research work [16][20] defines the security attributes which an IBAKA protocol should possess. In some environment, IBAKA protocol without escrow is necessary. We highlight security attributes of IBAKA protocols without escrow against a more powerful adversary which could issue some queries.

(1) **Known-Key Secrecy**. Session keys in one run of the protocol are independent of those ones generated during other executions of the protocol. Even if an adversary has obtained the users' private key and some session keys, the adversary cannot still obtain the other session keys of the same participants.
(2) **Perfect Forward Security**. Even if an adversary has obtained secret keys of all the participants (except the PKG) and the ephemeral private key, the previously established session keys should not be leaked.
(3) **PKG Forward Security**. If an adversary has obtained the master key of the PKG and the ephemeral private keys, the previously established session keys should not be revealed.
(4) **Key-Compromise Impersonation Resilience**. 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*.

(5) **Unknown Key-Share Resilience**. If one entity, say *Alice*, believes that she shares a key with an entity, say *Bob*, but while *Bob* mistakenly believes that the key is shared with another entity, say *Cindy*, then the protocol is said to suffer from unknown-key share attack.

(6) **No Key Control**. The session keys should be determined jointly by both the communicating entities.

Remarks. PKG forward security implies perfect forward security since the compromise of the PKG's master key will lead to the compromise of the private keys of all the other participants. In fact, an IBAKA protocol holds perfect forward security (sometimes called forward security with session key escrow in the literature), which does not mean that the protocol holds PKG forward security.

So far, a few identity-based authenticated multiple key agreement (IBAMKA) protocols have been proposed [21][22][23]. The main difference between IBAMKA and IBAKA protocols is that one run of an IBAMKA protocol will produce more than one session keys instead of only one session key. Therefore, it is necessary to consider the difference of the security attributes between IBAMKA protocols and IBAKA protocols. For example, how will compromise of one or more session keys affect other session keys produced in the same run of the protocols? The stronger security of AMKA protocols has already been addressed fully in [12][24]. However, the security of IBAMKA protocols has not been discussed in detail yet. Further, how will compromise of one or more session keys affect other session keys when PKG's master key is disclosed? In the paper, we call such stronger security *Mutual Security*. This is the particular security attribute of IBAMKA protocols. We explain it in the following.

(7) **Mutual Security**. Assume that an adversary has obtained the master key of the PKG or an adversary has obtained the ephemeral private keys. While the adversary has further obtained some session keys, none of other session keys which are produced in the same run of the protocol can be derived by the adversary. Note that the adversary is not allowed to issue *RevealEphemeralKey* queries and *RevealMasterKey* queries at the same time.

Remarks. *Mutual Security* does not imply PKG forward security. *Mutual Security* means that the disclosure of PKG's master key and one or more session keys will not lead to the compromise of other session keys in the same run of the IBAMKA protocol, while PKG forward security means that the disclosure of PKG's master key will not lead to the compromise of all the previous session keys.

Recently, Dehkordi *et al*. [23] proposed an IBAMKA protocol. Their protocol has less computational cost. In this paper, we will show that Dehkordi *et al*.'s IBAMKA protocol suffers from impersonation attacks. And it also fails to provide PKG Forward Security and Mutual Security. Then we propose an improved IBAMKA protocol which removes the weaknesses of Dehkordi *et al*.'s protocol.

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, Dehkordi *et al*.' IBAMKA protocol is reviewed. In Section 4, we present its weakness. We propose an improved IBAMKA protocol in Section 5. Some cryptanalysis of the proposed IBAMKA protocol is given in Section 6. Section 7 concludes.

2. Preliminaries

2.1 Bilinear Pairings

Let G_1 and G_2 be two cyclic groups of prime order q, P be a generator of G_1. Let e be an admissible bilinear pairing from $G_1 \times G_1$ to G_2, which satisfies the following properties:

- Bilinearity: For any $U, V \in G_1$ and $a, b \in Z_p^*$, $e(aU, bV) = e(U, V)^{ab}$.
- Non-degenerate: $e(P, P) \neq 1_{G_2}$.
- Computability: There exists a probabilistic polynomial time algorithm to compute $e(U, V)$ for any $U, V \in G_1$.

2.2 Cryptographic Assumptions

Definition 1 (*Bilinear Diffie–Hellman (BDH) Problem*) Given the elements (P, aP, bP, cP) in an additive cyclic group G_1 for some unknown $a, b, c \in Z_q^*$, to compute $e(P, P)^{abc}$.

Define *the advantage of a distinguisher A against the BDH problem* as
$\mathrm{Succ}_{A,G_1}^{BDH} = \Pr[A(P, aP, bP, cP) = e(P, P)^{abc})$.

Definition 2 (*Bilinear Diffie–Hellman (BDH) Assumption*) Given (P, aP, bP, cP) for some unknown $a, b, c \in Z_q^*$, $\mathrm{Succ}_{A,G_1}^{BDH}$ of a distinguisher A which solves the BDH problem is negligible.

Definition 3 (*Decisional Bilinear Diffie–Hellman (DBDH) Problem*) Given the elements (P, aP, bP, cP) in G_1 for some unknown $a, b, c \in Z_q^*$ and $Q \in G_2$, to determine if $e(P, P)^{abc} = Q$.

Definition 4 (*Gap Bilinear Diffie–Hellman (GBDH) Assumption*) Even if there exists a probabilistic polynomial time algorithm to solve the *DBDH* problem, there is still no probabilistic polynomial time algorithms to solve the *BDH* problem.

Definition 5 (*Computational Diffie–Hellman (CDH) Problem*) Given (P, aP, bP) in G_1 for some unknown $a, b \in Z_q^*$, to compute abP.

The advantage of any probabilistic polynomial time algorithm A in solving CDH problem is defined as

$$\mathrm{Succ}_{A,G_1}^{CDH} = \Pr[A(aP, bP) = abP, a, b \in Z_q^*].$$

Definition 6 (*Computational Diffie–Hellman (CDH) Assumption*) Given (P, aP, bP) in G_1 for unknown $a, b \in Z_q^*$, $\mathrm{Succ}_{A,G_1}^{CDH}$ of any probabilistic polynomial time algorithm A in solving CDH problem is negligible.

3. Review Of Dehkordi Et Al.'s IBAMKA Scheme

In this section, we review Dehkordi *et al.*'s IBAMKA scheme [23]. Their scheme is composed of three phases: setup, key-extract and key-agreement. There are three participants, a private key generator PKG, an initiator Bob with identity ID_1 and a responder Alice with identity ID_2.

3.1. Setup Phase

Assume that $E(F_p)$ is an elliptic curve over the field F_p. PKG selects the point $P \in E(F_p)$ with order q and cyclic group G_1 of order q, $G_1=\langle P\rangle$. Let e an admissible bilinear map be: $G_1 \times G_1 \rightarrow G_2$. PKG selects two cryptographic hash functions H: $\{0,1\}^* \rightarrow G_1$, H_1 : $\{0,1\}^* \rightarrow Z_q^*$. Then PKG chooses a random value $s \in Z_q^*$ as the master key and computes the public key $P_{pub}=sP$. The system public parameters include $\{G_1, G_2, e, H_1(), H(), q, P, P_{pub}\}$.

3.2. Key-Extract Phase

For each user with identity ID, the private key generator computes $Q_{ID}=H(ID)$ as the user's public key and $S_{ID}=sQ_{ID}$ as the private key. PKG sends S_{ID} to the user with identity ID through a secure channel. The user with identity ID can verify the private key by checking if this equation holds: $e(P_{pub},Q_{ID})=e(P,S_{ID})$. Alice and Bob have their public/secret key pair (Q_2, S_2) and (Q_1, S_1), respectively.

3.3. Key-Agreement Phase

In order to agree on the session keys, Bob and Alice execute the operations:

Commitment: Bob randomly selects $c \in Z_q^*$, computes $C=c\,Q_1$ and sends C to Alice.

Challenge: Alice randomly selects $t \in Z_q^*$, computes

$$T=tQ_2,\ \bar{Y}=(t+H_1(C \| ID_1 \| ID_2))S_2,$$

and transmits (T, $\bar{Y}$) to Bob.

Response: Bob computes $f=H_1(T \| ID_2 \| ID_1), Y=(c+f)S_1$ and sends Y to Alice.

Verification: Alice calculates f and checks if the equation holds:

$$e(P,Y)=e(P_{pub},C+fQ_1). \quad (1)$$

If the above equation does not hold, Alice refuses the session request. Otherwise, Alice accepts Bob's identity.

Key agreement: Upon accepting Bob's identity, Alice computes four shared keys as follows:

$$K_1=e(S_2,C)^t, K_2=e(S_2,Q_1)K_1,$$
$$K_3=e(S_2,C)K_1, K_4=e(S_2,Q_1)^t K_1.$$

Similarly, Bob computes $f'=H_1(C \| ID_1 \| ID_2)$ and validates Alice's identity by checking if the following equation holds:

$$e(P,\bar{Y})=e(P_{pub},T+f'Q_2). \quad (2)$$

If Alice's identity is confirmed, Bob calculates four shared keys as follows:

$$K_1=e(T,S_1)^c, K_2=e(Q_2,S_1)K_1,\ K_3=e(Q_2,S_1)^c K_1, K_4=e(T,S_1)K_1.$$

4. Weaknesses Of Dehkordi Et Al.'s IBAMKA Scheme

Dehkordi *et al*. claimed that their scheme satisfied strong security. However, we found that their IBAMKA scheme suffers from impersonation attack. Moreover, their scheme cannot provide PKG Forward Security. In addition, the compromise of long-term private keys and the compromise of a session key will reveal the other three session keys of the same protocol run. In other words, their scheme lacks Mutual Security.

4.1. Impersonation Attack

During key agreement phase in [23], when Bob receives the response $(T, \overline{Y})$ to the commitment C, Bob computes $f = H_1(T \| ID_2 \| ID_1)$ and confirms Alice by verifying Eqn. (2). If Eqn. (2) holds, Bob will confirm that the user who issues challenge is the intended party Alice with identity ID_2.

Unfortunately, we found that a malicious attacker Eve could impersonate Alice to make response. Eve can easily forge the signature $\overline{Y}$ of Alice. When Bob sends the commitment C to Alice, Eve intercepts it. Then Eve randomly selects $t \in Z_q^*$ and computes $f' = H_1(C \| ID_1 \| ID_2)$, $\overline{Y} = tP_{pub}$. Next, Eve calculates $T = tP - f'Q_2$. Finally, Eve transmits $(T, \overline{Y})$ to Alice. The responses satisfy the verification equation (2). This is because

$$e(P, \overline{Y}) = e(P, tP_{pub}) = e(sP, tP)$$
$$= e(P_{pub}, T + f'Q_2).$$

It shows that the verification relation is true. Hence, Bob believes that the received message has been generated by Alice whereas it really was from the adversary. Consequently, Bob mistakenly believes that the communicating party Eve must be Alice. Thus, the adversary has mounted the impersonation attack successfully.

4.2. Lack Of PKG Forward Security

Dehkordi *et al*.'s scheme achieves perfect forward secrecy. Since long-term private keys of one or more participants (except the PKG) are disclosed, the secrecy of previous session keys established by honest participants is not affected. However, we found that their scheme cannot provide PKG forward security. When an adversary obtains the master key s of PKG, the adversary could recover four shared keys like this

$$K_1 = e(T, C)^s, K_2 = e(Q_2, S_1)K_1, K_3 = e(S_2, C_1)K_1 \text{ and } K_4 = e(T, S_1)K_1.$$

4.3. Lack Of Mutual Security

Dehkordi *et al*. claimed that their scheme has the strong security property (for details, see Theorem 14 in [23]). However, we show that there exists a potential weakness in their scheme. The compromise of long-term private keys and compromise of one session key run will lead to the compromise of the other three session keys in the same run of the protocol. The type of potential weakness for AMKA schemes is also discussed in [24]. An AMKA scheme should provide mutual security [12]. The security concept is stronger than the "strong security" mentioned in [23].

Assume that Eve has intercepted all the transmitted messages $\{C, T, \overline{Y}, Y\}$ between Alice and Bob. Suppose that Alice's long-term private key S_2 and Bob's long-term private key S_1 are compromised to an adversary Eve.

Note that four shared keys of Dehkordi *et al.*'s IBAMKA scheme can be represented as

$$K_1 = e(T, S_1)^c, K_2 = e(Q_2, S_1)K_1, \ K_3 = e(S_2, C_1)K_1, K_4 = e(T, S_1)K_1. \quad (3)$$

If one of previous session keys is revealed, say K_1, with knowledge of the private keys of participants, Eve can recover $K_2 = e(Q_2, S_1)K_1$, $K_3 = e(S_2, C_1)K_1$, $K_4 = e(T, S_1)K_1$. If other session key K_i (i=2,3,4) is disclosed, say K_2, Eve first recovers K_1 by calculating $K_1 = e(Q_2, S_1)^{-1}K_2$. Then Eve can compute $K_3 = e(S_2, C_1)K_1$ and $K_4 = e(T, S_1)K_1$.

5. The Proposed IBAMKA Scheme

In the following, we will present a new IBAMKA protocol. The proposed IBAMKA scheme is also composed of three phases: setup, key-extract and key-agreement. It is involved with a private key generator, an initiator Bob with identity ID_1 set and a responder Alice with identity ID_2.

5.1. Setup Phase

The proposed scheme has the same system parameters as in Dehkordi et al.'s IBAMKA scheme. Assume that E is an elliptic curve over the field F_p. PKG selects an elliptic curve $E(F_p)$ and one element P of order q in $E(F_p)$. Set $G_1=\langle P \rangle$. PKG selects two cryptographic hash functions H: $\{0,1\}^* \rightarrow G_1$, H_1: $\{0,1\}^* \rightarrow Z_q^*$ and an admissible bilinear map e: $G_1 \times G_1 \rightarrow G_2$. Then PKG chooses a random value $s \in Z_q^*$ as the master key and computes the public key $P_{pub} = sP$. The system public parameters consist of $\{G_1, G_2, e, H_1, H, q, P, P_{pub}\}$.

5.2. Key-extract Phase

For Bob and Alice, PKG computes $Q_1 = H(ID_1)$, $S_1 = sQ_1$, $Q_2 = H(ID_2)$ and $S_2 = sQ_2$. Therefore, Alice and Bob have their public/secret key pair (Q_2, S_2) and (Q_1, S_1), respectively.

5.3. Key-agreement Phase

To agree the session keys, Bob and Alice execute the following operations:

Commitment: Bob randomly selects $r \in Z_q^*$ and computes $R_1=rQ_1$, $R_2=rP$. Then Bob sends $\{R_1, R_2\}$ to Alice.

Challenge: Alice first validates $\{R_1,R_2\}$ by checking if the equation holds: $e(P, R_1) = e(R_2, Q_1)$. Then Alice randomly selects $t \in Z_q^*$ and computes

$$T_1=tQ_2,\ T_2=tP,\ u = H_1(T_1 \| T_2 \| ID_2 \| R_1 \| R_2 \| ID_1), Z = (t+u)S_2. \quad (4)$$

Next, Alice transmits $\{T_1, T_2, Z\}$ to Bob.

Response: Bob first computes $u' = H_1(T_1 \| T_2 \| ID_2 \| R_1 \| R_2 \| ID_1)$ and validates the message $\{T_1, T_2, Z\}$ by checking if the equations hold:

$$e(P, T_1) = e(T_2, Q_1),\ e(P, Z) = e(P_{pub}, T_1 + uQ_2). \quad (5)$$

If either of the above equations does not hold, Bob aborts. Otherwise, Bob computes

$$v = H_1(R_1 \| R_2 \| ID_1 \| T_1 \| T_2 \| ID_2), Y = (r+v)S_1. \quad (6)$$

Finally, Bob sends Y to Alice.

Verification: Alice calculates $v' = H_1(R_1 \| R_2 \| ID_1 \| T_1 \| T_2 \| ID_2)$ and confirms Bob by checking if the equation holds:

$$e(P,Y) = e(P_{pub}, R_1 + v'Q_1). \quad (7)$$

If the above equation does not hold, Alice refuses the response.

Key agreement: Alice first computes the shared secrets

$$\sigma_{A0} = tR_2, \sigma_{A00} = e(tS_2, R_1), \sigma_{A1} = e(S_2, Q_1),$$
$$\sigma_{A2} = e(tP_{pub}, R_1), \sigma_{A3} = e(tS_2, R_2), \sigma_{A4} = e(tP_{pub}, R_2),$$
$$\sigma_{A5} = e(S_2, R_2), \sigma_{A6} = e(tP_{pub}, Q_1), \sigma_{A7} = e(tS_2, Q_1), \sigma_{A8} = e(S_2, R_1).$$

Then Alice computes the session keys:

$$K_1 = H_1(\sigma_{A0}, \sigma_{A00}, \sigma_{A1}, ID_1, ID_2, T_1, T_2, R_1, R_2), \quad (8)$$
$$K_2 = H_1(\sigma_{A0}, \sigma_{A00}, \sigma_{A2}, ID_1, ID_2, T_1, T_2, R_1, R_2), \quad (9)$$
$$K_3 = H_1(\sigma_{A0}, \sigma_{A00}, \sigma_{A3}, ID_1, ID_2, T_1, T_2, R_1, R_2), \quad (10)$$
$$K_4 = H_1(\sigma_{A0}, \sigma_{A00}, \sigma_{A4}, ID_1, ID_2, T_1, T_2, R_1, R_2). \quad (11)$$

By replacing σ_{A1} in Eqn. (9) with σ_{Ai}, i=5,6,7,8, Alice can obtain other four session keys $\{K_5, K_6, K_7, K_8\}$.

Similarly, Bob computes the shared secrets

$$\sigma_{B0} = rT_2, \sigma_{B00} = e(T_1, rS_1), \sigma_{B1} = e(Q_2, S_1),$$
$$\sigma_{B2} = e(T_2, rS_1), \sigma_{B3} = e(T_1, rP_{pub}), \sigma_{B4} = e(T_2, rP_{pub}),$$
$$\sigma_{B5} = e(Q_2, rP_{pub}), \sigma_{B6} = e(T_2, S_1), \sigma_{B7} = e(T_1, S_1), \sigma_{B8} = e(Q_2, rS_1).$$

Then Bob computes the session keys:

$$K_1' = H_1(\sigma_{B0}, \sigma_{B00}, \sigma_{B1}, ID_1, ID_2, T_1, T_2, R_1, R_2), \quad (12)$$
$$K_2' = H_1(\sigma_{B0}, \sigma_{B00}, \sigma_{B2}, ID_1, ID_2, T_1, T_2, R_1, R_2), \quad (13)$$
$$K_3' = H_1(\sigma_{B0}, \sigma_{B00}, \sigma_{B3}, ID_1, ID_2, T_1, T_2, R_1, R_2), \quad (14)$$
$$K_4' = H_1(\sigma_{B0}, \sigma_{B00}, \sigma_{B4}, ID_1, ID_2, T_1, T_2, R_1, R_2). \quad (15)$$

By replacing σ_{B1} in (13) with σ_{Bi}, i=5,6,7,8, Alice can obtain other four session keys $\{ K_i' | i = i=5,6,7,8\}$.

6. Analysis On The Proposed IBAMKA Protocol

6.1 Security Model

The security models of AKA protocols are first formalized in [3]. Recently, a strong security model of AKA protocols is constructed in [25]. Here, we sketch its IBAKA version. $\prod_{I,J}^{l}$ represents the l-th session of session owner I executing the IBAMKA protocol with the session peer J. Let SID be the session identifier. If two oracle $\prod_{I,J}^{l}$ and $\prod_{J,I}^{l'}$ have the same SID, they are matching sessions. In the model, an adversary has powerful attacks.

The adversary *Eve* is a probabilistic Turing machine and controls all communications. The adversary presents parties with incoming messages via *Send*(message), thereby controlling the activation of parties. In addition, *Eve* is allowed to make the following queries:

- *RevealStaticKey*(A): *Eve* obtains *A*'s static private key. The participant *A* is called *corrupted*.
- *RevealMasterKey*: *Eve* obtains the master secret key used by PKG to generate private keys. Consequently, M can obtain private keys for all the participants.

- *RevealEphemeralKey*($\prod$): *Eve* obtains the ephemeral private key held by session $\prod$.
- *RevealSessionKey*($\prod$): If $\prod$ has completed, then *Eve* obtains the session key.
- *EstablishID*: This query allows *Eve* to register any identity chosen by *Eve* and obtain the corresponding private key from PKG.

Eve is allowed to make only one query *Test* to a *fresh* session. As a response, *Eve* is given with equal probability either the session key held by the test session or a random number. If *Eve* guesses correctly whether the given response is random, then the adversary is said to be successful. Note that after *Eve* issues the *Test* query, *Eve* can continue the above queries in an adapting way with only one condition that these queries must ensure that the test session remains fresh.

Definition 7 [25] ($\prod_{I,J}$-fresh) Let $\prod_{I,J}$ be a completed session owned by an honest party I with the honest peer J. Let $\prod^{*}_{I,J}$ be its the matching session if the matching session exists. $\prod_{I,J}$ is called fresh if none of the following conditions hold:

1. *E* issued RevealSessionKey($\prod_{I,J}$) or RevealSessionKey($\prod^{*}_{I,J}$) (if $\prod^{*}_{I,J}$ exists).
2. $\prod^{*}_{I,J}$ exists and *E* issued one of the following:
   (a) Both RevealStaticKey(I) and RevealEphemeralKey($\prod_{I,J}$).
   (b) Both RevealStaticKey(J) and RevealEphemeralKey($\prod^{*}_{I,J}$).
3. $\prod^{*}_{I,J}$ does not exist and *E* issued one of the following:
   (a) Both RevealStaticKey(I) and RevealEphemeralKey($\prod_{I,J}$).
   (b) RevealStaticKey(J).

If *Eve* can distinguish the session key of a fresh session from a randomly chosen number with a probability greater than 1/2, then *Eve* wins the game. *Eve*'s advantage *Adv*(*Eve*) is defined as the probability of winning the above game.

Definition 8. *An IBAKA protocol is secure if*

1. *In the presence of the benign adversary on* $\prod^{l}_{I,J}$ *and* $\prod^{l'}_{J,I}$, *both oracles always accept holding the same session key, and this key is distributed uniformly.*
2. *For any polynomially bounded adversary Eve, the value* |*Adv*(*Eve*)- 1/2| *in the above game is negligible.*

6.2 Security Aanalysis

In this section, we analyze the correctness and security of the proposed IBAMKA protocol.

It is straight to show that the proposed IBAMKA protocol holds the completeness property, that is, $K_i = K_i'$, *i*=1,2,3,…,8. Say, K_1, K_1'. Since

$$\sigma_{A0} = tR_2 = trP = rT_2 = \sigma_{B0}, \sigma_{A1} = e(tS_2, R_1) = e(tQ_2, sR_1) = e(T_1, rS_1) = \sigma_{B1},$$

we can have $K_1 = K_1'$.

Next, we prove the security of our IBAMKA protocol under the strong IBAKA security model. In the following, we only give the security proof of the first session key in the proposed IBAMKA protocol in Theorem 1.

Theorem 1. *The proposed IBAMKA protocol is a secure authentication key agreement protocol upon GBDH assumptions in Random Oracle Model.*

Proof: According to Definition 8, we have to verify if the proposed IBAMKA protocol satisfies two conditions. Condition 1 has already been shown to be right in the above soundness property. Now, it is our task to verify Condition 2. We will show it using proof by contradiction.

Assume that an adversary *Eve* could distinguish the session key of a fresh session from a randomly chosen number at the probabiliy (1/2+$p(k)$) where $p(k)$ is a non-negligible function of the system security parameter k. We will construct a simulator D which runs *Eve* as a subroutine to solve an instance of GBDH problems: given the elements (P , aP , bP , cP) in G_1 for unknown $a,b,c \in Z_q^*$ and a *DBDH* solver (hereinafter called *DBDH* oracle), to compute $e(P,P)^{abc}$. Here, we assume that a BDDH oracle is available to the simulator D. That is, on input any three element { xP , yP , zP } in G_1 and one element g in G_2, BDDH oracle can output **1** if $e(P,P)^{xyz} = g$ and **0** otherwise.

By E, we denote the event that *Eve* is successful. Let H be the event that *Eve* queries the hash random oracle $H_1()$ with the message $(\sigma_0, \sigma_{00}, \sigma_1, ID_i, ID_j, T_1, T_2, R_1, R_2)$ in the test session or its matching session. Since $E=(E \cap H) \cup (E \cap \overline{H})$, $(E \cap H) \cap (E \cap \overline{H}) = \Phi$, we have that $\Pr(E) = \Pr(E \cap H) + \Pr(E \cap \overline{H})$. When the event $E \cap \overline{H}$ happens, the adversary does not issue *RevealSessionKey* against the test session and its matching session. Hence, $\Pr(E \cap \overline{H}) \le \frac{1}{2}$. Therefore, we have $\Pr(E \cap H) \ge p(k)$.

Let n be the number of the users, m be the maximum of the run times of the protocol for one user. Let $\prod^t$ be the test session and $\prod^{t^*}$ be its matching session (if it exists). D randomly selects two target users U_1 with identity ID_1, U_2 with identity ID_2 and two integers i, j from $\{1,2,\ldots,m\}$. Let $\prod_1^i$ be the i-th protocol run of U_1 and $\prod_2^j$ be the j-th protocol run of U_2. D interacts with *Eve*. D simulates the responses to all the queries issued by *Eve*.

Setup: D chooses the system parameters as in the Setup phase of the proposed IBAMKA protocol. D randomly chooses s in Z_q^* as the master secret key and sets the system public key $P_{pub} = sP$.

H(ID): D maintains a list which records identity and its value. For a new ID, D returns a random number which is different from the values in the list. Otherwise, D returns the hash value in the list.

EstablishID: For *ID*, D first deals with H(ID). Then D randomly selects $t_{ID} \in Z_q^*$ and

responds with ($t_{ID}sP$, $t_{ID}P$) as the private/public key pair. Thus, U_1 and U_2 have the private/public key pair (t_1sP , t_1P) and (t_2sP , t_2P), respectively.

H_1( ): For hash-query about $(\sigma_0, \sigma_{00}, \sigma_1, ID_i, ID_j, T_1, T_2, R_1, R_2)$, D uses the DBDH oracle to check if the second and third shared secrets σ_{00}, σ_1 are consistent with ephemeral public keys and system public keys. D uses the DDH oracle to check if the first shared secret share σ_0 is consistent with ephemeral public keys.

When *Eve* queries *Send, RevealStaticKey*, *RevealMasterKey*, *RevealEphemeralKey* and *RevealSessionKey*, if neither $\prod_1^i$ nor $\prod_2^j$ is involved, D responds to these queries as in the actual protocols. If $\prod_1^i$ or $\prod_2^j$ is involved, D makes responses like this:

(1) D randomly selects $t_0, r_0 \in Z_q^*$ and responds with t_0P, r_0P to hash-queries about ID_1 and ID respectively.

(2) D responds with (t_1aP, aP) to the ephemeral public key query about $\prod_1^i$.

(3) D responds with (t_2bP, bP) to the ephemeral public key query about $\prod_2^j$.

(4) If either $\prod_1^i$ or $\prod_2^j$ is involved, say $\prod^{1,i}$ (suppose U_1 is an initiator), when *Eve* issues *RevealSessionKey query*, D first computes the shared secrets

$$\sigma_0 = t_{ID}(aP),\ \sigma_1 = e(st_{ID}(aP), t_{ID}t_{ID0}P),$$

where ID is the identity of the peer and $(t_{ID}t_{ID0}P, t_{ID0}P)$ is the ephemeral public key query about the matching session of $\prod_1^i$.

Then D responds to the hash-query about $(\sigma_0, \sigma_1, ID_1, ID, t_0(aP), aP, t_{ID}t_{ID0}P, t_{ID0}P)$.

(5) If *Eve* queries for the ephemeral private key of either $\prod_1^i$ or $\prod_2^j$, then D aborts.

Let **Fail** be the event that D will fail if the test session and its matching session are not $\prod_1^i$ or $\prod_2^j$. If **Fail** has never happened, for the key derivation Hash-query involving $\prod_1^i$ and $\prod_2^j$, D uses the DBDH and DDH oracles to check if two shared secrets are consistent with ephemeral public keys. If both DBDH and DDH oracles return 1, D computes $e(\sigma_0, (cP))$ as the solution to the GBDH problem.

Since $\Pr(\overline{\text{Fail}}) \geq \frac{1}{n^2m^2}$, the probability of solving the GBDH problem is at least

$\Pr(\overline{\text{Fail}}) \geq \frac{1}{n^2m^2}\Pr(E \cap H)$. □

Now, we show that the proposed scheme satisfies the security requirements listed in Section 1.

Theorem 2. *The proposed IBAMKA scheme has PKG Forward Security.*

Proof: Assume that an adversary Eve has obtained master key s after Eve has issued *RevealMasterKey* queries. Then the secret keys $\{S_1, S_2\}$ will be compromised. The adversary Eve is allowed to issue *RevealEphemeralKey* queries about the ephemeral private keys of the other runs but the target run. Eve has intercepted all the messages $\{T_1, T_2, Z, R_1, R_2, Y\}$ transmitted between the users Alice and Bob in the target run of the protocol. However, the knowledge of $\{S_1, S_2\}$, $\{T_1, T_2, Z, R_1, R_2, Y\}$ and other ephemeral private keys $\{t', r'\}$ cannot help Eve to obtain the ephemeral private key $\{t, r\}$ corresponding to $\{T_1, T_2, R_1, R_2\}$. Without lost of generality, assume that the adversary attempts to learn the used session keys,

say, K_1. Since the hash function is one-way and cryptographically secure, in order to compute K_1, Eve has to recover $(\sigma_{A0}, \sigma_{A00}, \sigma_{A1})$ or $(\sigma_{B0}, \sigma_{B00}, \sigma_{B1})$. With knowledge of the secret keys $\{S_1, S_2, s\}$, the adversary could compute $\sigma_{B1} = e(Q_2, S_1)$ or $\sigma_{A1} = e(S_2, Q_1)$ and $\sigma_{A00} = e(T_1, R_1)^s = \sigma_{A00}$. But when the adversary tries to obtain $\sigma_{A0} = tR_2$ or $\sigma_{B0} = rT_2$, he/she has to be faced with a CDH problem $\{T_2, R_2\}$. If the adversary could compute the session keys, the CDH problem $\{T_2, R_2\}$ would be solved. This is a contradiction with the CDH assumptions.

Therefore, even if PKG's master key is disclosed, the previous session keys cannot be revealed. The proposed IBAMKA scheme has Perfect Forward Security. □

Theorem 3. *The proposed IBAMKA scheme has Known-Key Secrecy.*

Proof: Assume that an adversary is allowed to issue *RevealStaticKey* queries about Alice and Bob. The adversary obtains their private key $\{S_1, S_2\}$. If the adversary issues *RevealSessionKey* queries, some session keys are comprised to the adversary. However, as the ephemeral private keys $\{r, t\}$ change, the agreed session keys in each run of the protocol will change. Take K_1 for example. Since $K_1 = H_1(\sigma_{A0}, \sigma_{A00}, \sigma_{A1}, ID_1, ID_2, T_1, T_2, R_1, R_2)$, $\sigma_{A0} = trP$, $\sigma_{A00} = e(T_1, rS_1)$ and $\sigma_{A1} = e(S_2, Q_1)$, the session key in one run of the protocol cannot help an adversary to compute $(\sigma_{A0}', \sigma_{A00}', \sigma_{A1}')$ in different runs of the protocol. Furthermore, even if the adversary obtained the master key (further the users' private keys) and one session key K_1, the adversary cannot still compute $\sigma_{A0}' = t'r'P$ and further compute another session key K_1'. From the similar analysis in Theorem 2, when the adversary mounts the known-key attacks, he/she will have to be faced with a new instance of the CDH problem.

Hence, the proposed IBAMKA scheme achieves Known-Key Secrecy**.** □

Theorem 1 and Theorem 3 imply the following results.

Theorem 4. *The proposed IBAMKA scheme has Key-Compromise Impersonation Resilience.*

Theorem 5. *The proposed IBAMKA scheme has Unknown Key-Share Resilience.*

Theorem 6. *The proposed IBAMKA scheme has No Key Control.*

Proof: In the proposed protocol, each session key is derived from $\{\sigma_{A0}, \sigma_{A00}\}$ (or $\{\sigma_{B0}, \sigma_{B00}\}$), σ_{Ai} (i=1,2,3,…,8) (or σ_{Bi}) and the ephemeral keys $\{T_1, T_2, R_1, R_2\}$. Since $\sigma_{A0} = tR_2 = rT_2 = \sigma_{B0}$ and $\sigma_{A00} = e(T_1, R_1)^s = \sigma_{B00}$, Alice(Bob) cannot compute $r(t)$ from the ephemeral public keys upon the CDH assumptions. Thus, each session key is determined cooperatively by Alice and Bob. □

Theorem 7. *The proposed IBAMKA scheme achieves Mutual Security.*

Proof: When we address Mutual Security of *IBAMKA* schemes, the adversary *Eve* is allowed to issue one of the following queries about the target run of the protocol: *RevealEphemeralKey* query and *RevealMasterKey* query. The assumption is reasonable. If the adversary has issued both the queries, all the shared secrets can be derived and all the session keys can also be computed from Eqn.(8)-(11) or Eqn.(12)-(15). But the adversary can issue both the queries about the other runs of the protocol.

(1) Assume that *Eve* has obtained some session keys $K_i, i \in [1,8]$ after *Eve* has issued *RevealEphemeralKey* queries.

Since session keys $K_i, i \in [1,8]$ are generated by using secure hash function $H_1()$, even if session keys are comprised, due to the onewayness of the hash function, *Eve* is still unable to

recover their pre-image $(\sigma_{B0}, \sigma_{B00}, \sigma_{Bi})$ or $(\sigma_{A0}, \sigma_{A00}, \sigma_{Ai})$ of $K_i, i \in [1,8]$ of the hash functions. When *Eve* attempts to obtain other session keys $K_j, j \in [1,8], j \neq i$, *Eve* has to compute $(\sigma_{B0}, \sigma_{B00})$ or $(\sigma_{A0}, \sigma_{A00})$ from the ephemeral keys $\{r,t\}$ which *Eve* has obtained by issuing *RevealEphemeralKey* queries. With these ephemeral keys, *Eve* can calculate σ_{A0} and σ_{B0}. Since $\sigma_{A00} = e(tS_2, R_1)$ and $\sigma_{B00} = e(T_1, rS_1)$, *Eve* cannot still work out σ_{A00} or σ_{B00} without the private key S_1 or S_2.

(2) Assume that *Eve* has issued *RevealMasterKey* queries and obtained some session keys $K_i, i \in [1,8]$.

None of $(\sigma_{B0}, \sigma_{B00}, \sigma_{Bi})$ and $(\sigma_{A0}, \sigma_{A00}, \sigma_{Ai})$ can be derived since they are protected by hash functions. *Eve* uses the master key to compute $\sigma_{A00} = e(sT_1, R_1) = \sigma_{B00}$, σ_{Aj} and σ_{Bj} ($j \neq 0$). However, when *Eve* attempts to compute $\sigma_{A0} = tR_2$ or $\sigma_{B0} = rT_2$ from $\{T_2, R_2\}$, *Eve* will be faced with an instance of the CDH problem. Therefore, *Eve* cannot obtain any session key $K_j, j \neq i$. □

6.3 Performance Comparison

Some identity-based key agreement protocols [14][16][17] cannot provide authentication function. The users confirm their session keys by the succedent communication. Moreover,

Table 1. Performance comparison

	[7]	[10]	[11]	[5]	[23]	**Ours**
Ephemeral key	$2T_e$	$2T_S$	$2T_S$	$2T_e$	$2T_S$ ($1T_S$)	$2T_S$
Authentication part	T_m	$T_A+2T_S+2T_m$	$T_A+2T_S+2T_m$	$3T_m+T_e$	$1T_S$	T_S
Verification	$2T_e+T_m$	$T_A+2T_S+T_m+3T_p$	$T_A+2T_S+3T_p$	$2T_m+3T_e$	$2T_p$	$4T_p(2T_p)+T_S$
Generation of one key	T_e	$T_A+T_S+T_p$	$T_A+3T_S+T_p$	T_e	T_p+T_E	$2T_S+2T_p$
Generation of four keys	$4T_e$	$4T_A+4T_S+4T_p$	$4T_A+8T_S+4T_p$	$4T_e$	$4T_p+2T_E+3T_M$	$2T_S+5T_p$
Total time	$8T_e+2T_m$	$6T_A+10T_S+3T_m+7T_p$	$6T_A+14T_S+2T_m+7T_p$	$10T_e+5T_m$	$3T_S(2T_S)+6T_p+2T_E+3T_M$	$5T_S+9T_p(7T_p)$
Identity-based	N	N	N	N	Yes	Yes
Number of randon number	2	2	2	2	1	1
Number of key agreement	4	4	4	4	4	8

Possible attacks	modification attack, forgery signature attack	impersonation attack	reflection attack, forgery attack	impersonation attack, forgery attack	impersonation attack	immune
(PKG) Forward security	N	N	N	N	N	Yes
Mutual security	N	N	N	N	N	Yes

**Note that N means "not support".*

these protocols [14][16][17] only can produce one session key at a time. We compare our IBAMKA scheme with the previous AMKE protocols [5][7][10][11][23] in term of computational efficiency and security property.

In the AMKE protocols [5][7][10][11], the number of randomly produced values by each user is 2. But in Dehkordi *et al*.'s scheme [23] and the proposed IBAMKA scheme, each user produces only one random value. In contrast with four session keys produced in one run of Dehkordi *et al*.'s protocol, the proposed protocol generates eight session keys. Next, we evaluate the efficiency performance of the proposed IBAMKA protocol and make comparison with some AMKE protocols. Let T_S,T_A,T_p,T_M,T_E,T_m and T_e represent 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, one exponent computation in G_2,one modular multiplication in Z_q^* and one modular exponent in Z_q^*, respectively. Because the addition in Z_q^* and hash operations require few computations, we neglect their computational costs. The time of different phases consumed by the session initiator or the session responder is listed in **Table 1**.

In Dehkordi et al.'s scheme, the initiator Bob needs two scalar multiplications in G_1 to compute the ephemeral keys while the responder Alice needs one scalar multiplication in G_1 to compute ephemeral keys. During the verication in our IBAMKA scheme, the initiator Bob needs four bilinear pairing computation in G_2 and one scalar multiplication in G_1 while the responder Alice requires two bilinear pairing computation in G_2 and one scalar multiplication in G_1. Compared with Dehkordi et al.'s IBAMKA scheme, our IBAMKA scheme needs roughly the same or even less computation cost. Compared with the AMKE protocol in [7], the proposed protocol requires more computation costs. However, the proposed protocol is an identity-based AMKE protocol which removes the management of public keys in the PKI. Moreover, the proposed protocol requires less communication cost.

As shown above, the proposed protocol satisfies stronger security properties and can resist all possible attacks. The AMKA schemes in [5][7][10][11][23] suffer from some attacks such as impersonation attacks, forgery attacks, etc. Furthermore, The AMKA schemes in [5][7][10][11] cannot provide perfect Forward Security and Mutual Security. Our analysis in Section 4 shows that the IBAMKA scheme in [23] cannot provide Forward Security and Mutual Security. The security properties between our proposed scheme and other related schemes are summarized in **Table 1**.

7. Conclusion

It is essential to establish multiple session keys within one run of key agreement protocol between both the communication parties. Dehkordi *et al*. used bilinear pairings to propose an identity-based authenticated multiple key agreement protocol [23]. We have pointed out that

Dehkordi *et al*.'s IBAMKA protocol suffers from the impersonation attack and lacks the PKG forward security and mutual security. To eliminate these security vulnerabilities, we propose an improved IBAMKA protocol, which successfully avoids the weaknesses existed in the original Dehkordi *et al*.'s protocol.

References

[1] W. Diffie, M.E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22,no. 6, pp. 644-654, 1976. Article (CrossRef Link).

[2] M. Bellare, D. Pointcheval and P. Rogaway, "Authenticated key agreement secure against dictionary attacks," *Advances in Cryptology-Eurocrypt'00*, Lecture Notes in Computer Science, vol. 1807, pp. 139-155, 2000. Article (CrossRef Link).

[3] S.M. Yen and M. Joye, "Improved authenticated multiple-key agreement protocol," *Electron Letter*,vol. 34, no. 18, pp. 1738–1739,1998. Article (CrossRef Link).

[4] T.S. Wu, W.H. He, C.L. Hsu, "Security of authenticated multiple-key,"*Electron Letter*, vol.35, no.5, pp.391–392,1999. Article (CrossRef Link).

[5] L. Harn, H.Y. Lin, "Authenticated key agreement without using one-way hash function," *Electron Letter*, vol. 37, no. 10, pp. 629-630, 2001. Article (CrossRef Link).

[6] H.S. Zhou, L. Fan and J.H. Li, "Remarks on unknown key-share attack on authenticated multiple-key agreement protocol,"*Electronic Letter*, vol. 39, no. 17, pp. 1248–1249, 2003. Article (CrossRef Link).

[7] R.J. Hwang, S.H. Shiau and C.H. Lai, "An enhanced authentication key agreement protocol," In *Proc. of the 17th international conference on AINA*, pp. 20-25, Mar. 2003.
Article (CrossRef Link).

[9] N.Y. Lee and C.N. Wu, "Improved authentication key agreement protocol without using one-way hash function," *ACM Operat Syst Rev*, vol. 38, no. 2, pp. 85-92,2004. Article (CrossRef Link).

[9] M.S. Hwang, T.Y. Chang, S.C. Lin and C.S. Tsai, "On the security of an enhanced authentication key agreement protocol," in *18th International Conference on Advanced Information Networking and Applications*, vol. 2, pp. 160-163, 2004. Article (CrossRef Link).

[10] N.Y. Lee, C.N. Wu and C.C. Wang, "Authenticated multiple key agreement protocols based on elliptic curves and bilinear pairings," *Computers and Electrical Engineering*, vol.34, no.1, pp. 12–20, 2008. Article (CrossRef Link).

[11] D.L. Vo, H. Lee, C.Y. Yeun and K. Kim, "Enhancements of authenticated multiple key exchange protocol based on pairings," *Computers and Electrical Engineering*, vol. 36, no. 1, pp. 155–159, 2010. Article (CrossRef Link).

[12] M.S. Farash, M. Bayat, M.A. Attari, "Vulnerability of two multiple-key agreement protocols," *Computers and Electrical Engineering*, vol. 37, no. 2, pp. 199–204, 2011. Article (CrossRef Link).

[13] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology-CRYPTO* '84, Springer-Verlag, Lecture Notes in Computer Science, vol. 196, pp. 47-53, 1984. Article (CrossRef Link).

[14] N.P. Smart, "An identity based authenticated key agreement protocol based on the Weil bilinear pairing," *Electronics Letters*, vol. 38,no. 13, pp. 630-632, 2002. Article (CrossRef Link).

[15] K. Shim, "Efficient ID-based authenticated key agreement protocol based on the Weil bilinear pairing", *Electronics Letters*, vol. 39, no. 8, pp. 653-654, 2003. Article (CrossRef Link).

[16] L. Chen, C. Kudla, "Identity based key agreement protocols from pairings," in *Proc. of the 16-th IEEE Computer Security Foundations Workshop*, IEEE Computer Society, pp. 219-233, 2002. Article (CrossRef Link).

[17] S.B. Wang, Z.F. Cao, X.L. Dong, "Provably secure Identity-based authenticated key agreement protocols in the standard model,"*Chinese Journal of Computer*, vol.30, no. 10, pp. 1842-1852, 2007. Article (CrossRef Link).

[18] 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*, Lecture Notes in Computer Science, vol. 5107, pp. 69–83, 2008. Article (CrossRef Link).

[19] K.-K.R. Choo, S.S.M. Chow, "Strongly-secure identity-based key agreement and anonymous extension," in *Information Security*, Lecture Notes in Computer Science, vol. 4779, pp. 203–220, Springer, Berlin, 2007. Article (CrossRef Link).

[20] S.B. Wilson, A. Menezes, "Authenticated Diffie-Hellman key agreement protocols," in *Proc. of the SAC' 98*, Lecture Notes in Computer Science, vol. 1556, pp. 339-361, 1999. Article (CrossRef Link)

[21] Z.W. Tan, "Efficient identity-based authenticated multiple key exchange protocol,"*Computers and Electrical Engineering*, vol. 37, no. 37, pp. 191–198, 2011. Article (CrossRef Link).

[22] K.W. Kim, E.K. Ryu and K.Y. Yoo, "ID-Based authenticated multiple-key agreement protocol from pairings," In *ICCSA 2004*, Lecture Notes in Computer Science,vol. 3046, pp. 672–680, 2004. Article (CrossRef Link).

[23] M.H. Dehkordi and R. Alimoradi, "Identity-based multiple key agreement scheme," *KSII Transactions on Internet and Information Systems*, vol. 5, no. 2, pp. 2392-2402, 2011. Article (CrossRef Link).

[24] K.A. Shim, "Vulnerabilities of generalized MQV key agreement protocol without using one-way hash functions," *Computer Standards & Interfaces*, vol. 29, no. 4, pp. 467-470, 2007. Article (CrossRef Link).

[25] B. Ustaoglu, "Integrating identity-based and certificate-based authenticated key exchange protocols," *International Journal of Information Security*, vol. 10, no. 4, pp. 201–212, 2011. Article (CrossRef Link).

Zuowen Tan got his Ph.D. degree in Applied Mathematics from Institute of Systems Science, Academy of Mathematics and System Science, CAS in 2005. Since 2006, he has been with Department of Computer Science & Technology, School of Information Technology, Jiangxi University of Finance & Economics, China. His research interests include e-commerce security, information security and cryptography.