An ID-Based Key Agreement Scheme from pairing

Guohong Xie
School of Mathematical Sciences,
Peking University,People's Republic of China
xgh@pku.edu.cn

Abstract:

We propose a two-party identity-based key agreement and key agreement with confirmation from pairing that can be used with escrow mode or without escrow mode.,It can also be used between users of different KGC (Key Generation Centre).We will show that the our scheme is a secure key agreement in the mode proposed by Bellare and Rogaway[2],and we will show that the scheme has the attribute of Known Key-Security ,Key-Compromise-Impersonation, Unknown-Key-Share, Perfect-Forward-Secrecy and Key-Control.

Keywords: authenticated key agreement, identity-based cryptography, Tate pairing.

1 Introduction

In this paper we propose a two-party authenticated identity-based key agreement from pairings. The basic idea of an identity-based cryptosystem is that end users can choose an arbitrary string, for example email addresses or other online identifiers, as their public key. This eliminates much of the overhead associated with key management. In traditional PKI settings, key agreement protocols relies on the parties obtaining each other's certificates, extracting each other's public keys, checking certificate chains (which may involve many signature verifications) and finally generating a shared secret. The technique of identity-based encryption (IBE) greatly simplifies this process. This idea was first proposed by Shamir [3] in 1984, made viable by Cocks [4] and Boneh and Frankl- in [5] in 2001, further streamlined by Sakai and Kasahara [6] in 2003, and is currently an area of very active research.

There are many key agreement protocols based on bilinear maps, and most have subsequently been broken. One of the first applications of pairing based cryptography was a tripartite key agreement protocol by Joux [7]. Although this key agreement does not authenticate the users, and thus is susceptible to the man-in-the-middle attack, it was a significant step in the development of pairing based cryptography. This original scheme was not identity-based. Many key agreements from bilinear maps have been since proposed. Scott [8], Smart [9], and Chen and Kudla [10]have proposed two-party key agreement protocols, none of which have been broken. All of these schemes require that all parties involved in the key agreement are clients of the same Key Generation Centre (KGC). Noel McCullagh and Paulo S. L. M[1] have propsed a two-party identity-based key agreement scheme which is affected by Key-Compromise Impersonation [11].

Most identity-based key agreement protocols have the property of key escrow: the trusted authority that issues private keys can recover the agreed session key.This feature is either acceptable, unacceptable, or desired depending on the circumstances. For example, escrow is essential in situations where confidentiality as well as an audit trail is a legal requirement, as in confidential communication in the health care profession. There are other examples, such as personal communications, where it would be advantageous to turn escrow off.

Most key agreements require both parties to be from the same domain. This, for example, might mean that two workers from the same company would be able to generate a shared secret, however employees from two different companies would not be able to generate such a shared secret. We suggest key agreement between domains is an important property of this scheme as, from a commercial viewpoint, identity-based cryptography (IBC) seems particularly well suited to encrypted telephony and encrypted VoIP. For encrypted VoIP to work on a global scale there simply must be compatibility between networks, and therefore key agreement between different networks is important.

Our contributions in this paper are:

-An efficient identity-based authenticated key agreement protocol that can be instantiated in either escrowed or escrowless mode without imposing extra computational steps.

-An efficient key agreement that allows users who have their private keys generated by distinct Key Generation Centres to establish a shared secret without additional overhead, provided standardised curve parameters are used.

-An efficient identity-based authenticated key agreement protocol with or without confirmation that can be instantiated in either escrowed or escrowless mode and that allows users who have their private keys generated by distinct Key Generation Centres to establish a shared secret without additional overhead, provided standardised curve parameters are used.

This paper is organised as follows. Section 2 introduces Technical Backgrounds. Section 3 describes our proposed authenticated key agreement with escrow, and section 4 introduces our proposed scheme without escrow., section 5 we present a key agreement protocol for members of distinct key generation domains, Section 6 describes our proposed authenticated key agreement with confirmation.we discuss the security issues in section 7 and draw our conclusions in section 8.

2 Technical Backgrounds

2.1 Bilinear Pairing

In this section, we shall briefly describe the properties of the bilinear pairings. The bilinear parings were used in negative meaning in cryptography, because the MOV attack using Weil pairing and FR attack using Tate pairing reduce the discrete logarithm problem on some elliptic curves to the discrete logarithm problem in a finite field[12,13]. Recently, the bilinear pairings have been used in positive meaning in cryptography, because they are important tools for construction of the identity-based schemes [5,7,14,15,16].

We let $G1$ be a cyclic additive group generated by P, whose order is a prime q, and $G2$ be a cyclic multiplicative group of the same order q. We assume that the discrete logarithm problem(DLP) in both $G1$ and $G2$ are hard. We let $\hat{e} : G1 \times G1 \rightarrow G2$ be a pairing which satisfies the following properties:

1. Bilinear

$\hat{e}(P1 + P2, Q) = \hat{e}(P1, Q) \cdot \hat{e}(P2, Q)$
$\hat{e}(P, Q1 + Q2) = \hat{e}(P, Q1) \cdot \hat{e}(P, Q2)$

i..e., $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ where $a, b \in Z^*q$, $P, Q \in G1$.

2. Non-degenerate
   There exists $P \in G1$ such that $\hat{e}(P, P) \neq 1$.

3. Computability
   There is an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in G1$.

The non-degeneracy does not hold for the standard Weil pairing e(P, Q), but it does hold for the modified Weil pairing ê(P, Q). We note that the Weil and Tate pairings associated with supersingular elliptic curves or abelian varieties can be modified to create such bilinear maps. We refer to [5,14] for more details.

3 An Authenticated Key Agreement With Escrow

As with all other identity-based cryptosystems we assume the existence of a trusted Key Generation Centre (KGC) that is responsible for the creation and secure distribution of users private keys. This agreement algorithm can be implemented using the modified Tate pairing.

Setup: The KGC inputs a security parameter *k* into a BDH parameter generator *B*mt which returns two groups G1and G2, both of prime order q, suitable bilinear map ê: $G1 \times G1 \rightarrow G2$ (which can be implemented as the modified Tate pairing), a generator element P such that <P> = G1 and a random oracle H,which is as H : {0, 1}* → Z_q^* . The KGC randomly generates a master secret s $\in_R$Zq*, and calculates a master public key sP. The parameters and master public key are distributed to the users of the system through a secure authenticated channel. We assume that the number of users is polynomial in *k*.

Extract: The KGC checks that a user has a claim to a particular online identifier. If they do, the KGC generates their private key and communicates it privately to them. Let Alice's online identifier map to a ∈ Zq* y means of the random oracle H. Alice's public key is (a+s)P, which can be computed as aP + sP. The KGC computes Alice's private key as Apri = $(a + s)^{-1}$P.While it may be argued that this key pair derivation is not as elegant as that in the Boneh-Franklin IBE[5].since the public key no longer relies on the user's identity alone, most key agreements, except Scott's and Ryu et al.'s [14],lso use the KGC's master secret in the key agreement stage.

Key Agreement: Assume that Alice and Bob have private keys issued by the same KGC, respectively Apri and Bpri. Alice and Bob each generate one unique random nonce x, y$\in_R$ Zq , respectively.

Alice Bob

$A_{KA} = x(bP + sP) \quad \longrightarrow$

$\longleftarrow \quad B_{KA} = y(aP + sP)$

$\text{key}_A = \hat{e}(B_{KA},Apri)^{(x+1)}\ \hat{e}(P,P)^x \qquad \text{key}_B = \hat{e}(A_{KA},Bpri)^{(y+1)}\ \hat{e}(P,P)^y$

This scheme is consistent because:

$$\begin{aligned}\text{key}_A &= \hat{e}(B_{KA},Apri)^{(x+1)}\ \hat{e}(P,P)^x = \hat{e}(P,P)^{y(a+s)(a+s)^{-1}(x+1)+x}\\ &= \hat{e}(P,P)^{xy+x+y} = \hat{e}(P,P)^{x(y+1)+y}\\ &= \hat{e}(P,P)^{x(b+s)(b+s)^{-1}(y+1)+y}\\ &= \hat{e}(A_{KA},Bpri)^{(y+1)}\ \hat{e}(P,P)^y\\ &= \text{key}_B\end{aligned}$$

The escrow property derives from the KGC's ability to recover the shared session key by computing:

$$xP = (s + b)^{-1}A_{KA},$$
$$yP = (s + a)^{-1}B_{KA},$$
$$\text{key} = \hat{e}(xP, yP)\ \hat{e}(xP, P)\ \hat{e}(P, yP).$$

Our scheme is role symmetric, with each party performing the same operations and thus incurring the same computational cost.

4 An Authenticated Key Agreement Without Escrow

The key agreement without escrow differs only slightly from the algorithm given in section 3. Again there are three algorithms, Setup, Extract and Key Agreement.

Setup: T Setup: The KGC inputs a security parameter *k* into a BDH parameter generator *B*mt which returns two groups G1and G2, both of prime order q, a suitable bilinear map ê:$G1 \times G1 \rightarrow G2$ (which can be implemented as the modified Tate pairing), a generator element P such that <P> = G0, and a random oracle H,which is as H : $\{0, 1\}^* \rightarrow Z_q^*$ and a random oracle H1,which is as H1 :G0 → G0, Q=H1(P) . H1 satisfies the condition that the KGC can't find a value λ such that H1(P) = λ Q, The KGC randomly generates a master secret s $\in$R Zq*, and calculates a master public key sP. The parameters and master public key are distributed to the users of the system through a secure authenticated channel. We assume that the number of users is polynomial in *k*.

Extract: The KGC checks that a user has a claim to a particular online identifier. If they do, the KGC generates their private key and communicates it privately to them. Let Alice's online identifier map to a $\in$Zq* y means of the random oracle H. Alice's public key is Apub = (a + s)P, which can be computed as aP +sP. Alice's private key is generated as Apri = $(a+s)^{-1}Q$. End user Alice is encouraged to check that the KGC has used the correct Q by checking whether H1(P) equals to Q.

Key Agreement: Assume that Alice and Bob have private keys issued by the same KGC, respectively Apri and Bpri. Alice and Bob each generate one unique random nonce x, $y \in_R$ Zq*, respectively.

$$\begin{array}{lll} \text{Alice} & & \text{Bob} \\ A_{KA} = x(bP + sP) & \longrightarrow & \\ & \longleftarrow & B_{KA} = y(aP + sP) \end{array}$$

$$key_A = \hat{e}(B_{KA},Apri)^{(x+1)}\,\hat{e}(P,Q)^x \qquad key_B = \hat{e}(A_{KA},Bpri)^{(y+1)}\,\hat{e}(P,Q)^y$$

This scheme is consistent because:

$$\begin{aligned} key_A &= \hat{e}(B_{KA},Apri)^{(x+1)}\,\hat{e}(P,Q)^x = \hat{e}(P,Q)^{y(a+s)(a+s)^{-1}(x+1)+x} \\ &= \hat{e}(P,Q)^{xy+x+y} = \hat{e}(P,Q)^{x(y+1)+y} \\ &= \hat{e}(P,Q)^{x(b+s)(b+s)^{-1}(y+1)+y} \\ &= \hat{e}(A_{KA},Bpri)^{(y+1)}\,\hat{e}(P,Q)^y \\ &= key_B \end{aligned}$$

Athough the KGC has the ability to generate the private keys of both users in the protocol,and The KGC is able to compute: xP = $(s + b)^{-1}A_{KA}$,yP = $(s + a)^{-1}B_{KA}$.Thus the KGC can compute ê(xP, Q)= $\hat{e}(P, Q)^x$, ê(yP, Q)= $\hat{e}(P, Q)^y$,and he can get $\hat{e}(P, P)^{xy}$ by computing ê(xP, yP).By the definition of H1,the KGC can't get $\hat{e}(P, Q)^{xy}$ from $\hat{e}(P, P)^{xy}$.If the KGC want to get$\hat{e}(P, Q)^{xy}$ from $\hat{e}(P, Q)^x$ and $\hat{e}(P, Q)^y$,he must solve the Computational Diffie-Hellman Problem (CDHP) over the group G1[18]

5 Key Agreement Between Members of Distinct Domains

We now look at key agreements between members of separate domains. This idea was first suggested in [10].We suggest a scheme that is twice as efficient as their scheme without precomputation, whilst being similar with precomputation. Again this protocol can be instantiated in escrowed or escrowless mode. For key agreement to be possible between members of different groups all that is needed is the points P in the case of the escrow mode and Q in the case of the escrowless mode，and the curve description to be the same (standardised). Elliptic curves, suitable group generator points and other cryptographic tools have been standardised for non-IBE applications, for example in the NIST FIPS standards. It is reasonable, therefore, to assume the availability of standard pairing-friendly curves as well. Once these group generator points and curves have been agreed upon, each KGC can generate their own random master secret. Alice's private key is generated by KGC_1 with a master secret s_1. Bob's private key is generated by KGC_2 with a master secret s_2. Alice's public key is $(a + s_1)P$ and her private key is $A_{pri} = (a + s_1)^{-1}P$. Likewise, Bob's public key is $(b + s_2)P$ and his private key is $B_{pri} = (b + s_2)^{-1}P$. Notice that now Alice must obtain s_2P (the master public key of Bob's KGC) and vice-versa; it is critical that the master public keys are obtained in an authenticated manner, as with any IBC scheme.Alice and Bob now perform the authenticated key agreement:

Alice **Bob**

$A_{KA} = x(bP + s2P) \longrightarrow$

$\longleftarrow B_{KA} = y(aP + s1P)$

$key_A = ê(B_{KA},Apri)^{(x+1)}\, ê(P,P)^{x}$ $key_B = ê(A_{KA},Bpri)^{(y+1)}\, ê(P,P)^{y}$

This scheme is consistent because:

$$\begin{aligned}key_A &= ê(B_{KA},Apri)^{(x+1)}\, ê(P,P)^{x}=ê(P,P)^{y(a+s1)(a+s1)^{-1}(x+1)+x}\\ &= ê(P,P)^{xy+x+y}=ê(P,P)^{x(y+1)+y}\\ &=ê(P,P)^{x(b+s2)(b+s2)^{-1}(y+1)+y}\\ &=ê(A_{KA},Bpri)^{(y+1)}\, ê(P,P)^{y}\\ &=key_B\end{aligned}$$

Or in the mode without escrow as below:

$$key_A = ê(B_{KA},Apri)^{(x+1)}\, ê(P,Q)^{x} \qquad key_B = ê(A_{KA},Bpri)^{(y+1)}\, ê(P,Q)^{y}$$

$$\begin{aligned}key_A &= ê(B_{KA},Apri)^{(x+1)}\, ê(P,Q)^{x}=ê(P,Q)^{y(a+s1)(a+s1)^{-1}(x+1)+x}\\ &= ê(P,Q)^{xy+x+y}=ê(P,Q)^{x(y+1)+y}\\ &=ê(P,Q)^{x(b+s2)(b+s2)^{-1}(y+1)+y}\\ &=ê(A_{KA},Bpri)^{(y+1)}\, ê(P,Q)^{y}\\ &=key_B\end{aligned}$$

6 . Key agreement with confirmation

Assume H1 is a random oracle ,We can use the key agreement with confirmation in escrow mode as below:

Alice **Bob**

$A_{KA} = x(bP + sP) \quad \longrightarrow$

$\longleftarrow \quad B_{KA}=y(aP+sP)||H(A_{KA}||\hat{e}(P,P)^{x})$

$H(B_{KA}||\hat{e}(P,P)^{y}) \quad \longrightarrow$

$key_{A} = \hat{e}(B_{KA},Apri)^{(x+1)}\ \hat{e}(P,P)^{x} \qquad key_{B} = \hat{e}(A_{KA},Bpri)^{(y+1)}\ \hat{e}(P,P)^{y}$

This scheme is consistent because:

$$\begin{aligned} key_{A} &= \hat{e}(B_{KA},Apri)^{(x+1)}\ \hat{e}(P,P)^{x}=\hat{e}(P,P)^{y(a+s)(a+s)^{-1}(x+1)+x} \\ &= \hat{e}(P,P)^{xy+x+y}=\hat{e}(P,P)^{x(y+1)+y} \\ &=\hat{e}(P,P)^{x(b+s)(b+s)^{-1}(y+1)+y} \\ &=\hat{e}(A_{KA},Bpri)^{(y+1)}\ \hat{e}(P,P)^{y} \\ &=key_{B} \end{aligned}$$

or in escrowless mode as below:

Alice **Bob**

$A_{KA} = x(bP + sP) \quad \longrightarrow$

$\longleftarrow \quad B_{KA}=y(aP+sP)||H(A_{KA}||\hat{e}(P,Q)^{x})$

$H(B_{KA}||\hat{e}(P,Q)^{y}) \quad \longrightarrow$

$key_{A} =\hat{e}(B_{KA},Apri)^{(x+1)}\ \hat{e}(P,Q)^{x} \qquad key_{B} =\hat{e}(A_{KA},Bpri)^{(y+1)}\ \hat{e}(P,Q)^{y}$

This scheme is consistent because:

$$\begin{aligned} key_{A} &=\hat{e}(B_{KA},Apri)^{(x+1)}\ \hat{e}(P,Q)^{x}=\hat{e}(P,Q)^{y(a+s)(a+s)^{-1}(x+1)+x} \\ &=\hat{e}(P,Q)^{xy+x+y}=\hat{e}(P,Q)^{x(y+1)+y} \\ &=\hat{e}(P,Q)^{x(b+s)(b+s)^{-1}(y+1)+y} \\ &=\hat{e}(A_{KA},Bpri)^{(y+1)}\ \hat{e}(P,Q)^{y} \\ &=key_{B} \end{aligned}$$

Any one not knowing the corresponding private key can not be able to calculate the value$\hat{e}(P,Q)^{x}$ or $\hat{e}(P,Q)^{y}$ in escrow mode or $\hat{e}(P,Q)^{x}$ or $\hat{e}(P,Q)^{y}$ in escowless mode,unless he can solve the DLP.So having got the value a user can confirm that the participant has get the correct message indeed. So this is a AKC protocol.

The AK protocol need two pairings,two expotions and one scale multiply,and the AKC protocol need two pairings ,two expentation and one scale multiply also,and can precompute one pairing.

7 Security of the proposed scheme

The proof of security of the above algorithm relies on the conjectured intractability of a problem which Zhang et al. [19] call the Bilinear Inverse Diffie-Hellman Problem: For α， $\beta \in Zq^*$, given P, αP, βP, compute v =$\hat{e}(P, P)^{\alpha^{-1}\beta}$。

We will prove the security of the protocol proposed in this paper using a communications model, which was proposed by Bellare and Rogaway incorporating later updates [2,17].

The protocol entities have an identifier *U* from a finite set {*U*1, *U*2, …, *Un*} and these entities are modeled by oracles that represent an instance in a specific protocol. And these oracles keep transcripts that keep track of messages they have sent and received,and of queries they have answered. Oracle $\Pi^{n}_{i,j}$ represents the actions between two entities, *I* and *J*, in the protocol run indexed by integer *n*. The number of entity *n* is polynomial bound in the security parameter *k*. Each entity has a pair of identity-based long-term keys, where the public key is generated using the entity's identifier and the private key is computed and issued by a key generation center(KGC) at the start of the protocol. The oracle queries are the interactions with the adversary *E*, which is a probabilistic polynomial machine that controls all the communications that take place and has access to the entities' oracles. We describe each query by the adversary *E*.

1. Send : This query allows the adversary to send a message to the entity U, say $\Pi^{n}_{i,j}$, or to initiate a new protocol run between two entities, I and J, by sending a message.
2. *Reveal* : This query allows the adversary to ask a particular oracle to reveal the session key. If a session key *Ks* has been accepted by $\Pi^{n}_{i,j}$ then it is returned to the adversary, and the oracle is called opened.
3. *Corrupt* : This query allows the adversary to ask a particular oracle to reveal the oracle's internal state and set the long-term private key of the entity to be the value *K* chosen by the adversary. The entity is called corrupted.
4. *Test* : This query allows the adversary to check the oracle's accepted a session key *Ks* by distinguishing it from a test key. A random bit *b* $\in$ {0, 1} is chosen, if *b* = 0 then the session key *Ks* is returned while if *b* = 1 then a random string is returned to guess *b*. The advantage of the adversary *E*, denoted advantage^{E}(*k*), is the probability that *E* can distinguish the queried oracle's session key *Ks* from a random string, and it is defined as:

$$Advantage^{E}(k) = |Pr[guess\ correct] - 1/2|.$$

And, an oracle's possible states are as follows:

1. *Accepted* : If an oracle decides to accept, holding a session key *Ks*, after receipt of properly formulated messages.
2. *Rejected* : If an oracle decides to reject holding a session key *Ks*.
3. *Opened* : If an oracle has answered a reveal query.
4. *Corrupted* : If an oracle has answered a corrupt query.

Definition 1.

A protocol is an AK protocol if:

1. In the presence of the benign adversary on $\Pi^{n}_{i,j}$ and $\Pi^{s}_{j,i}$, both oracles always accept holding the same session key, and this key is distributed uniformly at random on G2; if for every adversary E:

2. If uncorrupted oracles $\Pi^{n}_{i,j}$ and Π^{s}_{ji}, have matching conversations then both oracles accept and hold the same session key;

3. AdvantageE^{E}(k) is negligible.

Theorem 1. The proposed key agreement protocol is a secure AK protocol.

Proof. Condition 1 holds as follows: Both oracles accept holding the same session key as a direct result of the commutativity of exponentiation of members of the group G2. The session key is distributed uniformly at random by the fact that both oracles generate truly random $x \in_R Z_p$*. Therefore the product of these elements will also be random. Since the exponent is random, and e(P, P) is a generator of the group G2, the session key will be uniformly distributed over G2.

Condition 2 holds by the fact that if they have matching conversations then the communication was generated entirely by the two oracle's. Therefore, by the bilinearity of the pairing and the commutativity of exponentiation they accept and hold the same session key.

Condition 3 holds as follows: Consider by contradiction that $\text{Advantage}^{E}(k)$ is non-negligible. Then we can construct from E an algorithm F that solves the BIDHP with non-negligible advantage. F is given as input the output of the BDH generator B. F's task is to solve the BIDHP, namely, given P, αP and βP, compute $v = \hat{e}(P, P)^{\alpha^{-1}\beta}$.

All queries by the adversary E now pass through F.

Create: For each oracle F chooses $y_i \in Zp$, creates a public key as $u_iP = (y_iP - sP)$,and computes the private key as y_i^{-1} P. Obviously $y_iP = u_iP + sP$. However,for the i-th oracle F answers αP. Since F does not know α, it cannot calculate $\alpha^{-1}P$, the correct private key for this oracle,and for the j-th oracle F answers bP. Since F does not know a, it cannot calculate $b^{-1}P$, the correct private key for this oracle.

Corrupt: F answers Corrupt queries in the usual way, revealing the private key of the oracle being queried. However, F does not know the private key for oracle I and j; if E asks a Corrupt query on oracle I or j, F gives up.

Send F answers all send queries in the usual way, except if E asks Send $\Pi^{n}_{i,j}$, F answers xbP, In response it will get a value from j, this is set as the value yαP — this is a genuine value from j and F does not influence it.

Test At some point E will ask a single Test query of some oracle, which we assume is oracle j; if it is not, F aborts. The chance of F picking j is 1/n where n is the number of oracles (Create queries). Since it is picked it must have accepted and it must be holding a session key of the form $e(P, P)^{xy+x+y}$. However, F cannot compute this key and hence cannot simulate the query, so it simply outputs a random element of the group G2.

If F does not abort and E does not detect F's inconsistency in answering the Test query then its advantage in predicting the correct session key is $\text{Advantage}^{E}(k)$ as before. For this to be non-negligible, E must have some advantage in calculating $e(P, P)^{xy+x+y}$, given βP = yaP as input from j.

If E does not detect any inconsistencies in F's responses, then F must have non-negligible advantage $^{A}(k)$ in calculating $e(P, P)^{xy+x+y}$, but, since F does not know j's private key the session key was calculated as $\hat{e}(B_{KA},Apri)^{(x+1)}\ \hat{e}(P,P)^{x}=\hat{e}(P,P)^{(x+1)\ \alpha^{-1}\beta}\hat{e}(P,P)^{x}$. Provided that F is able to calculate $e(P, P)^{(x+1)\ \alpha^{-1}\beta}\hat{e}(P,P)^{x}$, it can calculate $e(P, P)^{\alpha^{-1}\beta}$,since it knows x. This is contradictory to the assume that there is no algorithm which can solve the BIDHP with non-negligible advantage in polynomial bounded time.

On the other hand,even if the adversary can solve the BIDHP with non-negligible advantage in polynomial bounded time,he can not compute the shared session key because he can only get $\hat{e}(P,P)^{x}$ from xβP and βP,and $\hat{e}(P,P)^{y}$ from yαP and αP.To get $e(P, P)^{xy}$ from $e(P, P)^{x}$ and $e(P, P)^{y}$ is still a hard CDH problem.

So we can say that the AK proposed is a secure AK protocol .the AKC protocol's security is same as the AK protocol.

Below are security attributes of the proposed protocol.

Known Session Key Security:

A protocol satisfies the *Known session key security*, if an adversary, having obtained some previous session keys, cannot get the session keys of the current run of the key agreement protocol.As with our protocol, suppose that the adversary knew the previous session key of a previous session key agreement protocol. In communications model, the adversary *E* is allowed to make the *Reveal* queries to any oracle except for $\Pi^{n}_{i,j}$ and $\Pi^{t}_{j,i}$. These *Reveal* queries do not help the adversary to obtain the session key between $\Pi^{n}_{i,j}$ and $\Pi^{t}_{j,i}$. Actually our protocol has the property of the known session key security because each run of the protocol produce a different session key and the adversary has to extract xP and yP from x(b+s)P and y(a+s)P to know a session key,which is equivalent to solve the Discrete-Log problem in G1. Without the ephemeral key x and y or xP and yP, the adversary cannot compute the session key $\hat{e}(P, P)^{xy+x+y}$. Therefore the knowledge of past session keys does not allow the adversary to deduce the future session keys.

Perfect Forward Secrecy:

A protocol satisfies the *Perfect Forward Secrecy*, if the compromise of the all long-term private keys of the all entities does not affect the security of the previous session keys. In communication model, the adversary *E* is allowed to make the *Corrupt* queries to any oracle.These *Corrupt* queries do not help the adversary to know the previous session keys, because the session key is generated by the entity's ephemeral private key, long-term private key and *P*(a generator of *G*1). Although the adversary knows the both entity's long-term private key and *P*, the adversary does not extract the entity's ephemeral private key. We can explain this problem as CDH problem.If the adversary knows both entity's long-term private key,he can get $\hat{e}(P, P)^{x}$ and $\hat{e}(P, P)^{y}$ from $\hat{e}(x(b+s)P,(b+s)^{-1}P)$ and $\hat{e}(y(a+s)P,(a+s)^{-1}P)$,but to get $\hat{e}(P, P)^{xy}$ from $\hat{e}(P, P)^{x}$ and $\hat{e}(P, P)^{y}$ is a CDHP.Therefore our protocol has the property of the *Perfect Forward Secrecy*.

No Key Control:

A protocol satisfies the *No Key Control*, if the session key is determined by the all entities and no one can influence the generation of the session key. In the entity *A*, the session key is generated using the *A*'s ephemeral private key x and the received messages. In other words, the generation of the session key uses the information of the all entities. All entities broadcast the messages in same time, and receive the messages. So, no one can enforce the session key to fall into a pre-determined interval. In other words, no one can influence the generation of the session key. Therefore our protocol has the property of the *No key Control*.

No Key-Compromise Impersonation:

A protocol satisfies the *No Key-Compromise Impersonation*, unless the compromise of one entity's longterm private key influences the compromise of the long-term private keys of other entities. The adversary *E* may impersonate the compromised entity in subsequent protocols, but he cannot impersonate other entities. If an adversary known entity A's long-term private key want to impersonate other entity B to share a session key with A,he chooses a value c,y and send y(c+s)P to A:

Alice E(B)

$A_{KA} = x(bP + sP) \longrightarrow$

$\longleftarrow \quad B_{KA} = y(cP + sP)$

$$\text{keya} = \hat{e}(B_{KA},\text{Apri})^{(x+1)}\ \hat{e}(P, P)^{x} \quad = \hat{e}(P, P)^{(x+1)y(c+s)(a+s)^{-1}+x}$$

$$\text{key}_E = \hat{e}(A_{KA},\text{Apri})^{(y+1)}\ \hat{}t(P, P)^{y} \quad = \hat{e}(P, P)^{(y+1)x(b+s)(a+s)^{-1}+y}$$

if E can impersonate B,keya should equal to $\text{key}_{E,}$,then c and y should satisfy the equation

$$(x+1)y(c+s)(a+s)^{-1}+x = (y+1)x(b+s)(a+s)^{-1}+y + kq,\text{for any } k \in Z,$$

so c=b+((x-y)(b-a)+kq(a+s))/(xy+y),since x is choosen by Alice,E does not know it's value,for E to choose such c and y to satisfy the equation is impossible.Therefore our protocol has the property of the *No Key-Compromise Impersonation.*

No Unknown Key-Share: A protocol satisfies the *No Unknown Key-Share*, if the all entities do not share the session key with the adversary. If the adversary convinces a group of entities, they share some session key with the adversary, and this protocol suffers from unknown key-share attack. In our protocol, the adversary must know the long-term private keys, the ephemeral private keys of the all entities to share the session key with adversary. Otherwise the adversary cannot share the session key. Therefore our protocol has the property of the *No Unknown Key-Share*.

8 Conclusion

We have presented a new ID-based key agreement protocol with or without confirmation inspired on the Noel McCullagh, Paulo S. L. M. Barreto two party key key agreement protocol. The proposed scheme can be instantiated in either escrowed or escrowless mode, and can be carried out by clients of distinct KGC's.

References

1. Noel McCullagh, Paulo S. L. M. Barreto, "A New Two-Party Identity-Based Authenticated Key Agreement," Topics in Cryptology—CT-RSA'2005, Lecture Notes in Computer Science 3376, Springer-Verlag (2005), pp. 262--274.
2. M. Bellare and P. Rogaway. Entity authentication and key distribution. In Advances in Cryptology – Crypto'93, volume 773 of Lecture Notes in Computer Science,pages 232–249. Springer-Verlag, 1994.
3. A. Shamir. Identity based cryptosystems and signature schemes. In Advances in Cryptology – Crypto'84, volume 0196 of Lecture Notes in Computer Science, pages 47–53. Springer-Verlag, 1984.
4. C. Cocks. An identity based encryption scheme based on quadratic residues. In VIII IMA International Conference on Cryptography and Coding, volume 2260 of Lecture Notes in Computer Science, pages 360–363. Springer-Verlag, 2001.
5. D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In Advances in Cryptology – Crypto'2001, volume 2139 of Lecture Notes in Computer Science, pages 213–229. Springer-Verlag, 2001.
6. R. Sakai and M. Kasahara. ID based cryptosystems with pairing on elliptic curve. In 2003 Symposium on Cryptography and Information Security – SCIS'2003,
7. A. Joux. A one round protocol for tripartite Diffie-Hellman. In Proceedings of Algorithmic Number Theory Symposium, volume 1838 of Lecture Notes in Computer Science, pages 385–394. Springer-Verlag, 2000.
8. M. Scott. Authenticated ID-based key exchange and remote log-in with insecure token and PIN number. Cryptology ePrint Archive, Report 2002/164, 2002. http://eprint.iacr.org/2002/164/.
9. N. P. Smart. An identity based authenticated key agreement protocol based on the Weil pairing. Electronics Letters, 38:630–632, 2002.
10. L. Chen and C. Kudla. Identity based authenticated key agreement from pairings.Cryptology ePrint Archive, Report 2002/184, 2002. http://eprint.iacr.org/ 2002/184.
11. Guohong Xie .Cryptanalysis of Noel McCullagh and Paulo S. L. M.Barreto's two-party identity-based key agreement.Cryptology ePrint Archive, Report 2004/308, 2004. http://eprint.iacr.org/2004/308.
12. A. Menezes, T. Okamoto, and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field," *IEEE Transaction on Information Theory*, Vol.39, pp.1639-1646,1993
13. G. Frey and H. Ruck, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves," *Mathematics of Computation*, Vol.62, pp.865-874,1994

14. D. Boneh, B. Lynn, and H. Shacham, “Short signature from the Weil pairing,” *Advances in Cryptology-Asiacrypt 2001*, LNCS 2248, pp.514-532, Springer-Verlag, 2001

15. R. Sakai, K. Ohgishi, M. Kasahara, “Cryptosystems based on pairing,” *SCIS 2000C20*,2000

16. E. R. Verheul, “Self-blindable credential certificates from the Weil pairing,” *Advances in Cryptology-Asiacrypt 2001*, LNCS 2248, pp.533-551, Springer-Verlag, 2001

17. M. Bellare and P. Rogaway, “Provable secure session key distribution – the three part case,” In Proceedings of the 27th ACM Symposium on the Theory of Computing, 1995

18. Y. Yacobi. A note on the bilinear Diffie-Hellman assumption. Cryptology ePrint Archive, Report 2002/113, 2002. http://eprint.iacr.org/2002/113.

19. F. Zhang, R. Safavi-Naini, and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In International Workshop on Practice and Theory in Public Key Cryptography – PKC’2004, Lecture Notes in Computer Science, pages 277–290. Springer-Verlag, 2004.