**ID-based tripartite Authenticated Key Agreement Protocols from pairings**

[1]Divya Nalla, K.C.Reddy

AILab, Dept of Computer/Info. Sciences, University of Hyderabad,
Gachibowli, Hyderabad, 500046, India
divya@msitprogram.net , kcrcs@uohyd.ernet.in

**Abstract**: This paper proposes ID-based tripartite authenticated key agreement protocols. The authenticated three party key agreement protocols from pairings [15], and the ID-based two party authenticated key agreement protocol [13] are studied. These two protocols are taken as the basis for designing three new ID-based tripartite authenticated key agreement protocols. The security properties of all these protocols are studied listing out the possible attacks on them. Further, these protocols are extended to provide key confirmation.

**Key Words**: Key Agreement, tripartite, elliptic curves, Weil pairing, cryptography, Identity based, ID-based, Diffie-Hellman, Key Agreement Protocols

## 1. Introduction

For secure communication, the basic problem to be resolved is to find a practical means to apply the mechanisms of encryption and authentication protocols to a multiparty conference without compromise of the security but within the constraints of the network. Asymmetric key agreement protocols are multi-party protocols in which entities exchange public information allowing them to create a common secret key with that information. This secret key is known only to those parties which are involved in the key generation and which cannot be determined by any external entity.

The first modern protocol for key agreement was the Diffie-Hellman protocol given in the seminal paper [17]. Diffie-Hellman key agreement provided the first practical solution to the key agreement problem, allowing two parties, never having met in advance or shared keying material, to establish a shared secret by exchanging messages over an open channel. The security rests in the intractability of the Diffie-Hellman(DH) problem and the related problem of computing discrete logarithms.

A huge number of two party key agreement protocols have been proposed [2]. The situation where three or more parties share a secret is often called Conference Keying [11]. The three party (or tripartite) case is of most practical importance not only because it is the most common size for electronic commerce, but because it can be used to provide a range of services for two party communications. For example, a third party can be added to chair, or referee a conversation for ad-hoc auditing, data recovery, or escrow purposes.

The MQV protocol [10], provides authentication of the parties but with message flows identical to the message flows in the naïve Diffie-Hellman protocol. Key confirmation can also be included in the MQV protocol. It works by assuming each entity has a static public/private DH key pair, and that each entity knows the public key of each other entity. When a session key has to be determined, each entity generates a pair of ephemeral public and private keys of their own. These ephemeral public keys are then exchanged to agree on a session key. Hence the problem of authenticating the session key is equivalent to the problem of authenticating the static public keys. This is usually done using a traditional approach based on the public key infrastructure.

Boneh and Franklin [6] and Cocks [4] have proposed two Identity based encryption systems which allow the replacement of a Public Key Infrastructure (PKI) with a system where the public key of an entity is given by its identity, and a Key Generation Centre (KGC) helps in generating the private key. Cocks' system is based on the Quadratic Residue theorem whereas Boneh and

[1] The author is currently associated with Center for Distributed Learning (APSCHE), IIITcampus, Gachibowli, Hyderabad, 500046, India

Franklin's system is based on the Weil pairing. A two pass Identity based (ID-based) Authenticated Key Agreement protocol based on Weil pairing has been proposed in [13].

Joux [1] proposed a tripartite generalisation of the Diffie-Hellman protocol using the Weil and Tate pairings. Joux's protocol for tripartite key agreement has been modified in [15] to provide authentication by including certificates.

This paper proposes a set of protocols fusing the ideas of both ID-based approach and the tripartite key agreement of Joux. The result is tripartite ID-based Authenticated Key Agreement protocols. Since they are ID-based protocols, they are more desirable compared to the PKI based protocols. Key confirmation is also provided for the protocols. The security properties of the protocol are studied, and it is found that they satisfy the desirable security properties. The next section discusses the advantages of an ID-based Public Key Cryptosystem (PKC).

The remaining paper is organised as follows. Section 3 gives the definition for a Weil pairing and discusses the Diffie-Hellman Problem. Section 4 describes the protocol attributes and the desired properties of the protocol, section 5 and 6 review the ID-based two party key agreement protocol, and the Joux's protocol. Section 7 presents the new protocols. Section 8 reviews the security properties of the protocol. The one-round ID-based tripartite key agreement protocol by Zhang et al is discussed and compared with the ID-AK-3 protocol in section 9. Section 10 defines ID based tripartite authenticated key agreement protocols with key confirmation. Conclusions are given in Section 11.

## 2. Identity based Public Key Cryptosystem

Problems with the traditional PKCs are the high cost of the infrastructure needed to manage and authenticate public keys, and the difficulty in managing multiple communities. Whilst ID-based PKC will not replace the conventional PKIs, it offers easier alternative solutions to some of these problems.

In ID-based PKC, everyone's public keys are predetermined by information that uniquely identifies them, such as their email address. This concept was first proposed by Shamir [3]. Shamir's original motivation for ID-based encryption was to simplify certificate management in e-mail systems. When Alice sends mail to Bob at bob@hotmail.com she simply encrypts her message using the public key string bob@hotmail.com. There is no need for Alice to obtain Bob's public key certificate. When Bob receives the encrypted mail he contacts the KGC. He authenticates himself to the KGC and obtains his private key from the KGC. Bob can then read his e-mail. Unlike the existing secure e-mail infrastructure, Alice can send encrypted mail to Bob even if Bob has not yet setup his public key certificate. It should be noted that key escrow is inherent in ID-based systems since the KGC knows Bob's private key. For various reasons, this makes implementation of the technology much easier, and delivers some added information security benefits. ID-based PKC remained a theoretical concept until [4] and [6] were proposed.

An ID-based system requires a trusted KGC. The KGC generates the private keys of the entities in the group using the public key, which in turn is based on the identity of the entity. Since a single authority for key generation may present an obvious point of compromise or system failure, and it can masquerade as any given entity, it is split into two or more cooperating parties. The authorities perform a one-time set-up in order to share the system secrets in a secure manner. The user proves itself to each authority. Each authority then returns its part of the private key. The user combines these parts to obtain his private key. This provides more security as the private key remains split until use. We can also have *n*-authorities in the system wherein no *n-1* of them can generate a key or compromise the system.

A hierarchy of KGCs is desirable in an ID-based encryption system, as it greatly reduces the workload on the master server(s) and allows key escrow at several levels. For instance, if the users of the system are employees of corporations, then it is natural to want each corporation to be able to generate private keys for their employees, so that employees request their keys from

their corporation, rather than the top-level KGC. Only corporations request their domain secret from the top-level KGC. This is an example of a 2-pass hierarchical ID-based encryption system (HIBE) [9]. The advantage of an HIBE over standard PKI is that senders can derive the recipient' s public key from their address without an online lookup.

With traditional PKI, sending the message implies that the recipient can read it since the private key must exist. This is not true in ID-based PKC. In ID-based PKC, the system (rather than the user / sender) determines whether the recipient is able to read a particular message. Another advantage is that, since the associated private key doesn' t necessarily exist, these conditions need not be pre-arranged and can be ephemeral down to a single transaction.

Also, an ID-based system allows more dynamism into the system by not requiring the certificates prior to a transaction. Any member can get his/her private key on-the-fly from the KGC. It should be noted that this centre is required only in obtaining the private key of the user, but not in generating the key (not as a key distribution centre).

Having studied the advantages of an ID-based public key cryptosystem, it would be advantageous to incorporate ID-based PKC in key agreement protocols. This has already been done in [13] and [7]. The next section discusses some preliminaries required for the ID-based protocols to be proposed in the following sections.

## 3. Preliminaries

### 3.1. The Weil Pairing

Let G be a subgroup of the group of points on the Elliptic curve $E$ over the finite field $F_q$. Let the order of G be denoted by $l$ , and define $k$ to be the smallest integer such that

$$l / q^k - 1$$

In practical implementations we will require $k$ to be small and so we will usually take $E$ to be a super singular curve over $F_q$

The modified Weil pairing [13] is a map $\hat{e}: G \times G \to F_{q^k}^*$ which satisfies the following properties:

1. Bilinear
   $\hat{e}(P_1 + P_2, Q) = \hat{e}(P_1, Q).\hat{e}(P_2, Q)$
   $\hat{e}(P, Q_1 + Q_2) = \hat{e}(P, Q_1).\hat{e}(P, Q_2)$
   $i.e., \hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ where $a, b \in Z_q^*$
2. Non-Degenerate
   There exists a $P \in$ G such that $\hat{e}(P, P) \neq 1$
3. Computable:
   One can compute $\hat{e}(P, Q)$ in polynomial time.

The non-degeneracy defined here does not hold for the standard Weil Pairing $e(P, Q)$. A more comprehensive description is provided in [6].

### 3.2. Diffie-Hellman Problem

The Decisional Diffie-Hellman (DDH) Problem is a gold mine. The key agreement protocols available today are mostly based on DDH problem. The problem is discussed thoroughly in [5].

When elliptic curves were first proposed in [16], computing the number of points of a given curve was a challenging task. For this reason and also to simplify the addition formulas, it was shown later on that some of these special cases are not good enough. Three weak special cases have been identified. In one of them, the discrete logarithm problem becomes easy (shown in [12]). In the other two cases, the discrete logarithm problem on the elliptic curve is transformed into a discrete logarithm problem in a small extension of the field of definition of the elliptic

curve. These two reductions are called the MOV reduction (which is based on the Weil Pairing) and the FR reduction (which is based on the Tate pairing) (survey of these is given in [14]). It is shown in [1] that it is not the case and that these reductions can be turned from cryptanalytic to cryptographic tools.

### 3.2.1. *Computational Diffie-Hellman Problem*

Consider a subgroup of $Z_p^*$ of order $q$, and generator $g$. The Computational Diffie-Hellman problem is to compute $g^{ab}$ from $g^a$ and $g^b$.

### 3.2.1. *Bilinear Diffie-Hellman Problem*

The security of the ID-based system in this paper depends on a variant of Computational Diffie-Hellman assumption called the Bilinear Diffie-Hellman assumption (also called the Weil Diffie-Hellman Assumption). [6]

Let $G_1$, $G_2$ be two groups of prime order q. Let $\hat{e}: G_1 \times G_1 \to G_2$ be a bilinear map and let $P$ be a generator of $G$. The BDH problem in ($G_1$, $G_2$, $\hat{e}$ ) is as follows:

Given ($P$, $aP$, $bP$, $cP$) for some $a, b, c \in Z_q^*$, compute W = $\hat{e}(P,P)^{abc} \in G_2$

## 4. Protocol Attributes

The various security attributes required for a key agreement protocol are discussed in this section along with the desirable properties. A key agreement protocol is required to have the following properties:

- **Key Freshness:** A key is **fresh** if it can be guaranteed to be new, as opposed to possibly an old key being reused by an adversary.
- **Contributory:** A key agreement protocol is **contributory** if each party equally contributes to the key and guarantees its freshness.
- **Implicit Key Authentication:** Let $R$ be an $n$-party key agreement protocol, $M$ be the set of group members and let $S_n$ be the secret key jointly generated as a result of $R$. We say that $R$ provides **implicit Key Authentication** if each $M_i \in M$ is assured that no party $M_q \notin M$ can learn the Key $S_n$.
- **Key Confirmation:** A protocol provides **Key Confirmation** if a party is assured that its peer actually has possession of a particular secret key.
- **Key Integrity:** A contributory key agreement protocol provides **Key Integrity** if a party is assured that its particular secret key is a function of **only** the individual contributions of all protocol parties.
- **Forward secrecy**: A protocol provides **perfect forward secrecy** (PFS) if compromise of a long-term key(s) cannot result in the compromise of the past session keys.
- **Key Independence: Key Independence** guarantees that a passive adversary who knows a proper subset of group keys cannot discover any other group key.
- **Know Key attacks:** A protocol is said to be vulnerable to **known key attacks** if compromise of past session keys allows: 1) a passive adversary to compromise future session keys, or 2) an active adversary to impersonate one of the protocol parties.

The desirable properties of a key agreement protocol would be:

- low communication overhead (the number of messages and message size should be less)
- minimal number of passes and rounds
- role symmetric (messages transmitted and computations performed by all the entities have the same structure).

Two important computational attributes also have to be considered for the key agreement protocol: *computation overhead* and the ability to perform *Precomputation*. It may be desirable that some computations may be done offline (i.e., pre-computed or computed during the protocol run)

## 5. ID-based two party Authenticated Key Agreement Protocol

### 5.1 Assumptions

Suppose we have a subgroup G of an Elliptic curve for which the modified Weil Pairing $\hat{e}$ maps into the finite field $F_{q^k}$. We assume the following:

- $q^k$ is large enough to make solving discrete logarithms in a finite field infeasible
- Elliptic curve contains a large prime subgroup of order $l$ such that solving discrete logarithms in the subgroup of order $l$ is also infeasible.

(In all the protocol messages "Sends to" is denoted by "→". The notation $A \rightarrow B$: x means that $A$ sends the message x to $B$)

Let V: $F^*_{q^k} \rightarrow \{0, 1\}^*$ be the key derivation function [13] and let H: $\{0, 1\}^* \rightarrow$ G denote a cryptographic hash function.

### 5.2 System Settings

The KGC chooses a secret key $s \in \{1,\ldots.. l\text{-}1\}$

The KGC produces a random $P \in G$ and computes $P_{KGC} = [s]\, P$. Then the KGC publishes $(P, P_{KGC})$

When a user with identity $ID$ wishes to obtain a public/private key pair, the public key is given by

$$W_{ID} = H(ID)$$

And the KGC computes the private key

$$w_{ID} = [s]\, W_{ID}$$

This calculation can be performed using multiple key generation centres, each generating a part of the private key (as mentioned in the previous section). These are combined by the user to obtain his / her private key.

### 5.3 Protocol

Suppose two users $A$ and $B$ wish to agree on a key. We denote the private keys of these users by $w_A = [s]\, W_A$ and $w_B = [s]\, W_B$ respectively, which have been obtained from the key generation centre (KGC).

Each user generates an ephemeral private key, say $a$ and $b$. The data flows of the protocol are as follows.

$$A \rightarrow B : T_A = [a]\, P$$

$$B \rightarrow A : T_B = [b]\, P$$

User $A$ then computes

$$k_A = \hat{e}\,([a]\, W_B, P_{KGC}) \,.\, \hat{e}\,(w_A, T_B)$$

and user $B$ computes

$$k_B = \hat{e}\,([b]\, W_A, P_{KGC}) \,.\, \hat{e}\,(w_B, T_A)$$

$$k_A = k_B = k_{AB} = \hat{e}\,([a]W_B + [b]W_A, [s]P)$$

hence the shared secret depends on the identities $W_A$, $W_B$ of the two parties, the secret key $s$ of the key generation centre and the two ephemeral keys $a$, $b$.

The protocol requires each party to compute two elliptic curve multiplications and two Weil pairings and provides Known key security, Forward secrecy, Key control.

## 6. Joux's Protocol and its improvements

Joux [1] introduced a simple one round tripartite key agreement protocol using Weil pairing. In Joux's protocol, $a, b, c \in Z_q^*$ are selected uniformly at random by *A*, *B*, and *C* respectively.

Protocol messages:

$A \rightarrow B, C : aP$

$B \rightarrow A, C : bP$

$C \rightarrow A, B : cP$

In this protocol, the ordering of the protocol messages is unimportant and any of the three entities can initiate the protocol. Once the communication is over, *A*, *B*, and *C* compute their keys $k_A, k_B, k_c$.

$k_A = \hat{e}\ (bP, cP)^a$ ; $k_B = \hat{e}\ (aP, cP)^b$ ; $k_C = \hat{e}\ (aP, bP)^c$

$k_A = k_B = k_C = k_{ABC} = \hat{e}\ (P, P)^{abc}$

$k_{ABC}$ is the common session key. Success of this protocol lies in the hardness of the Bilinear Diffie-Hellman problem (BDHP).

Just like the unauthenticated two party Diffie-Hellman protocol, Joux's protocol is subject to a classic man-in-the-middle attack (discussed in section 8). Including authentication in the protocol can thwart this attack. Al-Riyami and Paterson [15] proposed a few improvements to the Joux's protocol. They are called Tripartite Authenticated Key agreement (TAK) Protocols.

A Certification Authority (CA) is used in the initial set up stage to provide certificates, which binds user's identities to long-term Keys. The certificate for *A* will be of the form:

$Cert_A = (I_A \| \mu_A \| P \| S_{CA}(I_A \| \mu_A \| P))$.

Where $I_A$ denotes the identity of *A*, $\|$ denotes the concatenation of data items, $S_{CA}$ denotes the CA's signature. *x*, *y*, and *z* are *A*, *B* and *C*'s long term private Keys, and $\mu_A = xP$, $\mu_B = yP$, $\mu_C = zP$ are the long term public Keys of *A*, *B* and *C*. Short-term Keys $a, b, c \in Z_q^*$ are selected uniformly at random by *A*, *B*, and *C* respectively.

**Protocol messages:**

$A \rightarrow B, C : aP$

$B \rightarrow A, C : bP$

$C \rightarrow A, B : cP$

**TAK Key generation:**

Four types of key generation are given below. The keys computed by the entities are given below

**Type 1**

$K_A = \hat{e}(bP, cP)^a . e(yP, zP)^x$

$K_B = \hat{e}(aP, cP)^b . e(xP, zP)^y$

$K_C = \hat{e}(aP, bP)^c . e(xP, yP)^z$

$K_{ABC} = K_A = K_B = K_C = \hat{e}(P, P)^{abc+xyz}$

**Type 2**

$K_A = \hat{e}(bP, zP)^a e(yP, cP)^a e(bP, cP)^x$

$K_B = \hat{e}(aP, zP)^b e(xP, cP)^b e(aP, cP)^y$

$K_C = \hat{e}(aP, yP)^c e(xP, bP)^c e(aP, bP)^z$

$K_{ABC} = K_A = K_B = K_C = \hat{e}(P, P)^{(ab)z+(ac)y+(bc)x}$

**Type 3**

$$K_A = \hat{e}(yP, cP)^a e(\hat{bP}, zP)^a e(\hat{yP}, zP)^x$$

$$K_B = \hat{e}(aP, zP)^b e(\hat{xP}, cP)^b e(\hat{xP}, zP)^y$$

$$K_C = \hat{e}(aP, yP)^c e(\hat{xP}, bP)^c e(\hat{xP}, yP)^z$$

$$K_{ABC} = K_A = K_B = K_C = \hat{e}(P, P)^{(xy)c+(xz)b+(yz)c}$$

**Type 4**

$$K_A = \hat{e}(bP + H(bP \,||\, yP)yP, cP + H(cP \,||\, zP)zP)^{a+H(aP||xP)x}$$

$$K_B = \hat{e}(aP + H(aP \,||\, xP)xP, cP + H(cP \,||\, zP)zP)^{b+H(bP||yP)y}$$

$$K_C = \hat{e}(aP + H(aP \,||\, xP)xP, bP + H(bP \,||\, yP)yP)^{c+H(cP||zP)z}$$

$$K_{ABC} = K_A = K_B = K_C = \hat{e}(P, P)^{(a+H(aP||xP)x)(b+H(bP||yP)y)(c+H(cP||zP)z)}$$

For a single round protocol, TAK-4 is the most secure, followed by TAK-2.
The comparison of the security attributes of these protocols is shown in Appendix A.

## 7. ID-based tripartite Key agreement protocol based on pairings

The advantage of the Joux's protocol over any other tripartite key agreement protocol is that a session key can be established in just one round. TAK protocols overcome the disadvantage of lack of authentication in Joux's protocol by inco rporating authentication into the protocol using certificates. This section proposes protocols fusing the ideas of both ID-based approach and the tripartite key agreement of Joux.

The assumptions and the system settings for the new protocol are same as in the two party ID-based key agreement protocol.

Suppose we have a subgroup $G$ of an Elliptic curve for which the modified Weil Pairing $\hat{e}$ maps into the finite field $F_{q^k}$. i.e., $\hat{e}$ is a modified Weil pairing from $G$ X $G$ to $F_{q^k}$

Let $V$: $F_{q^k}^* \rightarrow \{0, 1\}^*$ be the key derivation function [13] and let $H$: $\{0, 1\}^* \rightarrow G$ denote a cryptographic hash function.

### 7.1 Initial Settings

The KGC chooses a secret key $s \in \{1,\ldots.. l\text{-}1\}$
The KGC produces a random $P \in G$ and computes $P_{KGC} = [s]\, P$. Then the KGC publishes $(P, P_{KGC})$

### 7.2 Protocol

Let $A$, $B$ and $C$ be the three parties wishing to compute a common shared key. $A$ sends its identity $A$ to the KGC and gets its private key from the KGC.
$A$'s Public key: $W_A = H(A)$
$A$'s Private key: $w_A = [s]\, W_A$ (computed by the KGC)
$B$'s public and private keys are given by $W_B$, $w_B = [s]\, W_B$ and $C$'s public and private keys are given by $W_C$, $w_C = [s]\, W_C$ respectively.
The pairs $(W_{ID}, w_{ID})$ for $A$, $B$, and $C$ are their static (or long term) public/private key pairs.

Three protocols are presented here: ID-AK-1 (ID based Authenticated Key Agreement Protocol-1), ID-AK-2 and ID-AK-3.

### 7.2.1 ID-AK-1

Each user generates a random number, say *a* and *b*, and *c*. The ephemeral (or short term) public keys would be $[a]P$ , $[b]P$, and $[c]P$, and the ephemeral or short term private keys would be *a*, *b*, and *c*.

$A \rightarrow B, C : [a]P$
$B \rightarrow A, C : [b]P$
$C \rightarrow A, B : [c]P$

User *A* computes

$$k_A = \hat{e}([b]P,[c]P)^a.e(W_B, P_{KGC}).e(W_C, P_{KGC}).e(w_A, P)$$
$$= \hat{e}(P,P)^{abc}.e(W_B,P)^s.e(W_C,P)^s.e([s]W_A,P)$$
$$= \hat{e}(P,P)^{abc}.e(W_A,P)^s.e(W_B,P)^s.e(W_C,P)^s$$

User *B* computes

$$k_B = \hat{e}([a]P,[c]P)^b.e(w_B,P).e(W_C,P_{KGC}).e(W_A,P_{KGC})$$
$$= \hat{e}(P,P)^{abc}.e(W_A,P)^s.e(W_B,P)^s.e(W_C,P)^s$$

and user *C* computes

$$k_C = \hat{e}([a]P,[b]P)^c.e(W_B,P_{KGC}).e(w_C,P).e(W_A,P_{KGC})$$
$$= \hat{e}(P,P)^{abc}.e(W_A,P)^s.e(W_B,P)^s.e(W_C,P)^s$$

The shared secret key is the output of the key derivation function *V* with $k_{ABC}$ as input where

$$k_{ABC} = \hat{e}(P,P)^{abc}.e(W_A,P)^s.e(W_B,P)^s.e(W_C,P)^s = e(P,P)^{abc}.e((W_A+W_B+W_C)),[s]P)$$

and hence depends on the identities of the three entities $W_A$, $W_B$, $W_C$, and the three ephemeral private keys *a*, *b*, and *c*. Hence the secret key is *V* ($k_{ABC}$).

In this protocol each user needs to compute four Weil pairings and one Elliptic curve scalar multiplication in this protocol. But three of the Weil pairings can be pre computed and only one pairing needs to be computed for each session.

### 7.2.2 Type 2 (ID-AK-2)

Each user generates a random number, say *a* and *b*, and *c*. The ephemeral (or short term) public keys would be $aP_{KGC}$ , $bP_{KGC}$, and $cP_{KGC}$, and the ephemeral or short term private keys would be *a*, *b* and *c*.

The data flows of the protocol are as follows.

$A \rightarrow B, C : [a]P_{KGC}$
$B \rightarrow A, C : [b]P_{KGC}$
$C \rightarrow A, B : [c]P_{KGC}$

User *A* then computes

$$k_A = \hat{e}([a]w_A,P).e(W_B,[b]P_{KGC}).e(W_C,[c]P_{KGC})$$
$$= \hat{e}(w_A,P)^a.e(W_B,P_{KGC})^b.e(W_C,P_{KGC})^c$$
$$= \hat{e}([s]W_A,P)^a.e(W_B,P_{KGC})^b.e(W_C,P_{KGC})^c$$
$$= \hat{e}(W_A,[s]P)^a.e(W_B,[s]P)^b.e(W_C,[s]P)^c$$
$$= \hat{e}([a]W_A+[b]W_B+[c]W_C,[s]P)$$

User *B* computes

$$k_B = \hat{e}(W_A,[a]P_{KGC}).e([b]w_B,P).e(W_C,[c]P_{KGC})$$
$$= \hat{e}([a]W_A+[b]W_B+[c]W_C,[s]P)$$

Similarly user *C* computes

$$k_C = \hat{e}(W_A, [a]P_{KGC}).e(W_B, [b]P).e([c]w_C, P_{KGC})$$
$$= \hat{e}([a]W_A + [b]W_B + [c]W_C, [s]P)$$

Hence the shared secret key is the output of the key derivation function *V* with $k_{ABC}$ as input where

$$k_{ABC} = k_A = k_B = k_C = \hat{e}([a]W_A + [b]W_B + [c]W_C, [s]P)$$

which depends on the identities of the three participants $W_A$, $W_B$, and $W_C$, the secret key *s* of the key generation centre and the three short term keys *a*, *b*, and *c*. The secret key is $V(k_{ABC})$.

In this protocol each user is required to compute two elliptic curve multiplications and three Weil pairings. This protocol provides forward secrecy, known key security and key control.

### 7.2.3 Type 3 (ID-AK-3)

Each user generates random $a, b, c \in Z_q^*$, which are the ephemeral private keys of *A*, *B*, and *C* respectively. The data flows of the protocol are as follows.

$A \rightarrow B$ : $[a]P$ , $[a]W_C$ ; $A \rightarrow C$: $[a]P$ , $[a]W_B$
$B \rightarrow A$: $[b]P$ , $[b]W_C$ ; $B \rightarrow C$: $[b]P$ , $[b]W_A$
$C \rightarrow A$: $[c]P$ , $[c]W_B$ ; $C \rightarrow B$: $[c]P$ , $[c]W_A$

User *A* computes the key

$$k_A = \hat{e}([a](W_B + W_C), P_{KGC}).e(w_A, ([b]P + [c]P)).e([b]W_C, P_{KGC}).e([c]W_B, P_{KGC})$$
$$k_B = \hat{e}([b](W_A + W_C), P_{KGC}).e(w_B, ([a]P + [c]P)).e([a]W_C, P_{KGC}).e([c]W_A, P_{KGC})$$
$$k_C = \hat{e}([c](W_A + W_B), P_{KGC}).e(w_c, ([a]P + [b]P)).e([a]W_B, P_{KGC}).e([b]W_A, P_{KGC})$$

Hence the shared secret key is the output of the key derivation function *V* with $k_{ABC}$ as input where

$$k_{ABC} = k_A = k_B = k_C = \hat{e}([a](W_B + W_C) + [b](W_A + W_C) + [c](W_A + W_B), [s]P)$$

depends on the identities $W_A$, $W_B$, and $W_C$ of the three parties, the secret key *s* of the key generation centre, and the ephemeral private keys *a*, *b*, and *c*. The secret key is $k = V(k_{ABC})$.

The protocol is role symmetric since each participant executes the same number of operations. This protocol requires each participant to compute two elliptic curve additions, four elliptic curve scalar multiplications, and four Weil pairings. Since one of the elliptic curve additions can be pre-computed (not required to be computed for each session), the total number of elliptic curve additions is one. The algorithm for elliptic curve addition is given in [8].

TAK protocols require deployed PKI to authenticate the long-term public keys, whilst these protocols use an ID-based system. Hence, depending on the application this protocol may be more applicable. i.e., for applications where PKI is not deployed, and for some specific applications this protocol would be more appropriate.

## 8. Protocol Properties

Table 1 gives a comprehensive idea about the number of computations per user in each of the above protocols. The basic computations being Elliptic curve Scalar Multiplications, Addition of points on the Elliptic curve, and the evaluation of Weil pairings.

| | EC Scalar Multiplications | EC Additions | Hash function | Weil pairings | Pairings that can be Precomputed (once for all the sessions) |
|---|---|---|---|---|---|
| Joux | 1 | none | none | 1 | none |
| TAK-1 | 1 | none | none | 2 | 1 |
| TAK-2 | 1 | none | none | 3 | none |
| TAK-3 | 1 | none | none | 3 | none |
| TAK-4 | 1 | none | 3 | 1 | none |
| ID-AK-1 | 1 | none | none | 4 | 3 |
| ID-AK-2 | 2 | none | none | 3 | none |
| ID-AK-3 | 4 | 1 | none | 4 | 1 |

Table 1: Number of computations to be performed by each user in each tripartite protocol.

**8.1 Attacks**

A variety of attacks on the TAK and ID-AK protocols are presented here.

**8.1.1 Man-in-the-middle attacks**

Suppose an adversary *D* is capable of intercepting *A*'s communications with *B* and *C*, impersonating *A* to the other entities and impersonating the other entities to *A*. Let $D_A$ denote the adversary *D* impersonating *A* in sending or receiving messages intended for or originating from *A*. Similarly, $D_{B,C}$ denotes an adversary impersonating both *B* and *C*.

Let $\delta$, $\delta'$ and $\delta''$ be random values from *D*'s choice. We assume *A* initiates a run of the protocol. The man-in-the middle attack is then executed as follows for the Joux's protocol.

1. $D_{B,C}$ intercepts $aP$ from *A*, and $D_A$ forwards $\delta P$ to *B*, *C*.
2. $D_A$ intercepts $bP$ from *B*, and $D_B$ forwards $\delta' P$ to *A*.
3. $D_A$ intercepts $cP$ from *C*, and $D_C$ forwards $\delta'' P$ to *A*.

At the end of this attack, *D* impersonating *A* has agreed a key $k_{D_A BC} = \hat{e}(P,P)^{\delta bc}$ with *B* and *C*, while *D* impersonating *B* and *C* has agreed a second Key $k_{AD_{B,C}} = \hat{e}(P,P)^{a\delta\delta'' }$ with *A*. If these keys are used to encrypt subsequent communications, then *D*, by appropriately decrypting and re-encrypting messages, can now continue masquerading as *A* to *B*, *C* and *B*, *C* to *A*.

Joux's protocol, ID-AK-1, and ID-AK-2 protocols are affected by this attack. Man-in-the middle attacks are prevented in TAK protocols and ID-AK-3 protocol since the computation of $K_{ABC}$ involves the long-term secret Keys.

**8.1.2 Two-Key Compromise Attacks**

This is a serious attack on TAK-1 and ID-AK-1. The prerequisites for the attack are:

1. An adversary *D*, by eavesdropping, has obtained the short term public values $bP$ and $cP$.
2. *D* also obtained the session key $K_{ABC}$
3. *D* has also somehow acquired the short term key $a$ used in that protocol run.

*D* can now calculate $K_{ABC}$. $\hat{e}(bP, cP)^{-a} = \hat{e}(P,P)^{xyz}$. *D* can then go on to impersonate any of *A*, *B*, or *C* in subsequent protocol runs. Thus TAK-1 is severely affected by this attack. It is even more easy to attack in the case of Joux's protocol. Having a hash function to perform key derivation can prevent this attack. Thus ID-AK-1 protocol is prevented from this attack. This attack does not apply to the ID-AK-2 and ID-AK-3 protocols because of the way long-term components are combined with short-term components in $k_{ABC}$.

**8.1.3. Forward Security Weakness**

A protocol is not forward secure if the compromise of the long-term secret keys of one or more entities also allows an adversary to obtain session keys previously established between honest entities.

- TAK-1 and TAK-4 are forward secure
- TAK-2 and TAK-3 are not forward secure
- ID-AK-1, ID-AK-2, and ID-AK-3 are forward secure since the key always depends on the short-term secret keys.

**8.1.4. Known Key Attacks**

A protocol is said to be vulnerable to known key attacks if a compromise of past session keys allows a passive adversary to compromise future session keys, and an active adversary to impersonate one of the protocol parties.

Suppose *D* is an adversary impersonating as *B* and *C* to *A*. The attack would be as follows.

Session 1: $A \rightarrow D_{B,C} : aP$

Session 2: $A \rightarrow D_{B,C} : a'P$

Session 3: $A \rightarrow D_{B,C} : a''P$

*D* reflects and replays pretending to be *B* and *C*, to complete session 1.

$D_B \rightarrow A$: $a'P$

$D_C \rightarrow A$: $a''P$

Then *A* computes the session Key $k_{AD_BD_C} = \hat{e}(P,P)^{aa'a''}$.

TAK-1, ID-AK-1 and ID-AK-2 are affected by this attack.

| | Joux | TAK-1 | TAK-2 | TAK-3 | TAK-4 | ID-AK-1 | ID-AK-2 | ID-AK-3 |
|---|---|---|---|---|---|---|---|---|
| Key freshness | √ | √ | √ | √ | √ | √ | √ | √ |
| Contributory | √ | √ | √ | √ | √ | √ | √ | √ |
| Implicit Key authentication | X | X | √ | √ | √ | | √ | √ |
| Key Integrity | √ | √ | √ | √ | √ | √ | √ | √ |
| Key Confirmation | X | X | X | X | X | X | X | X |
| Forward secrecy | NA | √ | X | X | √ | √ | √ | √ |
| Key Compromise impersonation | NA | X | √ | √ | √ | X | √ | √ |
| Key Independence | √ | √ | √ | √ | √ | √ | √ | √ |
| vulnerability to known Key attacks | X | X | √ | √ | √ | X | X | √ |
| Identity based | X | X | X | X | X | √ | √ | √ |

√ - The property is satisfied ; X – The property is not satisfied ;
NA – The property is not applicable to the protocol

Table 2: Comparison of security attributes for tripartite protocols based on pairings

### 8.1.5. Security Summary

The new protocols have been examined for some important attacks in the previous sections. Though this paper does not provide an extensive study of all the possible attacks on the protocols, some of the important ones have been considered. Table 2 compares the security attributes of Joux' s protocol, TAK1, TAK-2, TAK-3, TAK-4, and ID-AK-1, ID-AK-2 and ID-AK-3 protocols. The next section discusses anotherID-based one-round tripartite authenticated key agreement protocol and compares the computations with the new ID-AK-3 protocol.

## 9. ID-based one-round tripartite authenticated key agreement protocol by Zhang et al

This section discusses the One-round ID-based tripartite authenticated key agreement protocol [7], and compares the computations in this protocol with the computations of the proposed ID-AK-3 protocol. Since it is shown already that ID-AK-3 is more secure compared to ID-AK-1 and ID-AK-2, we compare the existing protocol with ID-AK-3 protocol.

### 9.1 Protocol

The initial settings in this protocol are same as those for ID-AK protocols proposed in this paper.

In this protocol, each user *A*, *B*, and *C* selects random values *a, a', b, b', c, c'* in $Z_q^*$ as their temporary private keys. The protocol messages are as follows:

$A \rightarrow B, C : P_A = aP, P_A' = a'P, T_A = \mathrm{H}(P_A, P_A')w_A + aP_A'$

$B \rightarrow A, C : P_B = bP, P_B' = b'P, T_B = \mathrm{H}(P_B, P_B')w_B + bP_B'$

$C \rightarrow A, B : P_C = cP, P_C' = c'P, T_C = \mathrm{H}(P_C, P_C')w_C + cP_C'$

Where *H* is a hash function defined as $H:\{0,1\}^* \rightarrow Z_q$

*A* verifies:

$$e(T_B + T_C, P) = e(H(P_B, P_B')Q_B + H(P_C, P_C')Q_C, P_{KGC}).e(P_B, P_B').e(P_C, P_C')$$

If the above equation holds, then *A* computes

$$k_A^{(1)} = e(P_B, P_C)^a, k_A^{(2)} = e(P_B, P_C')^a, k_A^{(3)} = e(P_B', P_C)^a, k_A^{(4)} = e(P_B, P_C')^{a'}$$

$$k_A^{(5)} = e(P_B, P_C)^{a'}, k_A^{(6)} = e(P_B, P_C')^{a'}, k_A^{(7)} = e(P_B', P_C)^{a'}, k_A^{(8)} = e(P_B, P_C')^{a'}$$

Similarly B and C compute $k_B^{(i)}$ and $k_C^{(i)}$ for i from 1 to 8.

$$k_B^{(1)} = e(P_A, P_C)^b, k_B^{(2)} = e(P_A, P_C')^b, k_B^{(3)} = e(P_A', P_C)^b, k_B^{(4)} = e(P_A, P_C')^{b'}$$

$$k_B^{(5)} = e(P_A, P_C)^{b'}, k_B^{(6)} = e(P_A, P_C')^{b'}, k_B^{(7)} = e(P_A', P_C)^{b'}, k_B^{(8)} = e(P_A, P_C')^{b'}$$

$$k_C^{(1)} = e(P_A, P_B)^c, k_B^{(2)} = e(P_A, P_B')^c, k_C^{(3)} = e(P_A', P_B)^c, k_C^{(4)} = e(P_A, P_B')^{c'}$$

$$k_C^{(5)} = e(P_A, P_B)^{c'}, k_B^{(6)} = e(P_A, P_B')^{c'}, k_C^{(7)} = e(P_A', P_B)^{c'}, k_C^{(8)} = e(P_A, P_B')^{c'}$$

it can be observed that the common keys are $k^{(i)} = k_A^{(i)} = k_B^{(i)} = k_C^{(i)}$ for i = 1,2,….8.

### 9.2 Computations

In this protocol, each entity is required to compute 4 pairings for verification, and 1 pairing to generate 1 session key. This protocol computes 8 session keys at a time (hence each entity has to compute 8 pairings per session) and the computations in this protocol are compared to eight times TAK-4 protocol computations in Zhang et al's paper[7]. But the requirement of generating eight session keys does not seem to be of much practical use. Instead, a single session key generated can be used to generate more session keys by any simple method like adding some number to the key and then hashing it.

Also, a simplified version of the same protocol is proposed in the same paper [7] where an entity needs to compute 5 pairings (4 for verification and 1 for the generation of the session key) to get one session key.

Comparing these computations with those in the ID-AK-3 protocol, ID-AK-3 protocol is computationally more efficient. Table 3 shows these comparisons. Also, the key computed by Zhang, Liu and Kim [7] is of the same type as in the Joux's protocol.

| | Number of Computations | | | |
|---|---|---|---|---|
| | Weil Pairings | Scalar multiplications | Exponentiations | Hash functions |
| Zhang's protocol | 8 | 6 | 8 | 3 |
| Simplified version of zhang's protocol | 5 | 5 | 1 | 3 |
| ID-AK-3 | 4 | 4 | - | 3 |

Table 3: Comparison of computations in Zhang's protocol and ID-AK-3 Protocol

## 10. Identity based Key Agreement Protocols based on pairings with Key confirmation (ID-AKC)

Just as with the MQV protocol [10] it is trivial to add key confirmation property to ID-AK protocols. ID-AKC protocol requires four rounds to complete key confirmation. We require a message authentication code MAC, and the key derivation function $V$ to give the MAC key and the shared key $k'$ and $k$ respectively.

We let $R = \hat{e}(P, P)^{abc}$ for protocols ID-AK-1 and ID-AK-2, and the protocol will be as follows:

- Round 1 will be the same as in the protocol
- Round 2:
  - $A \rightarrow B, C$: $M_1 = \text{MAC}_{k'}(2, A, B, C, R)$
- Round 3:
  - $B \rightarrow A, C$: $M_2 = \text{MAC}_{k'}(3, A, B, C, R)$
- Round 4:
  - $C \rightarrow B, A$: $M_3 = \text{MAC}_{k'}(5, A, B, C, R)$

$A$ checks $M_2$ and $M_3$, $B$ checks $M_1$ and $M_3$, and $C$ checks $M_2$ and $M_1$.Assuming that all parties choose a different ephemeral key for each run of the protocol, one can heuristically argue that we will obtain the desired key confirmation. In case of ID-AK-3 protocol, $R$ is taken as $k$. The rest of the protocol is same as that discussed in section 7.

## 11. Conclusions

Three new ID based tripartite key agreement protocols have been proposed in this paper. Having studied their security properties, it can be concluded that ID-AK-3 protocol is more secure compared to ID-AK-1 and ID-AK-2 protocols. These three protocols have been compared with the TAK protocols [15] and the Joux protocol. TAK protocols require deployed PKI to authenticate the long-term public keys, whilst this system uses an ID-based system. Though the

computations are slightly more in ID-AK-3 protocol compared to the other two protocols ID-AK-1, and ID-AK-2, it is found to be more secure, and also it is seen that it is computationally efficient compared to the existing ID-based one-round tripartite authenticated key agreement protocol by Zhang, Liu and Kim [7].. ID-AK-3 protocol is compared to the existing ID-based one-round authenticated key agreement protocol with pairings [7] and is found to be more efficient computationally.

## 11. References


1. A. Joux. A one round protocol for tripartite Diffie-Hellman. In W. Bosma, editor, *Proceedings of Algorithmic Number Theory Symposium*. ANTS IV, volume 1838 of Lecture notes in Computer Science , pages 385-394, Springer-Verlag, 2000.
2. A. Menezes, P.C. Van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, 1997
3. A. Shamir, *Identity based cryptosystems and signature schemes*. Advances in Cryptology – Proceedings of Crypto' 84.
4. C. Cocks. *An Identity based encryption scheme based on quadratic residues*. To appear in Cryptography and Coding, 2001.
5. D. Boneh, *The Decision Diffie-Hellman Problem*. In Proceesings of the 3rd Algorithmic Number theory Symposium, Lecture Notes in Computer Science, Volume 1423, Springer, pp 48 – 63, 1998
6. D. Boneh and M. Franklin. *Identity-based encryption from the Weil Pairing*. In Advances in Cryptology – CRYPTO 2001, Springer-Verlag LNCS 2139, 213-229, 2001.
7. Fangguo Zhang, Shengli Liu and Kwangjo Kim, *ID-based one-round authenticated tripartite key agreement protocol with pairings*, Cryptology eprint Archive, Report 2002/122. http://eprint.iacr.org/
8. IEEE P1363, *Standard Specifications for Public-Key Cryptography*, working draft, July 1998.
9. J.Horowitz and B.Lynn, *Toward hierarchical identity-based encryption*, Proc. of Eurocrypt 2002, LNCS 2332, pp 466-481, Springer-Verlag, 2002.
10. L. Law, A. J. Menezes, M. Qu, J. Solinas, and S. Vanstone. *An efficient protocol for authenticated Key agreement*. Technical Report CORR 98-05, Dept. of C & O, Univ. of Waterloo, 1998.
11. M. Burmester and Y. Desmedt. *A secure and efficient conference Key distribution system*. In A. De Santis, editor, Advances in Crptology EUROCRYPT ' 94,Workshop on the theory and Application of Cryptographic Techniques, volume 950 of Lecture notes in Computer Science, pages 275-286, Springer-Verlag, 1995.
12. N.P. Smart. *The discrete logarithm problem on elliptic curves of trace one*. preprint, 1997.
13. N.P. Smart. *An Identity based authenticated Key Agreement protocol based on the Weil Pairing*. Cryptology ePrint Archive, Report 2001/111, 2001. http://eprint.iacr.org/.
14. R. Harasawa, J. Shikata, J. Suzuki, and H. Imai. *Comparing the MOV and FR reductions in elliptic curve cryptography*. In J. Stern, editor, Advances in Cryptology, EUROCRYPT' 99, Vol. 1592 of LNCS, pg. 190-205, Springer 1999.
15. Sattam S. Al-Riyami, Kenneth G. Paterson, *Authenticated Three Party Key Agreement Protocols from* Pairings, Information security group, Royal Holloway, University of London, March 2002.
16. V. Miller, *Use of Elliptic curves in Cryptography*. In H. Williams, editor, Advances in Cryptology, CRYPTO' 85, VOLUME 128 OF lecture notes in Computer Sciencepages 417 – 428, Springer, 1986

17. W. Diffie and M. Hellman. *New directions in cryptography*. IEEE Trans. Info. Th., 22, 644-654, 1976.

**Appendix A.**

The comparison of the security attributes of these protocols can be shown as follows (taken from [13] )

| | Joux | TAK-1 | TAK-2 | TAK-3 | TAK-4 |
|---|---|---|---|---|---|
| Implicit Key authentication | No | Yes | Yes | Yes | Yes |
| Known session Key secure | No | No | Yes | Yes | Yes |
| Perfect forward secure | n/a | Yes | No(i) | No(i i) | Yes |
| Key compromise impersonation secure | n/a | No | Yes | Yes | Yes |
| Unknown Key share secure | No | Yes(iii) | Yes(i v) | Yes(i v) | Yes(iii) |

Table A: Comparison of security attributes of TAK protocols and Joux protocol.

(i) Not forward secure when a fatal compromise occurs on all three long-term secret Keys, but still forward secure for a compromise of two or less such Keys.

(ii) Not forward secure when two long-term secret Keys are compromised, but still forward secure if only one is compromised.

(iii) If the CA checks that public Keys are only registered once, and if inconvenient use (iv)

(iv) If the CA verifies that each user is in possession of the long-term secret Key corresponding to his public Key.