Comparative Study of Tripartite Identity-Based Authenticated Key Agreement Protocols

Marko Hölbl, Tatjana Welzer and Boštjan Brumen
Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova ulica 17, 2000 Maribor, Slovenia
marko.holbl@uni-mb.si

Keywords: authentication, identity-based, key agreement, pairing, security

Received: April 8, 2008

Key agreement protocols are used to exchange keys between two or multiple entities. The exchanged key can be later used to assure confidentiality through encryption. Additionally authenticated key agreement protocols offer implicit authentication. In this paper we conduct a security and efficiency comparison of tripartite authenticated identity-based key agreement protocols and review all of the protocols from the group. From the security perspective the protocols are compared with respect to the level to which they comply with defined security properties for authenticated key agreement protocols and the number of known attacks, whereas from the efficiency perspective the protocols are compared regarding computational effort. The comparative study enables in-depth analysis of existing protocols and the development of new ones.

Povzetek: Podana je primerjava protokolov za izmenjavo ključev.

1 Introduction

In key agreement protocols two or more entities agree upon a session key to be used for assuring a confidentiality or similar cryptographic goals. In 1976, Diffie and Hellman proposed the first key agreement protocol [13]. However, the basic Diffie-Hellman protocol does not authenticate the two communication entities, thus is susceptible to the man-in-the-middle attack. Later, different approaches and protocols have been developed to solve the problem [20, 12].

A research direction in the field of key agreement protocols are key agreement protocols for multi-party settings. A special case of multi-party key agreement protocols are tripartite (or three-party) protocols, which are of special interest as they are applicable to many practical scenarios such as e-commerce (two users and a merchant). Moreover, they implementation is easier and often more efficient than in case of multi-party protocols which are often very complex. The pioneer work by Joux [17] has shown how to implement a tripartite key agreement protocol employing pairings. In the protocol only one broadcast is required for each entity. However, just like the basic Diffie-Hellman protocol, Joux's protocol does not provide authentication of the three communicating entities and thus is vulnerable to the man-in-the-middle attack. To solve the problem with Joux's protocol, Al-Riyami et al. presented several protocols [1] which assure authenticity through use of certificates issued by a Certificate Authority (CA). The session keys are generated by both ephemeral (short-term) keys and static (long-term) keys. The signature of the CA assures that only the entities who possess the static keys are able to compute the session keys. However, in a certificate system, before using the public key of a user, the participants must first verify the certificates which requires a large amount of computing time and storage. The set of key pairs, certificates and certification authorities is referred to as public key infrastructure (PKI).

As an alternative to certificate-based PKIs, Shamir introduced the concept of an identity-based cryptosystem [24] in which the user's public key is an easily calculated function of her identity (e.g. social security number), while the user's private key is calculated by a trusted authority referred to as Key Generation Center (KGC). Shamir provided the first identity-based key construction based on the RSA problem, and presented an identity-based signature scheme [24]. The identity-based public key cryptosystem simplifies the process of key management, therefore can be an alternative for certificate-based public key infrastructure (PKI). In such cryptosystems, entity A can send encrypted messages to entity B by using her identity information even before B obtains her private key from the KGC. Hence, the idea also provides a way to construct authenticated key agreement protocols.

Recently, bilinear pairings have found positive application in cryptography [3, 6, 17, 29]. They can also be applied for constructing identity-based cryptographic protocols. Many identity-based cryptographic protocols for two and three-party setting have been proposed using the bilinear pairings. Some examples are Boneh-Franklin's identity-based encryption scheme [3], identity-based authentication key agreement protocol by Smart [28], McCullagh-Barreto [18] and several identity-based signatures schemes [29, 23, 9].

In this paper we will conduct a comparative study of identity-based authenticated key agreement protocols using

pairing operations for three-party settings. As far as we are aware, no tripartite identity-based authenticated key agreement protocol without pairings were proposed and that is why the comparative study includes only protocols employing pairing operations. In addition, we review all the protocols. The comparative study of the protocols is conducted as to security and efficiency. Both comparisons will be conducted using defined criteria. Thus the security criteria is defined by the fulfillment of security properties as described in [4, 7] and existence of attacks on the protocols. The efficiency comparison is realized using efficiency criteria; i.e., the number of computational operations required by a protocol. Even if a protocol fulfills all the security properties, its usage is questionable in case attacks for the protocol were published.

The rest of the paper is organized as follows: the next section briefly explains the identity-based public key infrastructure and the corresponding concepts (bilinear maps, the Weil pairing and the associated computational problems). Section 3 gives details on the security properties desired for a sound authenticated key agreement protocol. In Section 4, tripartite identity-based authenticated key agreement protocols using pairing operations are reviewed. For every protocol a description of the phases, security and efficiency discussion are given. A comparative study of the reviewed protocols regarding security and efficiency is conducted in section 5. Finally, a conclusion is made in section 6.

2 Identity-based Public Key Infrastructure employing pairing

In this section, we briefly describe the basic definition and properties of the bilinear pairing, the Weil pairing and the computational problems which form the basis for identity-based public key infrastructure employing pairings.

Traditional PKI (public key infrastructure) is expensive mainly because of the infrastructure needed to manage and authenticate public keys, and the difficulty in managing multiple communities. It is not believed that identity-based public key cryptography would replace the conventional PKIs, but can be rather seen as an alternative solution. In identity-based public key cryptography, one's public key is predetermined by information that uniquely identifies them. The idea of this concept, that was first proposed by Shamir [24], was to simplify certificate management in e-mail systems. When A sends e-mail to B, she encrypts the message using the public key string of B's e-mail (e.g. Bob@email.com). No public key certificate for B has to be obtained by A. When B receives the encrypted mail she contacts the key generation center (KGC), authenticates herself and thus can obtain the private key from the KGC, which enables her to decrypt the e-mail. In contrast to existing PKI, A is able to send encrypted mail to B even if B has not setup her public key certificate yet. A special case of identity-based public key cryptography (PKC) can be implemented using bilinear pairings, which will be described next.

2.1 Bilinear Maps

In this section we describe bilinear maps, pairings and their properties. More details can be found in Joux [17] and Boneh-Franklin [3].

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ denote two groups of prime order q. $\mathbb{G}_1$ is an additive group and $\mathbb{G}_2$ a multiplicative group. Let P be a generator of $\mathbb{G}_1$. A pairing is a computable bilinear map between these two groups. Two pairings have been studied for cryptographic use, namely the Weil pairing [19, 27, 31, 3] and the Tate pairing [14, 15, 16].

For our purpose, let $\hat{e}$ denote a general bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, which satisfies the following three properties:

1. *Bilinear:* If $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$, then $\hat{e}(aP, bQ) = e(P, Q)^{ab}$.
2. *Non-degenerative:* There exist non-trivial points $P, Q \in \mathbb{G}_1$ both of order q such that $\hat{e}(P, Q) \neq 1$.
3. *Computable:* If $P, Q \in \mathbb{G}_1$, $\hat{e}(P, Q) \in \mathbb{G}_2$ is efficiently computable (in polynomial time).

We say that $\mathbb{G}_1$ is a bilinear group if the group action in $\mathbb{G}_1$ can be computed efficiently and there exists a group $\mathbb{G}_2$ and an efficiently computable bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ as above. Weil and Tate pairings associated with super singular elliptic curves or Abelian varieties can be modified in order to create such bilinear maps. Concrete examples and details are given in [3], [17], [5].

2.2 The Weil Pairing

Let $\mathbb{G}_1$ be a subgroup of the group of points on the Elliptic curve E over the finite field $\mathbb{F}_q$. Let the order of $\mathbb{G}_1$ 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 $\mathbb{F}_q$. The Weil pairing [31, 3] is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ which satisfies the properties given in section 2.1 (bilinearity, non-degeneration and computability).

2.3 Computational Problems

Many pairing-based cryptographic protocols are based on the hardness of the BDHP (Bilinear Diffie-Hellman Problem) for their security [3, 10]. Some computational problems related to the elliptic curve cryptography:

– **Bilinear Diffie-Hellman Problem (BDHP)**
Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of prime order q. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, be a bilinear map and let P be a generator of $\mathbb{G}_1$. The BDH problem in $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e} \rangle$ is defined as: Given $(P, xP, yP, zP) \in$

$\mathbb{G}_1$ for some x, y, z chosen at random from $\mathbb{Z}_q^*$, compute $\hat{e}(P,P)^{xyz} \in \mathbb{G}_2$.

- **Discrete Logarithm Problem (DLP)**
  Given $P, Q \in \mathbb{G}_1$, find an integer n such that $P = nQ$.

- **Computational Diffie-Hellman Problem (CDHP)**
  Given a tuple $(P, aP, bP) \in \mathbb{G}_1$ for $a, b \in \mathbb{Z}_q^*$, find the element abP.

3 Security properties

In order to get a sound key agreement protocol, we need to define properties, which are described in detail in [4]. Here we assume A, B and C are three honest entities. It is desired for authenticated key agreement protocols to possess the following security attributes [4, 7, 10, 22]:

- **Implicit Key Authentication.** A key agreement protocol provides *implicit key authentication* if each entity A is assured that no other entity besides entities B and C can determine the value of a particular secret key. A protocol which provides implicit key authentication for entities A, B, and C is called an authenticated key agreement protocol (AK).

- **Known-Key Security.** In each round of a key agreement protocol, A, B and C should generate a unique secret key. Each key generated in one protocol round is independent and should not be exposed if other secret keys are compromised, i.e. the compromise of one session key should not compromise other session keys.

- **Forward Secrecy.** If the long-term private keys of one or more of the entities are compromised, the secrecy of previously established session keys should not be affected. We say that a system has ***partial forward secrecy*** if some but not all of the entities' long-term keys can be corrupted without compromising previously established session keys, and we say that a system has ***perfect forward secrecy*** if the long-term keys of all the entities involved may be corrupted without compromising any session key previously established by these entities.

- **Unknown Key-Share resilience.** After the protocol run, entity A believes she shares a key with B and C, whereas B and C mistakenly believe that the key is instead shared with an adversary. Therefore, a sound authenticated key agreement protocol should prevent the unknown key-share situation.

- **Key-Compromise Impersonation.** Assume that A, B and C are three principals. Suppose A's secret key is disclosed. Obviously, an adversary who knows this secret key can impersonate A to B and C. However, it is desired that this disclosure does not allow the adversary to impersonate other entities (e.g. B and C) to the real A.

- **Key Control.** The key should be determined jointly by all A, B and C. Neither A, B nor C can control the key alone.

4 Review of tripartite identity-based authenticated key agreement protocols employing pairings

In this section we will review tripartite identity-based authenticated key agreement protocol employing pairings. Some protocol derive multiple keys for later encryption, like [32], [30], whereas other compute just one key [21], [21], [26] and simplified variant of [32]. All protocols consist of three phases, namely the *system setup*, *private key extraction* and *key agreement* phase. Furthermore, each protocol requires three entities (e.g. A, B and C) and a key generation center (KGC) that is relied upon to create and deliver private keys to entities and to not abuse its knowledge of those keys.

When describing the *key agreement* phase we will give only examples of computations performed by entity A. Observe that entities B and C perform almost identical computational operations in the particular key agreement phases of the reviewed protocols, except for the change of indexes in the equations.

All key agreement scheme feature a key derivation functions kdf defined as $kdf = \mathbb{F}_q^* \rightarrow \{0,1\}^*$. The key derivation function is needed in every scheme because the session keys are subsequentially used for encrypting data with it usually realized using block ciphers. These require bit strings as keys.

All of the reviewed protocols features the same *system setup* and *private key extraction* phases. Therefore the first two phases will be reviewed here, whereas the *key agreement* phase will be described for every protocol separately.

System Setup. The Key Generation Center (KGC) constructs two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ and a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Next it computes a cryptographic hash function $H : \mathbb{Z}_q^* \rightarrow \mathbb{G}_1$, a generator (primitive root) $P \in \mathbb{G}_1$, a random integer $s \in \mathbb{Z}_q^*$ as KGC's private key and KGC's public key as $P_{KGC} = sP$. All elements are of order q. Finally, the following parameters are published: $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{KGC}, H \rangle$ and the master key is s.

Private Key Extraction. For user with identity ID_i the public key is derived as $Q_i = H(ID_i)$ and the private key as $S_i = sQ_i$. Both parameters are computed by the KGC and afterwards S_i is issued to the entity via a secure channel.

4.1 Zhang-Liu-Kim's Protocol

In 2002, Zhang, Liu and Kim proposed the first tripartite identity-based authenticated key agreement protocol [32]. Each instance of the protocol results in multiple session keys. The way the session key is produced makes use of the Weil pairing and the identity-based static public keys.

Key Agreement. To establish a session key, the three communication entities, A, B and C must proceed as follows.

1. $A \rightarrow B, C$: $P_A = aP$, $P_A^{'} = a^{'}P$, $T_A = H(P_A, P_A^{'})S_A + aP_A^{'}$.
2. $B \rightarrow A, C$: $P_B = bP$, $P_B^{'} = b^{'}P$, $T_B = H(P_B, P_B^{'})S_B + bP_B^{'}$.
3. $C \rightarrow A, B$: $P_C = cP$, $P_C^{'} = c^{'}P$, $T_C = H(P_C, P_C^{'})S_C + cP_C^{'}$.

A verifies: $\hat{e}(T_B + T_C, P) = \hat{e}(H(P_B, P_B^{'})Q_B + H(P_C, P_C^{'})Q_C, P_{KGC}) \cdot \hat{e}(P_B, P_B^{'}) \cdot \hat{e}(P_C, P_C^{'})$.
If the above equation holds, then A computes the 8 session keys:
$K_A^{(1)} = \hat{e}(P_B, P_C)^a$, $K_A^{(2)} = \hat{e}(P_B, P_C^{'})^a$, $K_A^{(3)} = \hat{e}(P_B^{'}, P_C)^a$, $K_A^{(4)} = \hat{e}(P_B^{'}, P_C^{'})^a$, $K_A^{(5)} = \hat{e}(P_B, P_C)^{a^{'}}$, $K_A^{(6)} = \hat{e}(P_B, P_C^{'})^{a^{'}}$, $K_A^{(7)} = \hat{e}(P_B^{'}, P_C)^{a^{'}}$, $K_A^{(8)} = \hat{e}(P_B^{'}, P_C^{'})^{a^{'}}$.

Each entity takes the eight values $K_{ID}^{(i)}, i = 1, 2, \ldots 8$, as the final session keys. The correctness of the protocol can be easily checked by the bilinear property of the pairing: $K_A^{(1)} = \hat{e}(P_B, P_C)^a = \hat{e}(abP, cP) = \hat{e}(aP, cP)^b = \hat{e}(P_A, P_C)^b = K_B^{(1)} = \hat{e}(bP, aP)^c = \hat{e}(P_B, P_A)^c = K_C^{(1)}$. Similarly, we get $K^{(i)} = K_A^{(i)} = K_B^{(i)} = K_C^{(i)}$, $i = 2, 3, \ldots 8$.

Security and attacks. From the security point of view the protocol has the following security properties: known key security, perfect forward secrecy, key control, key-compromise impersonation and unknown key-share.

Shim and Woo developed an attack on Zhang-Liu-Kim's protocol [30]. They showed that the protocol is insecure against an unknown key-share (UK-S) which enables the adversary to make entity a believe she shares a key with B and C, whereas B and C mistakenly believe that the key is instead shared with an adversary.

In the UK-S attack scenario B and C compute the same 8 session keys, while A computes his session keys, from which 4 of 8 keys are equal. Thus A, B and C share the first four session keys and A thinks that the session keys are shared with B and C, while B (resp. C) mistakenly believes that she shares the keys with E and C (resp. E and B). Moreover, both A and B come to share the same eight session keys. The weakness of the protocol against the unknown-key share attack is due to the fact that anyone who does not know an ephemeral private key a corresponding to $P_A = aP$ can generate her own signature on P_A and the lack of explicitness in cryptographic messages, i.e., the signed messages of the protocol do not include some information to confirm that the sender is identical to a genuine communicating entity.

Efficiency. In the protocol each entity uses 4 pairings for verification of the broadcast messages from the other two entities, and 4 pairings to computes the 8 session keys. Additionally, each entity has to compute 6 scalar multiplication and 8 exponentiations. Because there are 8 keys derived, the computational overhead per derived key for each entity is 1 pairing operation, $0,75$ scalar multiplication and 1 exponentiations.

4.2 Simplified Zhang-Liu-Kim's Protocol

In the same paper [32], Zhang-Liu-Kim also published a simplified version of identity-based tripartite authenticated key agreement, i.e., the 3 entities agree to 1 session key instead of 8 keys.

Key Agreement. A, B and C compute and broadcast the following:

1. $A \rightarrow B, C$: $P_A = aP, T_A = H(P_A)S_A + aP_A$.
2. $B \rightarrow A, C$: $P_B = bP, T_B = H(P_B)S_B + bP_C$.
3. $B \rightarrow A, B$: $P_C = cP, T_C = H(P_C)S_C + cP_C$.

A verifies: $\hat{e}(T_B + T_C, P) = \hat{e}(H(P_B)Q_B + H(P_C)Q_C, P_{KGC})\hat{e}(P_B, P_B)\hat{e}(P_C, P_C)$.
If the above equation holds, then A computes: $K_A = \hat{e}(P_B, P_C)^a$.

Then the session key is $K_A = K_B = K_C = \hat{e}(P, P)^{abc}$.

Security and Attacks. The authors claim that their protocol has the following security properties: known key security, perfect forward secrecy, key control, key-compromise impersonation and unknown key-share. No attacks on the protocols are known so far.

Efficiency. With the simplified version of the protocol, an entity needs to compute 5 pairings, 4 for verification and 1 for the generation of the session key.

4.3 Nalla-Reddy's Protocol

Nalla and Reddy proposed their identity-based tripartite authenticated key agreement protocol employing pairings in 2003 [22]. They employ ideas by Shim two-party identity-base authenticated key agreement protocol employing pairings [25] and Joux's tripartite identity-based authenticated key agreement protocol [17].

The authors present 3 protocols: ID-AK-1 (Identity-based Authenticated Key Agreement Protocol 1), ID-AK-2 and ID-AK-3, which will be reviewed separately.

Key Agreement
ID-AK-1. Each user generates a random number a, b and

c. The ephemeral (or short term) public keys would be aP, bP and cP, and the ephemeral or short term private keys would be a, b and c.

1. $A \rightarrow B, C$: aP.
2. $B \rightarrow A, C$: bP.
3. $B \rightarrow A, B$: cP.

User A computes $K_A = \hat{e}(bP, cP)^a \cdot \hat{e}(Q_B, P_{KGC}) \cdot \hat{e}(Q_C, P_{KGC}) \cdot \hat{e}(S_A, P) = \hat{e}(P, P)^{abc} \cdot \hat{e}(Q_A, P)^s \cdot \hat{e}(Q_B, P)^s \cdot \hat{e}(Q_C, P)^s$.

The session key is computed as $K_{ABC} = \hat{e}(P, P)^{abc} \cdot \hat{e}(Q_A, P)^s \cdot \hat{e}(Q_B, P)^s \cdot \hat{e}(Q_C, P)^s = \hat{e}(P, P)^{abc} \cdot \hat{e}((Q_A + Q_B + Q_C), P_{KGC})$ and hence depends on the identities of the three entities Q_A, Q_B, Q_C, and the three ephemeral private keys a, b and c.

ID-AK-2. Similarly as in ID-AK-1, each user generates a random number a, 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.

1. $A \rightarrow B, C$: aP_{KGC}.
2. $B \rightarrow A, C$: bP_{KGC}.
3. $B \rightarrow A, B$: cP_{KGC}.

User A computes $K_A = \hat{e}(aS_A, P) \cdot \hat{e}(Q_B, bP_{KGC}) \cdot \hat{e}(Q_C, cP_{KGC}) = \hat{e}(aQ_A + bQ_B + cQ_C, sP)$ and user B computes $K_B = \hat{e}(Q_A, aP_{KGC}) \cdot \hat{e}(bQ_B, P) \cdot \hat{e}(Q_C, cP_{KGC}) = \hat{e}(aQ_A + bQ_B + cQ_C, sP)$ Similarly C computes $K_C = \hat{e}(Q_A, aP_{KGC}) \cdot \hat{e}(Q_B, bP) \cdot \hat{e}(cS_C, P_{KGC}) = \hat{e}(aQ_A + bQ_B + cQ_C, sP)$.

Hence the session key is computed as $K_{ABC} = K_A = K_B = K_C = \hat{e}(aQ_A + bQ_B + cQ_C, sP)$.

ID-AK-3. Each user generates random a, b, $c \in \mathbb{Z}_q^*$, which are the ephemeral private keys of A, B and C. The data flows of the protocol are as follows.

1. $A \rightarrow B$: aP, aQ_C; $A \rightarrow C$: aP, aQ_B;
2. $B \rightarrow A$: bP, bQ_C; $B \rightarrow C$: bP, bQ_A;
3. $C \rightarrow A$: cP, cQ_B; $C \rightarrow B$: cP, cQ_A;

A computes the key $K_A = \hat{e}(a(Q_B + Q_C), P_{KGC}) \cdot \hat{e}(S_A, (bP + cP)) \cdot \hat{e}(bQ_C, P_{KGC}) \cdot \hat{e}(cQ_B, P_{KGC})$.

Hence the session key is derived as $K_{ABC} = K_A = K_B = K_C = \hat{e}(a(Q_B + Q_C) + b(Q_A + Q_C) + c(Q_A + Q_B), sP)$.

Security and Attacks. The authors claim different security properties fulfillment for each of the three protocols. *ID-AK-1* complies to forward secrecy, key control and unknown key-share. The *ID-AK-2* protocol conforms to the properties of forward secrecy, key control, key-compromise impersonation and unknown key-share. The third protocol, *ID-AK-3*, fulfills the following security properties: known key security, forward secrecy, key control, key-compromise impersonation and unknown key-share.

However, passive attacks on ID-AK-2 and ID-AK-3 protocols were published by Chen [8] in 2003. In a passive attack, the adversary is able to derive the session keys just eavesdropping on the communication line and use the intercepted data to compute the key.

The passive attack on ID-AK-2 is carried out as follows: since in the ID-AK-2 protocol the key is computed as $K_{ABC} = \hat{e}(aQ_A + bQ_B + c_QC, sP) = \hat{e}(Q_A, aP_{pub}) \cdot \hat{e}(Q_B, bP_{pub}) \cdot \hat{e}(Q_C, cP_{pub})$ and Q_A, Q_B, Q_C and P_{KGC} are publicly known, a passive attacker can eavesdrop aP_{pub}, bP_{pub} and cP_{pub}, and is able to compute K_{ABC}.

Additionally to the presented attack on ID-AK-2, Chen also demonstrated attack on ID-AK-3 [8]. In the protocol the key is computed as $K_{ABC} = \hat{e}(a(Q_B + Q_C) + b(Q_A + Q_C) + c(Q_A + Q_B), sP)$. Q_A, Q_B, Q_C and P_{KGC} are publicly known, and a passive attacker can know them. In a protocol run, the passive attacker can eavesdrop aQ_b, aQ_C, bQ_C, bQ_A, cQ_A, and cQ_B and is able to compute K_{ABC}.

Additionally, Shim published a man-in-the-middle attack on Nalla-Reddy's ID-AK-1 protocol [26]. In the attack the adversary is able to compute and share session keys with all three entities by intercepting the original messages aP, bP, cP and inserting her own messages $a^{'}P$, $b^{'}P$, $c^{'}P$. At the end E is can compute K_A, K_B and K_C and therefore shares a key with A, B and C.

Efficiency. Because of the different computational task performed by each of the protocol (ID-AK-1, ID-AK-2 and ID-AK-3), we will discuss efficiency of each of them separately.

In the *ID-AK-1* protocol each user needs to compute 4 Weil pairings and 1 scalar multiplication. However, 3 of the Weil pairings can be precomputed and only 1 pairing needs to be computed for each session. To sum up, each entity has to perform: 5 pairing operation and 5 scalar multiplication.

In the second protocol, *ID-AK-2*, each user is required to compute 2 scalar multiplications and 3 Weil pairings.

The last of the three presented protocols (*ID-AK-3*) is role symmetric since each participant executes the same number of operations. It requires each participant to compute 2 additions, 4 scalar multiplications, and 4 Weil pairings.

4.4 Nalla's Protocol with Signatures

Nalla, proposed another tripartite key agreement protocol for identity-based systems employing identity-based signatures in 2003 [21]. Because some identity-based tripartite key agreement protocols proposed in Nalla-Reddy's previous work [22] suffered passive attacks, and Joux' s protocol [17] suffered man-in-the-middle attack, Nalla proposed a new protocol including signature in Joux's protocol. It resulted in much simpler identity-based key agreement protocols.

Key Agreement. Let A, B and C be the three parties wishing to compute a session key. First, A, B and C select

random number a, b and $c \in \mathbb{Z}_q^*$ and perform the following actions:

1. $A \rightarrow B, C$: $P_A = aP, T_A = a^{-1}(H(P_A)S_A)$
2. $B \rightarrow A, C$: $P_B = bP, T_B = b^{-1}(H(P_B)S_B)$
3. $C \rightarrow A, B$: $P_C = cP, T_C = c^{-1}(H(P_C)S_C)$

A verifies: $\hat{e}(P_B, T_B) \cdot \hat{e}(P_C, T_C) = \hat{e}(P_{KGC}, H(P_B)Q_B + H(P_C)Q_C)$ and computes $K_A = \hat{e}(P_B, P_C)^a = \hat{e}(P, P)^{abc}$.

This verification ensures the authenticity of the senders. The session key is the value $K_{ABC} = K_A = K_B = K_C = \hat{e}(P, P)^{abc}$.

Security and Attacks. The author claims that his protocol has the following security properties: known key security, perfect forward secrecy, key control, key-compromise impersonation and unknown key-share.

In 2003, Shim published an impersonation attack on the Nalla's protocol with signatures [26]. According to [26], the adversary is able to broadcast such messages with help of which she can impersonate an entity (in the paper an example for entity A is given). The messages sent by the adversary E impersonating A are successfully verified by B and C. Additionally, E can compute the session key K_A and finally succeed to impersonate A to B and C and compute the session key.

Shim claims [26] that Nalla's protocol is insecure against the man-in-the-middle attack because of the impersonation attack. She further claims that the weakness of the protocol against the attack is due to the fact that anyone who does not know each otherŠs private key (S_{ID}) can generate a valid pair (P_{ID}, T_{ID}).

Efficiency. Regarding efficiency, in each protocol run the following operations have to be computed: 4 pairing operation, 5 scalar multiplication and 1 exponentiations.

4.5 Shim's Protocol with Signatures

Due to the flaws in Nalla-Reddy's and Nalla's protocols, Shim proposed a modified identity-based tripartite key agreement protocol with signatures [26].

Key Agreement. Let A, B and C be the three parties wishing to compute a session key. A, B and C select random number a, b and $c \in \mathbb{Z}_q^*$ and exchange the following messages:

1. $A \rightarrow B, C$: $P_A = aP, T_A = H(P_A)S_A + aP_{KGC}$
2. $B \rightarrow A, C$: $P_B = bP, T_B = H(P_B)S_B + bP_{KGC}$
3. $C \rightarrow A, B$: $P_C = cP, T_C = H(P_C)S_C + cP_{KGC}$

A verifies: $\hat{e}(T_B + T_C, P) = \hat{e}(P_{KGC}, H(P_B)Q_B + H(P_C)Q_C + P_B + P_C)$.
If the equation holds, then A computes $K_A = \hat{e}(P_B, P_C)^a = \hat{e}(P, P)^{abc}$.

This verification ensures the authenticity of the senders. The session key is the value $K_{ABC} = K_A = K_B = K_C = \hat{e}(P, P)^{abc}$.

Security and Attacks. From the security perspective the protocol features known key security, perfect forward secrecy, key control, key-compromise impersonation and unknown key-share. No attacks on the protocols are known so far.

Efficiency. In the reviewed protocol the computation effort includes 3 pairing operations, 5 scalar multiplications and 1 exponentiation.

4.6 Shim-Woo's Protocol

Recently, Shim and Woo proposed a more efficient identity-based tripartite multiple-key agreement protocol which satisfies all the required security attributes and does not use any one-way hash functions.

Key Agreement. Suppose three communication entities, A, B and C want to establish a secret session key. To achieve this, they perform:

1. $A \rightarrow B, C$: $P_A = aP, P_A^{'} = a^{'}P, T_A = S_A + a^2P + a^{'}P_{KGC}$.
2. $B \rightarrow A, C$: $P_B = bP, P_B^{'} = b^{'}P, T_B = S_B + b^2P + b^{'}P_{KGC}$.
3. $B \rightarrow A, B$: $P_C = cP, P_C^{'} = c^{'}P, T_C = S_C + c^2P + c^{'}P_{KGC}$.

A verifies $\hat{e}(T_B + T_C, P) = \hat{e}(Q_B + Q_C + P_B^{'} + P_C^{'}, P_{KGC}) \cdot \hat{e}(P_B, P_B) \cdot \hat{e}(P_C, P_C)$.
If the above equation holds, then A computes the 8 session keys:
$K_A^{(1)} = \hat{e}(P_B, P_C)^a$, $K_A^{(2)} = \hat{e}(P_B, P_C^{'})^a$, $K_A^{(3)} = \hat{e}(P_B^{'}, P_C)^a$, $K_A^{(4)} = \hat{e}(P_B^{'}, P_C^{'})^a$, $K_A^{(5)} = \hat{e}(P_B, P_C)^{a^{'}}$, $K_A^{(6)} = \hat{e}(P_B, P_C^{'})^{a^{'}}$, $K_A^{(7)} = \hat{e}(P_B^{'}, P_C)^{a^{'}}$, $K_A^{(8)} = \hat{e}(P_B^{'}, P_C^{'})^{a^{'}}$.

Each entity takes the eight values $K_{ID}^{(i)}$, $i = 1, 2, \ldots 8$, as the final session keys. The correctness of the protocol can be easily checked by the bilinear property of the pairing:
$K_A^{(1)} = \hat{e}(P_B, P_C)^a = \hat{e}(abP, cP) = \hat{e}(aP, cP)^b = \hat{e}(P_A, P_C)^b = K_B^{(1)} = \hat{e}(bP, aP)^c = \hat{e}(P_B, P_A)^c = K_C^{(1)}$. Similarly, we get $K^{(i)} = K_A^{(i)} = K_B^{(i)} = K_C^{(i)}$, $i = 2, 3, \ldots 8$.

Security and Attacks. From the security point of view, the protocol features known key security, perfect forward secrecy, key control, key-compromise impersonation and unknown key-share.

However, Chou-Lin-Shiu published an impersonation attack on Shim-Woo's protocol [11] in 2005. As a result, the adversary can share the 4 keys $K^{(1)}$, $K^{(2)}$, $K^{(5)}$, $K^{(6)}$ of the 8 session keys. Under this situation, two of the three entities (e.g. A and C) involved in the protocol, think

these 4 session keys are shared the third entity (e.g. B), but indeed, they are shared with the adversary. Besides, both A and C come to share the same 8 session keys. Thus, the impersonation attack on 4 of the 8 session keys can be successfully mounted. More precisely, the adversary can use the 4 session keys to communicate with A and C, and he can have one half of the probability to realize what the communication contents are between A and C.

Efficiency. According to the authors, their protocol requires the following computational operations: 8 pairing operation, 4 scalar multiplication, 8 exponentiations. Additionally, since 8 key are derived, the effort per key for each entity is 1 pairing operation, $0,5$ scalar multiplication, 1 exponentiations.

5 Comparative study

In the following section we compare the reviewed protocols with respect to security and performance. From the security point of view the criteria to compare security of the protocols is given by the extent to which a specific protocol fulfills the security properties as discusses in section 3. Additionally, attacks for each protocol are analyzed and included in the criteria for comparing security. From the performance point of view the criteria for comparing efficiency is defined as the number of computational operations required per protocol run.

In general authenticated key agreement protocols have to be secure and at the same time as efficient as possible. Therefore the security factor is more important when assessing and comparing the reviewed protocols.

5.1 Security Comparison

The security comparison of the reviewed protocols is conducted as to two criteria: the fulfillment of security properties as defined in section 3 and the existence of attacks due to errors in the design of the protocols. Often, an attack on a protocol results in the not-fulfillment of specific security properties, but please observe that this is not always the case, since ID-AK-1 and ID-AK-2 are susceptible to passive attack and yet this does not violate any security property.

Further please notice that Nalla-Reddy's ID-AK-1, ID-AK-2 and ID-AK-3 were broken due to various attacks (please refer to section 4), but are included in the comparison for completeness.

5.1.1 Security Properties

Table 1 summarizes the fulfillment of security properties for each of the reviewed protocols. For definition and details regarding the security properties the reader is referred to section 3.

The majority of protocols do not fulfill the security properties, with exception simplified ZLK and Shim's protocol. Additionally, ZLK's, the simplified ZLK and Shim-Woo's protocols offer prefect forward secrecy, whereas the rest has the property of partial forward secrecy. Nalla's and Shim-Woo's protocols do not fulfill particular security properties because of attacks (please refer to section 4 for further details).

Table 1: Security properties

Protocol	KKS	FS	KCI	UKS	KC
ZLK	+	+*	+	$-^a$	+
simplified ZLK	+	$+^*$	+	+	+
ID-AK-1	-	+	-	+	+
ID-AK-2	-	+	+	+	+
ID-AK-3	+	+	+	+	+
Nalla	+	+	$-^b$	+	+
Shim	+	+	+	+	+
Shim-Woo	+	$+^*$	$-^c$	+	+

KKS - Known-Key Secrecy
FS - Forward Secrecy
* - perfect forward secrecy
KCI - Key-Compromise Impersonation
UKS - Unknown Key-Share
KC - Key Control
a - as to Shim-Woo's unknown key-share attack [30]
b - Shim's Impersonation attack [26]
c - Chou-Lin-Chiu's Impersonation attack [11]

5.1.2 Known Attacks

Some of the reviewed protocols have been shown to have weaknesses, which were exploited for attacks. Table 2 sums up attack for the reviewed protocols. For two of the reviewed protocols (i.e., ZLK's and Shim's protocol) there are no known attacks. When a protocol suffers from attacks it lacks security and as a consequence sometimes does not

Table 2: Known attacks

Protocol	Attacks
ZLK	Shim-Woo's unknown key-share attack [30]
simplified ZLK	\
ID-AK-1	Shim's Man-in-the-middle attack [26]
ID-AK-2	Chen's Passive attack [8]
ID-AK-3	Chen's Passive attack [8]
Nalla	Shim's Impersonation attack [26]
Shim	\
Shim-Woo	Chou-Lin-Chiu's Impersonation attack [11]

fulfill a defined security property (see section 3).

5.2 Efficiency Comparison

The computations effort per user (number of computations performed) of the reviewed protocols is given in table 3. We compare operations which are expensive from the computational point of view - pairing operations, scalar scalar multiplications and exponentiation. Additions and hash operations are ignored since they are much less computationally expensive. As precomputing pairing operations increases the performance and lowers the computational effort, we also give data regarding precomputation. A precomputed pairing operations must only be carried out when the three entities conduct a key agreement for the first time and can be later omitted.

Table 3: Computation effort per user.

Protocol	PairOp	ScMul	Exp	PP
ZLK	8	6	8	0
simplified ZLK	5	5	1	0
ID-AK-1	4	1	0	3
ID-AK-2	3	2	0	0
ID-AK-3	4	4	0	0
Nalla	4	5	1	0
Shim	3	5	1	0
Shim-Woo	8	4	8	0

PairOp - pairing operations
ScMul - scalar multiplications in $\mathbb{G}_1$
Exp - exponentiation in $\mathbb{G}_2$
PP - pairings that can be pre-computed

Before discussing the efficiency of the reviewed protocols, it should be noted that according to [2], the effort to evaluate one pairing operation is approximately equal to the effort of computing three scalar multiplications. As can bee seen from table 3 the most efficient protocol is Nalla-Reddy's ID-AK-2, followed by Shim's protocol. In contrary, the least efficient protocol is the ZLK protocol resp. the simplified ZLK protocol.

The most robust and most efficient protocol from both the security and efficiency point of view is not straightforward. Due to the discussed attacks, we have to rule out all the protocols susceptible to attacks and not fulfilling all security requirements. This leaves us with only two protocols, namely the simplified ZLK protocol and Shim's protocol. When taking the results of the efficiency analysis into account Shim's protocols prevails as it offers best performance while fulfilling the desired security properties and immune to attacks.

6 Conclusion

Identity-based authenticated key agreement protocols can be an alternative for certificate-based protocols. This is true, especially when efficient key management and moderate security are required. In the paper we have made a review and comparative study of tripartite authenticated identity-based key agreement protocols using pairings. We have presented the state of the art in attacks on the reviewed protocols and conducted a comparative study regarding the fulfillment of security properties, attacks published and the computational effort required by each protocol. The prevailing protocol considering security and efficiency is Shim's protocol as it is efficient and at the same time offers all security properties. Future development of protocols must take the analysis results and attacks into account when developing new protocols.

References

[1] S. Al-Riyami, and K. Paterson (2002) Authenticated three party key agreement protocols from pairings, *Cryptology ePrint Archive, Report 2002/035*.

[2] P.S.L.M. Barreto, H.Y. Kim, B. Lynn, M. Scott (2002) Efficient algorithms for pairing-based cryptosystems, *Advances in Cryptology - Crypto'02, LNCS Vol. 2139*, Springer, UK, pp. 213 - 229 .

[3] D. Boneh, and M. Franklin (2003) Identity-based encryption from the Weil pairing, *Advances in Cryptology - Crypto'01, LNCS Vol. 2442*, Springer, UK, pp. 354-368.

[4] S. Blake-Wilson, D. Johnson, and A. Menezes (1992) Key agreement protocols and their security analysis (Extended abstract), *6th IMA International Conference on Cryptography and Coding, LNCS Vol. 1355*, Springer, UK, pp. 30-45.

[5] D. Boneh, B. Lynn, and H. Shacham (2002) Short signatures from the Weil pairing, *Advances in Cryptology - AsiacryptŠ01, LNCS 2248*, Springer, UK, pp. 514-532.

[6] D. Boneh, B. Lynn, and H. Shacham (2001) Short signatures from the Weil pairing, *Advances in Cryptology-Asiacrypt 2001, LNCS 2248*, Springer, UK, pp. 514-532.

[7] Bellare, M., Rogaway, P. (1993) Entity Authentication and Key Distribution, *Advances in Cryptology - CRYPTO '93*, Springer, UK, pp. 232-249.

[8] Chen, Z. (2003) Security analysis on Nalla-Reddy's ID-based tripartite authenticated key agreement protocols, *Cryptology ePrint Archive, Report 2003/103*.

[9] J.C. Cha, and J.H. Cheon (2002) An identity-based signature from gap Diffie-Hellman groups, *Cryptology ePrint Archive, Report 2002/018*.

[10] L. Chen, C. Kudla (2003) Identity Based Authenticated Key Agreement Protocols from Pairings, *16th IEEE Computer Security Foundations Workshop*, IEEE Press, USA, pp. 219-233.

[11] J.S. Chou, C.H. Lin, C.H. Chiu (2005) Weakness of Shim's New ID-based tripartite multiple-key agreement protocol, *Cryptology ePrint Archive, Report 2005/457*.

[12] R. Dutta, R. Barua (2005) Overview of Key Agreement Protocols, *Cryptology ePrint Archive, Report 2005/289*.

[13] W. Diffie, M. Hellman (1976) New directions in cryptography, *IEEE Transactions on Information Theory, vol.22, no.6*, IEEE Press, USA, pp. 644-654.

[14] G. Frey, M. Mller and H. Rock (1999) The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems, *IEEE Transactions on Information Theory, vol. 45, no.5*, IEEE Press, USA, pp. 1717-1719.

[15] S. Galbraith (2001) Supersingular curves in cryptography, *Advances in Cryptology - Asiacrypt' 01, LNCS Vol. 2248*, Springer, UK, pp. 495-513.

[16] S.D. Galbraith, K. Harrison, and D. Soldera (2002) Implementing the Tate Pairing, *5th International Symposium on Algorithmic Number Theory, LNCS Vol. 2369*, Springer, UK, pp. 324-337.

[17] A. Joux (2000) A one round protocol for tripartite Diffie-Hellman, *4th International Symposium on Algorithmic Number Theory, LNCS Vol. 1838*, Springer, UK, pp. 385-393.

[18] N. McCullagh, and P.S.L.M. Barreto (2005) A new two-party identity-based authenticated key agreement, *Topics in Cryptology - CT-RSA 2005, LNCS Vol. 3376*, Springer, UK, pp. 262-274.

[19] A. Menezes, T. Okamoto, and S. Vanstone (1993) Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Transactions on Information Theory Vol. 39*, IEEE Press, USA, pp. 1639-1646.

[20] A. Menezes, P.C. Van Oorschot, and S. Vanstone (1997) *Handbook of Applied Cryptography*, CRC Press, USA.

[21] Nalla, D. (2003) ID-based tripartite key agreement with signatures, *Cryptology ePrint Archive, Report 2003/144*.

[22] D. Nalla, K.C. Reddy (2003) ID-based tripartite key agreement with signatures, *Cryptology ePrint Archive, Report 2003/004*.

[23] K.G. Paterson (2002) ID-based signatures from pairings on elliptic curves, *Cryptology ePrint Archive, Report 2002/004*.

[24] A. Shamir (1985) Identity-Based Cryptosystems and Signature Schemes, *Advances in Cryptology - CRYPTO 84*, pp. 47-53 .

[25] K. Shim (2003) Efficient ID-based authenticated key agreement protocol based on Weil pairing, *Electronics Letters, Vol. 39, No. 8*, IEEE Press, USA, pp. 653-654.

[26] K. Shim (2003) Cryptanalysis of ID-based Tripartite Authenticated Key Agreement Protocols, *Cryptology ePrint Archive, Report 2003/115*.

[27] J.H. Silverman (1994) Advanced topics in the arithmetic of elliptic curves, *Graduate Texts in Mathematics Vol. 151*, Springer, UK.

[28] N.P. Smart (2002) Identity-based authenticated key agreement protocol based on Weil pairing, *Electronics Letters, Vol. 38, No. 13*, IEEE Press, USA, pp. 630-632.

[29] R. Sakai, K. Ohgishi, M. Kasahara (2000) Cryptosystems based on pairing, *Symposium on Cryptography and Information Security (SCIS2000)*, Japan.

[30] K. Shim, S. Woo (2005) Weakness in ID-based one round authenticated tripartite multiple-key agreement protocol with pairings, *Applied Mathematics and Computation, Vol. 166, No. 3*, Elsevier, USA, pp. 523-530.

[31] E. Verheul (2001) Evidence that XTR is more secure than supersingular elliptic curve systems, *Advances in Cryptology - EUROCRYPT 2001, LNCS Vol. 2045*, Springer, UK, pp. 195-210.

[32] F. Zhang, S. Liu, K. Kim (2002) ID-based one-round authenticated tripartite key agreement protocol with pairings, *Cryptology ePrint Archive, Report 2002/122*.