Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol I,
IMECS 2013, March 13 - 15, 2013, Hong Kong

New Authenticated Key Agreement Protocols

Mohamed Nabil, Yasmine Abouelseoud, Galal Elkobrosy, and Amr Abdelrazek

Abstract— **In this paper, new authenticated key agreement (AKA) protocols are proposed to be used by two entities and three entities in order to establish a common session key between these entities. This key is used later to encrypt the data exchanged between the entities to assure confidentiality over public insecure channels. Authenticated key agreement protocols additionally offer authentication; that is, verifying the identities of the entities involved in the protocol. The security properties of the proposed schemes are investigated and this revealed that they resist various attacks that can be mounted against a key agreement protocol promoting their use in practical scenarios such as secure remote access to a shared database.**

Index Terms—authentication; public key infrastructure (PKI); key agreement; security; bilinear maps

I. Introduction

Living in the information age, the deployment of security mechanisms has become an impelling need to protect the easy to manipulate digital data being exchanged over public insecure channels. Users acquiring digital services from remote servers, such as in mobile communications, need first to be checked for authorization to be granted access to network services and then the data transmitted should be kept confidential. Thus, a common secret needs to be shared between the user and the access granting server to encipher the exchanged information thereafter.

In key agreement protocols, two or more entities agree on a session key to be used later to assure the confidentiality of the communication between them. The first protocol was proposed in 1976 by W. Diffie and M. Hellman [1]. This protocol does not authenticate the entities, and thus suffers from man-in-the-middle attack. Different approaches have been developed to address this problem [2,3]. The use of authenticated key agreement protocols, which provide implicit authentication, solves the problem of man-in-the middle attack. This implicit authentication is achieved by using a public key infrastructure (PKI). A PKI enables users of a basically insecure public network such as the Internet to securely and privately exchange data and money through the use of a pair of cryptographic keys that is maintained through a trusted certifying authority. One of the two keys is made public and the other key is kept secret. Though, protocols providing implicit authentication are computationally efficient, yet their security properties are usually not strong enough for critical applications requiring high levels of confidentiality.

Bilinear maps were used at first to mount cryptanalysis attacks against cryptographic schemes. Bilinear maps then found positive applications in cryptography [4,5,6,7]. Many traditional PKI, as well as identity-based, key agreement protocols for two and three parties have been proposed employing bilinear pairings. Some examples include Joux's one-round unauthenticated key agreement protocol and the four Tripartite Authenticated Key (TAK) agreement protocols (TAK-1, TAK-2, TAK-3, TAK-4) for sharing a session key among three parties [6, 8].

Tripartite key agreement protocols are of particular importance. They are useful in providing essential security in several vital applications such as in e-commerce where the three entities involved in the protocol are the merchant, the customer and the bank. Other interesting applications include a third party being added to chair or referee a conversation for the purpose of ad hoc auditing, data recovery or escrow purposes.

In this paper, new authenticated key agreement protocols are developed based on the existence of a PKI within which the entities involved in the protocols are registered. Both two-party and three-party cases are considered. The security properties of these protocols are studied.

The rest of the paper is organized as follows. In the next section, the public key infrastructure concept, elliptic curves, bilinear maps, the Weil pairing and the computationally hard related problems are explained. Section III gives details on the desirable security properties for a sound key agreement protocol. Section IV describes our proposed schemes for two and three parties. The performance and security properties of the proposed protocols are examined in Section V. A comparative study is provided in the section that follows. The proposed protocols implementation details are provided in Section VII. Finally, Section VIII concludes the paper.

II. Basic Concepts

In this section, some preliminary concepts necessary to the development of the proposed protocols are introduced.

A. Public Key Infrastructure

The public key infrastructure is based on the existence of a trusted certifying authority (CA), which is the most common method on the Internet for authenticating a message sender or encrypting a message. The basic role of this trusted authority is to provide a certified link between the user's identity and its public key.

Earlier private key cryptography usually involved the

Manuscript received October, 2012; revised December 2012. This work was supported in part by Alexandria University, Egypt.

M. N. Tolba is a teaching assistant in the Department of Engineering Mathematics, Faculty of Engineering, Alexandria University, Egypt, P.O. Box 21544, e-mail: eng.m.nabil.m@gmail.com.

Y. A. Saleh is an assistant professor in the Department of Engineering Mathematics, Alexandria University, Egypt, P.O. Box 21544. Cell phone: +2 0100 37 27 019, e-mail: yasmine.abouelseoud@gmail.com (corresponding author).

G. Elkobrosy is a professor of Engineering Mathematics, Facukty of Engineering, Alexandria University, Egypt (e-mail: elkobrosy@yahoo.com).

A. Abdelrazek is an assistant professor in Engineering Mathematics Department, Faculty of Engineering, University of Alexandria, Egypt (e-mail: amr_abdelrazek_62@yahoo.com).

creation and sharing of a secret key for the encryption and decryption of messages. This secret or private key system has the significant flaw that every pair of users has to share a different key making key management a difficult task over large networks. For this reason, public key cryptography and the public key infrastructure is the preferred approach on the Internet. The private key system is sometimes known as symmetric cryptosystem and the public key system as asymmetric cryptosystem.

A public key infrastructure consists of:

- A certifying authority (CA) that issues and verifies a digital certificate. A certificate includes the public key and information about the identity of the public key owner.
- A registration authority (RA) that acts as the verifier for the certifying authority before it issues a digital signature for the public key of a new user.
- A certificate management system.

B. Elliptic Curves

Recently, elliptic curves have received much attention in the field of cryptography. They are slowly replacing finite fields in the design of new cryptographic schemes. This is due to the fact that the discrete logarithm problem (defined below) over well-chosen elliptic curves is more difficult than the corresponding problem over finite fields. Consequently, smaller key sizes, in the order of 160 bits instead of 256 bits, can be used while achieving the same level of security [9].

An elliptic curve E [10] over a finite field F_p is defined by the Weirestrass equation

$$y^2 = x^3 + ax^2 + bx + c$$

where $D = a^2b^2 - 4a^3c - 4b^3 + 18abc - 27c^2 \neq 0$ and $x \in F_p$ with p a prime greater than 3.

For efficiency purposes, usually a point over an elliptic curve is stored in compressed format. In compressed format, the x-coordinate is only stored along with a single bit indicating whether the positive or negative square root of $x^3 + ax^2 + bx + c$ is the designated y-coordinate.

The set of points on an elliptic curve (E) generated by some point (P) together with the addition operation are known to form an abelian group.

An elliptic curve E over the finite field Z_p^* should be carefully chosen to avoid specialized attacks such as the MOV- attack and the FR- attack [11,12]. Specifications of safe elliptic curves can be found in [13].

C. Bilinear Maps

Bilinear maps and their properties are provided in what follows. More details can be found in Joux [6]. Consider the two groups G_1 (additive) and G_2 (multiplicative) of prime order q, and P a generator for G_1. A symmetric pairing is a computable bilinear map between these two groups.

For our purpose, let $\hat{e}$ be a symmetric bilinear map $\hat{e}: G_1 \times G_1 \longrightarrow G_2$ which satisfies the following three properties.

1- Bilinear:
if $P, Q \in G_1$ and $a, b \in Z_q^*$, then $\hat{e}(aP, bQ) = \hat{e}(abP, Q) = \hat{e}(P, Q)^{ab}$, and $\hat{e}(P, Q + R) = \hat{e}(P, Q).\hat{e}(P, R)$.

2- Non-degenerative: there exist non-trivial points $P, Q \in G_1$ both of order q such that $\hat{e}(P, Q) \neq 1$.

3- Computable: if $P, Q \in G_1$, $\hat{e}(P, Q) \in G_2$ is efficiently computable in polynomial time.

D. The Weil Pairing

Let G_1 be a subgroup of the group of points on the elliptic curve E over the finite field F_q. Let the order of G_1 be denoted by ℓ where q and ℓ are relatively prime. Let G_2 be a finite field extension of F_q. The Weil pairing [4, 14] is a well-known map $\hat{e}: G_1 \times G_1 \longrightarrow G_2$ which satisfies the properties given above.

E. Hard Computational Problems

Many pairing-based cryptographic protocols are based on the hardness of the BDHP (Bilinear Diffie-Hellman Problem) for their security [4,15]. Some computational problems related to the elliptic curve cryptography are defined below.

- **Bilinear Diffie-Hellman Problem (BDHP)**

Given $(P, xP, yP, zP) \in G_1$ for some x, y, z chosen at random from Z_q^*, compute $\hat{e}(P, P)^{xyz} \in G_2$.

- **Discrete Logarithm Problem (DLP)**

Given $P, Q \in G_1$, find an integer n such that $P = nQ$.

- **Computational Diffie-Hellman Problem (CDHP)**

Given a tuple $(P, aP, bP) \in G_1$ for $a, b \in Z_q^*$, find the element abP.

III. Desirable Security Properties of a Key Agreement Protocol

In order to develop a sound key agreement protocol, the desirable security properties it must satisfy should be carefully understood. These properties are described in detail in [16]. Here, assume A and B are two honest entities. It is desired for an authenticated key agreement protocol to possess the following properties [15, 16, 17, 18]:

A. Known-Key Security

Each key generated in one protocol round is independent and should not be exposed if other secret keys are compromised.

B. 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.

C. Key-Compromise Impersonation

Assume that A and B are two entities. Suppose A's secret key is disclosed. Obviously, an adversary who knows this secret key can impersonate *A* to *B*. However, it is desired that this disclosure does not allow the adversary to impersonate *B* to the real *A*. In the two-party case, only an outsider would impersonate the communicating parties. However, in the *n*-party case, for $n \geq 3$, one party of the communicating group might impersonate another party to the rest of the parties of the group. This kind of impersonation attack is called the insider impersonation attack.

D. Key Control

The key should be determined jointly by both *A* and *B*. Neither *A* nor *B* can control the key alone.

IV. The Proposed Protocols

In this section, new schemes for authenticated key agreement are developed, which are extensions of the schemes in [19] to the traditional PKI-based cryptosystems. These schemes consist of two phases, which are the setup phase and the session key generation phase. The setup phase is common to all schemes and it is described here.

Setup:

The system set up algorithm generates the following parameters for the users. The public domain parameters are $(p, q, E, P, \hat{e}, H)$, where E is an elliptic curve defined over Z_p , P is a generator for a group of points on E (the group G_1) with order q. The hash function H is a one-way hash function that maps from G_1 into G_1 and $\hat{e}$ is a bilinear map.

Each entity obtains a certificate for its static public key. Let $Cert_A$ denote *A*'s public-key certificate, which includes her static public key $Q_A = aP$ and a certification authority (CA) signature over this information, where *a* is the long-term private key of the entity *A*.

A. Protocol 1

Suppose there are two entities *A* and *B* who want to agree on a session key. They exchange their public key certificates and the CA signature is verified.

Key generation:

A and *B* select *x, y* randomly and independently, then they compute and broadcast the following:

1. $A \longrightarrow B$: $P_A = xP$, $T_A = H(P_A)a + xP_A$
2. $B \longrightarrow A$: $P_B = yP$, $T_B = H(P_B)b + yP_B$

A verifies $\hat{e}(T_B, P) =?\ \hat{e}(H(P_B), Q_B).\hat{e}(P_B, P_B)$

B also verifies $\hat{e}(T_A, P) =? \hat{e}(H(P_A), Q_A).\hat{e}(P_A, P_A)$

If the above equations hold, then *A* and *B* compute:

$$K_A = \hat{e}(P_B, Q_B)^{ax}, K_B = \hat{e}(P_A, Q_A)^{by}$$

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

The correctness of the protocol can be easily verified as follows based on the properties of the bilinear map. The verification equation that *A* uses is only investigated and clearly similar arguments hold for *B*.

$$\begin{aligned}\hat{e}(H(P_B), Q_B).\hat{e}(P_B, P_B) &= \hat{e}(H(P_B), bP).\hat{e}(yP, yP)\\ &= \hat{e}(H(P_B)b, P).\hat{e}(y^2P, P)\\ &= \hat{e}(H(P_B)b+y^2P, P)\\ &= \hat{e}(T_B, P)\end{aligned}$$

B. Protocol 2

This protocol extends the above protocol to the case where two entities *A* and *B* need to agree on a set of ***four*** session keys. The public key certificates are exchanged and the associated CA signatures are verified.

Key generation:

A and *B* select the pairs (*x*, *x'*) and (*y*, *y'*) randomly and independently, and then compute and broadcast the following:

1. $A \longrightarrow B$: $P_A = xP$, $P'_A = x'P$, $T_A = H(P_A, P'_A)a + xP'_A$
2. $B \longrightarrow A$: $P_B = yP$, $P'_B = y'P, T_B = H(P_B, P'_B)b + yP'_B$

Upon receiving the broadcasted points, each entity proceeds to verify the authenticity of the received data.

A verifies $\hat{e}(T_B, P) =? \hat{e}(H(P_B, P'_B), Q_B).\hat{e}(P'_B, P_B)$

B verifies $\hat{e}(T_A, P) =? \hat{e}(H(P_A, P'_A), Q_A).\hat{e}(P'_A, P_A)$

If the above equations hold, then *A* and *B* compute the first key as:

$$K_{A(1)} = \hat{e}(P_B, Q_B)^{ax},\ \ K_{B(1)} = \hat{e}(P_A, Q_A)^{by}$$

Then, the first session key is

$$K_{A(1)} = K_{B(1)} = \hat{e}(P, P)^{axby}$$

The remaining three session keys as will be computed by *A* are given below.

$$K_{A(2)} = \hat{e}(P_B, Q_B)^{ax'}\ ,\quad K_{A(3)} = \hat{e}(P'_B, Q_B)^{ax}$$
$$K_{A(4)} = \hat{e}(P'_B, Q_B)^{ax'}$$

Again, the consistency check of the verification equation for one of the entities (*A*) is provided below based on the properties of the bilinear map.

$$\begin{aligned}\hat{e}(H(P_B, P'_B), Q_B).\hat{e}(P'_B, P_B) &= \hat{e}(H(P_B, P'_B), bP).\hat{e}(y'P, yP)\\ &= \hat{e}(H(P_B, P'_B)b + yy'P, P)\\ &= \hat{e}(H(P_B, P'_B)b + yP'_B, P) = \hat{e}(T_B, P)\end{aligned}$$

C. Protocol 3

Suppose there are three entities *A*, *B* and *C* who want to agree on a session key. They exchange their public key certificates and the CA signature is verified.

Key generation:

A, *B* and *C* select *x, y, z* randomly and independently, then they compute and broadcast the following:

1. $A \longrightarrow B, C$: $P_A = xP$, $T_A = H(P_A)a + xP_A$
2. $B \longrightarrow A, C$: $P_B = yP$, $T_B = H(P_B)b + yP_B$
3. $C \longrightarrow A, B$: $P_C = zP$, $T_C = H(P_C)c + zP_C$

A verifies $\hat{e}(T_B + T_C, P) =?$
$\hat{e}(H(P_B), Q_B).\hat{e}(H(P_C), Q_C).\hat{e}(P_B, P_B).\ \hat{e}(P_C, P_C)$

B verifies $\hat{e}(T_A + T_C, P) =?$
$\hat{e}(H(P_A), Q_A).\hat{e}(H(P_C), Q_C).\hat{e}(P_A, P_A).\ \hat{e}(P_C, P_C)$

C verifies $\hat{e}(T_A + T_B, P) =?$
$\hat{e}(H(P_A), Q_A).\hat{e}(H(P_B), Q_B).\hat{e}(P_A, P_A).\ \hat{e}(P_B, P_B)$

If the above equations hold, then *A*, *B* and *C* compute:

$$K_A = \hat{e}(P_B, P_C)^x, K_B = \hat{e}(P_A, P_C)^y, K_C = \hat{e}(P_A, P_B)^z$$

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

The correctness of the protocol can be easily verified as follows based on the properties of the bilinear map. The verification equation that A uses is only investigated and clearly similar arguments hold for B and C.

$$\begin{aligned}&\hat{e}(H(P_B),Q_B).\hat{e}(H(P_C),Q_C).\hat{e}(P_B,P_B).\ \hat{e}(P_C,P_C)\\&= \hat{e}(H(P_B),bP).\hat{e}(H(P_C),cP).\hat{e}(yP,yP).\ \hat{e}(zP,zP)\\&= \hat{e}(H(P_B)b,P).\hat{e}(H(P_C)c,P).\hat{e}(y^2P,P).\ \hat{e}(z^2P,P)\\&=\hat{e}(H(P_B)b+yP_B,P).\hat{e}(H(P_C)c+zP_C,P)\\&=\hat{e}(T_B+T_C,P)\end{aligned}$$

D. Protocol 4

Again, the above protocol is extended to the case where there are three entities A, B and C who want to agree on a set of ***eight*** session keys. The public key certificates as usual are exchanged and the associated CA signatures are verified.

Key generation:

A, B and C select the pairs (x, x'), (y, y') and (z, z') randomly and independently, and then compute and broadcast the following:

1. $A \longrightarrow B,C$: $P_A = xP, P'_A = x'P$, and $T_A = H(P_A,P'_A)a + xP'_A$
2. $B \longrightarrow A,C$: $P_B = yP, P'_B = y'P$, and $T_B = H(P_B,P'_B)b + yP'_B$
3. $C \longrightarrow A,B$: $P_C = zP, P'_C = z'P$, and $T_C = H(P_C,P'_C)c + zP'_C$

Upon receiving the broadcasted points, each entity proceeds to verify the authenticity of the received data.

A verifies

$$\hat{e}(T_B+T_C,P) =?\ \hat{e}(H(P_B,P'_B),Q_B).\hat{e}(H(P_C,P'_C),Q_C).\hat{e}(P_B,P'_B)\cdot\hat{e}(P_C,P'_C)$$

B verifies

$$\hat{e}(T_A+T_C,P) =?\ \hat{e}(H(P_A,P'_A),Q_A).\hat{e}(H(P_C,P'_C),Q_C).\hat{e}(P_A,P'_A)\cdot\hat{e}(P_C,P'_C)$$

C verifies

$$\hat{e}(T_A+T_B,P) =?\ \hat{e}(H(P_A,P'_A),Q_A).\hat{e}(H(P_B,P'_B),Q_B).\hat{e}(P_A,P'_A)\cdot\hat{e}(P_B,P'_B)$$

If the above equations hold, then A, B and C compute the first key as:

$$K_{A(1)} = \hat{e}(P_B,P_C)^x,\ K_{B(1)} = \hat{e}(P_A,P_C)^y,\ K_{C(1)} = \hat{e}(P_A,P_B)^z$$

Then, the first session key is

$$K_{A(1)} = K_{B(1)} = K_{C(1)} = \hat{e}(P,P)^{xyz}$$

The remaining seven session keys as will be computed by A are given below.

$$\begin{aligned}&K_{A(2)} = \hat{e}(P_B,P'_C)^x\ ,K_{A(3)} = \hat{e}(P'_B,P_C)^x\\&K_{A(4)} = \hat{e}(P'_B,P'_C)^x\ ,K_{A(5)} = \hat{e}(P_B,P_C)^{x'}\\&K_{A(6)} = \hat{e}(P_B,P'_C)^{x'},\ K_{A(7)} = \hat{e}(P'_B,P_C)^{x'}\\&K_{A(8)} = \hat{e}(P'_B,P'_C)^{x'}\end{aligned}$$

Again, the consistency check of the verification equation for one of the entities (A) is provided below.

$$\begin{aligned}&\hat{e}(H(P_B,P'_B),Q_B).\hat{e}(H(P_C,P'_C),Q_C).\hat{e}(P_B,P'_B)\cdot\hat{e}(P_C,P'_C)\\&= \hat{e}(H(P_B,P'_B),bP).\hat{e}(H(P_C,P'_C),cP)\cdot\hat{e}(yP,y'P)\\&\quad\cdot\hat{e}(zP,z'P)\\&= \hat{e}(H(P_B,P'_B)b,P).\hat{e}(H(P_C,P'_C)c,P)\cdot\hat{e}(y\,y'P,P)\\&\quad\cdot\hat{e}(z\,z'P,P) =\hat{e}(T_B,P)\cdot\hat{e}(T_C,P)=\hat{e}(T_B+T_C,P)\end{aligned}$$

V. Performance Analysis And Security Analysis for the Proposed Protocols

In this section, the performance of the proposed schemes is investigated. In addition, the increase in computations involved in the schemes is justified due to the high security guarantees offered by these schemes and the possibility of off-loading some of the computational burden to a trusted third party such as a firewall.

A. Computational Burden

First, the two-party schemes are studied. Three (Four) scalar point multiplications and one (four) pairing evaluation are needed for the generation of the session key for protocol 1 (2). In addition, three pairing evaluations are required in the authentication phase for protocol 1 and 2; that is, verifying the identities of the parties involved in the protocol. However, it is clear from the verification equation that neither long-term nor short-term keys are required in this phase and thus the verification step can be done by a firewall reducing the computational load significantly.

As for the proposed three-party schemes, three (four) scalar point multiplications and one (eight) pairing evaluations are required for the generation of session keys for protocol 3 (protocol 4). In the authentication step, five pairing evaluations are needed for protocol 3 and 4. However, in protocol 4, since eight session keys are generated in one step, it can be envisioned that the computational load per key is just about one pairing evaluation and one-half of a scalar point multiplication. Again, the verification equations in this phase involve no private keys and hence the computational load can be easily moved to a more powerful server such as a firewall.

B. Security Properties

The two and three party schemes security properties are examined in what follows.

Security Properties of Protocols 1 and 2

Known key security: In each run of these protocols, a new session key is computed which depends on short-term private keys x and y ((x, x') and (y, y')) selected randomly in each session. Thus, the knowledge of a past session key will not allow an adversary to deduce the future keys.

Partial forward secrecy: if the adversary knows the long-term private key of one entity, he will not be able to compute a previous session key. Assume, for example, that A's private key is compromised. It is clear that computing $\hat{e}(P_B,Q_B)^{ax}$ is infeasible without the knowledge of the short-term private key that is chosen randomly every session. However, if he knows long-term private keys of all entities, he will able to compute a previous session key by the relation $\hat{e}(P_A,P_B)^{ab}$. In practical scenarios, there is usually a highly secure end involved in the communication (a remote server), whose key compromise is rather difficult and thus the proposed protocol can still provide a desirable level of security.

Key control: All the entities contribute an equal share to the computation of the key. No one can force the session key to take on a specific pre-computed value.

Key-compromise impersonation: Suppose an adversary E knows the private key of A. He will not able to impersonate B to A unless he knows the private key of B, because of the fact that A authenticates B before computing the session key. No one can impersonate B unless he knows his private key; this is clear from the calculation of T_B.

Security Properties of Protocols 3 and 4

Known key security: In each run of Protocol 3 (4), keys are computed depending on short-term private key pairs x, y and z ((x, x'), (y, y') and (z, z')) which are selected randomly in each session.

Prefect forward secrecy: Even if the adversary knows the long-term private keys of all entities, he will not able to compute a previous session key. Assume, for example, that A's private key is compromised. It is clear that computing $\hat{e}(P_B, P_c)^x$ is infeasible without the knowledge of the short-term private key that is chosen randomly every session.

Key control: All the entities contribute an equal share to the computation of the key. No one can force the session key to take on a specific pre-computed value.

Key-compromise impersonation: Suppose an adversary E knows the private key of A. He will not able to impersonate B to A unless he knows private key of B, because A - before computing the session key- authenticates both B and C. No one can impersonate B or C unless he knows their private keys; as is apparent from the calculations of T_B and T_C. Moreover, this protocol provides explicit authentication and not just implicit authentication, which makes this protocol resistant to insider impersonation attack (suppose A, B and C are the communicating entities, insider impersonation means that one of them, suppose C, impersonates other entities like B to A. Thus, C will talk with A once as he is C and another time as if he is B). Explicit authentication avoids this attack, in addition to its resistance to the outsider impersonation attack.

VI. COMPARATIVE STUDY

In this section, we compare our protocols with other protocols with regard to security and performance. From the security point of view, the criterion to compare the security of the protocols is given by the extent to which a specific protocol fulfills the security properties as discussed in Section III. From the performance point of view, the criterion for comparing the efficiency is expressed in terms of the number of arithmetic operations required per generated key.

A. Security Comparison

The security comparison of the protocols involves three criteria: the fulfillment of security properties as defined in Section III, and the existence of insider impersonation attack, and type of authentication (implicit, explicit).

Table I compares the fulfillment of security properties of some 2-party protocols in literature and our protocols. The following abbreviations and notations are used in Table I and Table II:

KKS: Known-Key Secrecy, FS: Forward Secrecy,
KCI: Key-Compromise Impersonation,
KC: Key Control,
IKA: Implicit Key Authentication,
EKA: Explicit Key Authentication,
II: Insider Impersonation,
+: means protocol satisfies the property,
- : means protocol does not satisfy the property,
*: perfect forward secrecy

TABLE I. SECURITY PROPERTIES FOR 2-PARTY PROTOCOLS

Protocol	KKS	FS	KCI	KC	IKA	EKA
ADHP$_1$ [16]	+	+*	-	+	+	-
ADHP$_2$ [16]	+	+	+	+	+	-
MTI/A0 [21]	+	+	+	+	+	-
Two-Pass Unified Model [21]	+	+*	-	+	+	-
Protocols 1 and 2	+	+	+	+	+	+

Table II provides a comparison for the fulfillment of security properties for some 3-party protocols in literature and our protocols.

TABLE II. SECURITY PROPERTIES FOR 3-PARTY PROTOCOLS

Protocol	KKS	FS	KCI	KC	IKA	EKA	II
TAK-1[8]	-	+*	-	+	+	-	-
TAK-2[8]	-	+	-	+	-	-	-
TAK-3[8]	+	-	-	+	+	-	-
TAK-4[8]	+	+*	-	+	+	-	-
Shim's Protocol[22]	+	+*	-	+	+	-	-
Protocols 3 and 4	+	+*	+	+	+	+	+

It is clear from the above tables that the proposed protocols satisfy various security requirements of a key agreement protocol.

B. Efficiency Comparison

The computational load per user per key (number of computations performed) for the reviewed protocols as well as the proposed ones is given in Table III and Table IV.

We consider operations which are expensive from the computational point of view - pairing operations, scalar point multiplications and exponentiations. The following abbreviations are used.

PairOpA: pairing operations in Authentication,
PairOpG: pairing operations in Generation,
ScMul: scalar point multiplications in G_1,
MULG2: scalar multiplications in G_2,
EXPMP: exponentiation modulo P,
MULMP: multiplication modulo P

TABLE III. COMPUTATIONAL LOAD PER USER OF 2-PARTY PROTOCOLS

Protocol	PairOpA	PairOpG	ScMul	EXPMP	MULMP
ADHP$_1$				3	
ADHP$_2$				3	
MTI/A0				3	1
Two-Pass Unified Model				3	
Protocol 1	3	1	3	1	
Protocol 2	3/4	4/4	4/4	4/4	

It is clear that, for frequently communicating parties with sufficient secure storage media, it is more efficient to use Protocol 2 rather than Protocol 1. Similar arguments hold for protocols 3 and 4.

TABLE IV. COMPUTATIONAL EFFORT PER USER OF 3-PARTY PROTOCOLS

Protocol	PairOpA	PairOpG	ScMul	EXPMP	MULG2
TAK-1		2	1	2	1
TAK-2		3	1	3	2
TAK-3		3	1	3	2
TAK-4		1	1	1	2
Shim's Protocol		2	1	2	
Protocol 3	5	1	3	1	
Protocol 4	5/8	8/8	4/8	8/8	

VII. IMPLEMENTATION

The proposed four protocols have been implemented using the C++ PBC Library under Ubuntu operating system on a Pentium(R) Dual Core PC. Type A elliptic curves have been used in our sample runs for testing the validity and ensuring the timeliness of the proposed protocols.

Type A pairings are symmetric pairings constructed on the elliptic curve $y^2 = x^3 + x$ over the field F_q for some prime q = 3 mod 4. G_1 is the group of points $E(F_q)$. It turns out that $\#E(F_q) = q + 1$ and $\#E(F_q^2) = (q + 1)^2$. Thus, the embedding degree k is 2, and hence G_2 is a subgroup of F_q^2. The order r is some prime factor of (q + 1). Write q + 1 = r * h. For efficiency, r is picked to be a Solinas prime, that is, r has the form of $2^a \pm 2^b \pm 1$ for some integers 0 < b < a. Moreover, q = - 1 mod 12 in order that F_q^2 can be implemented as $F_q[i]$ (where i = sqrt(-1)). The values used in one of the sample runs were:

q
6748275749396084910780880425190587742657653654723396365613146028221304479278136879384643445483363971199436778850236944766802842904329974680684969486323800985884225263988186901190289777518592545214467032660799233622336396538016986710325909583260317868359292408436891364703128957677891007814533963825387100012

q
674827574939608491078088042519058774265765365472
339636561314602822130447927813687938464344548336
397119943677885023694476680284290432997468068496
948632380098588422526398818690119028977751859254
521446703266079923362233639653801698671032590958
326031786835929240843689136470312895767789100781
4533963825387100012

h
923471472737000152987412339637424041780981361280
746223306168381085611698240770485974526866113232
485174709743945038228058188028652534108231651231
645806953171793301047418865563926133737701045576
718511862641424137218497194561883029964110489148
2528426788797817913

r
730750862221594424981965739670091261094297337857

a 159

b 135

VIII. CONCLUSION

In this paper, four new authenticated key agreement protocols offering high level security guarantees have been proposed. The main advantage of the proposed schemes is that they provide explicit authentication. This makes it possible for the authenticity of the identities of the communicating parties to be done by means of a firewall relieving the users involved from much of the computational burden associated with the authentication step. Moreover, in the tripartite case, explicit authentication prevents insider impersonation attacks.

The first two schemes are two-party schemes, while the remaining two are tripartite schemes. All schemes resist various known attacks suggesting their use for highly confidential communications. Moreover, implementation of the schemes revealed that the protocols can be used in real-time applications. For devices with limited computational capabilities, the verification of user identities can be moved to a trusted third party such as a firewall and dedicated hardware can be used for pairings evaluation.

REFERENCES

[1] W. Diffie, M. Hellman, New directions in cryptography, IEEE Transactions on Information Theory, IT-22(6),November 1976, pp.644-654

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

[3] R. Dutta, R. Barua, Overview of Key Agreement Protocols, Cryptology ePrint Archive, Report 2005/289, 2005.

[4] D. Boneh, and M. Franklin, Identity-based encryption from the Weil pairing, *Advances in Cryptology- Crypto'01, LNCS Vol. 2442*, Springer, UK, pp. 354-368, 2001.

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

[6] A. Joux, A one round protocol for tripartiteDiffie-Hellman, 4th International Symposium on AlgorithmicNumber Theory, LNCS Vol. 1838, Springer,UK, pp. 385-393, 2000.

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

[8] Sattam S. Al-Riyami, Cryptographic Schemes based on Elliptic Curve Pairings, Information Security Group, Department of Mathematics , Royal Holloway, University of London, 2004.

[9] K. Giuliani, Attacks on the Elliptic Curve Discrete Logarithm Problem, Master of Mathematics, University of Waterloo, Ontario, Canada, 1999.

[10] J. H. Silverman, The Arithmetic of Elliptic Curves, GTM 106, Springer-Verlag, 1986.

[11] A. Menezes, T. Okamoto and S. Vanstone, Reducing Elliptic CurveLogarithm to Logarithms in a Finite Field, IEEE Transactions on Information Theory, vol. 39, pp. 1639-1646, 1993.

[12] G. Frey and H. Ruck, A Remark Concerning *m*-divisibility and theDiscrete Logarithm Problem in the Divisor Class Group of Curves, Mathematics of Computation, vol. 62, pp. 865-874, 1994.

[13] Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curves Domain Parameters, Certicom Research, Version 1.0, September 2000.

[14] V. Miller, The Weil Pairing and Its Efficient Calculation, Journal of Cryptology, vol. 17(4), pp. 235-262, 2004.

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

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

[17] Bellare, M., Rogaway, P. Entity Authenticationand Key Distribution, Advances in Cryptology -*CRYPTO '93*, Springer, UK, pp. 232-249, 1993.

[18] D. Nalla, K.C. Reddy, ID-based tripartitekey agreement with signatures, *Cryptology ePrint* Archive, Report 2003/004.

[19] Marko Hölbl, Tatjana Welzer and Boštjan Brumen, Comparative Study of Tripartite Identity-Based Authenticated Key Agreement Protocols, Informatica, vol.33 (2009), pp. 347–355.

[20] Benn Lynn, PBC Library Manual 0.5.11, 2006.

[21] B. Song and K. Kim, Two-Pass Authenticated Key Agreement Protocol with Key Confirmation, Progress in Cryptology – Indocrypt'2000, LNCS 1977, Springer-Verlag, pp.237-249, December 2000.

[22] S. Sun and B. Hsieh, Security analysis of Shim's authenticated key agreement protocols from Pairings. Cryptology ePrint Archive, Report 2003/113 (2003).