A Certificateless Multiple-key Agreement Protocol without Hash Functions Based on Bilinear Pairings

Mohammad Sabzinejad, Mahmoud Ahmadian Attari, and Majid Bayat

Abstract—Certificateless cryptosystems were proposed by Al-Riyami and Paterson in 2003 [1] to solve problems of public key cryptosystems based on PKI and based on identity. Up to now, various types of certificateless cryptographic primitives as encryption functions, signature schemes, key agreement protocols and etc, have been designed. But to the best of our knowledge, multiple-key agreement protocols have not been proposed based on certificateless cryptosystem yet. So in this paper we propose a certificateless authenticated multiple-key agreement protocol with bilinear pairings.

Index Terms—Certificateless, cryptosystem, multiple-key agreement protocol, signature schemes, bilinear pairing.

I. INTRODUCTION

Public key cryptography is a main area in the cryptography. In this area each user has a private key and the corresponding public key. The main problem in these cryptosystems is how establishing a link between user's identity (ID) and her/his public key. A general solution for this problem is based on Public Key Infrastructure (PKI), defined in ISO/IEC 9594-8 [14], in this solution a trust authority, called Certificate Authority (CA), issues a certificate contained user's ID and user's public key signed with the private key of CA.

Because of issuing and using the certificate are costly, other solutions as Identity Based Cryptography (IBC) have been proposed. The IBC idea first was proposed by Shamir in 1984 [18]. In an IBC system user's ID is considered as her/his public key and the user's private key is generated by a trust authority, called Key Generation Center (KGC) or Private Key Generation (PKG). The main advantage of the IBC systems is that unlike PKI systems, issuing a certificate for each user isn't needed because there is an inherent link between user's ID and her/his public key. Nevertheless, the disadvantage of IBC systems is that the KGC knows the user's private key and subsequently he may impersonate users or a user may deny what he had done with his own private key (e.g. signing a message). This property is called key escrow. For obtaining more information about IBC systems the reader may refer to [3, 8, 9, 11].

To solve the key escrow problem in IBC, researchers have proposed two types of solutions. In the one solution, the user's private key is split to two parts such that the KGC is just allowed to escrow the one part, for instance [1, 2, 12].

In the other solution, the KGC is split to multiple KGCs such that each of them is allowed to escrow a part of the user's private key and multiple contributions of KGCs are used to create the user's private key, for example [6, 7, 10].

One of the solutions that split user's private key is called certificateless public key cryptosystem that proposed by Al-Riyami and Paterson [1] in 2003. In this cryptosystem, the private key consists of two parts which one of them is made by the user and the other is made by KGC. Up to now, various types of certificateless cryptographic primitives as encryption functions [20], signature schemes [21], key agreement protocols [19] and so on, have been designed.

In this paper, we proposed a certificateless authenticated multiple-key agreement protocol with bilinear pairings. Up to now many certificateless key agreement protocols and many multiple-key agreement protocols have independently been proposed but to the best of our knowledge, the proposed protocol is the first protocol that combines multiple-key agreement idea with certificateless cryptosystem. Multiple-key agreement protocols that conformed to the idea of MQV [17] protocol, not using hash function, was introduced by Harn and Lin [13] in 1998.

A. Security Properties of Key Agreement Protocols

The most important security properties of key agreement protocols [5] are indicated in the following. Let that A and B are two participants who are intended to agree on a secret key after executing a key agreement protocol.

- **Known-Key Security**: This property says that the adversary who has obtained some previous session keys cannot compute the next session keys.
- **Forward Secrecy**: This property implies that revealed one or more long-term private keys of two participants do not cause the previous session keys be obtained for adversary. If this property only remains for one of the long-term private keys, this property is called partial forward secrecy. Perfect forward secrecy emphasizes that if both private keys of the participants are disclosed, the adversary is unable to compute the previous session keys.
- **Key-Compromise Impersonation**: This property expresses that if the long-term private key of one entity (e.g. A) is disclosed, the adversary is unable to impersonate the other entity to the compromised entity (e.g. B to A).
- **Unknown key security**: This property implies that the active adversary C should not enable to interfere in a key agreement protocol run such that A believes that B is her participant while B believes that he shared the session key with C.

In addition, two essential properties are regarded for key agreement protocols as follows:

A A Ax D X protocols. So designer of key agreement protocols are done by each participant and communication cost is dependent to the amount of calculation. The efficiency is evaluated by computation and communication cost in a key agreement protocol. The proposed scheme also is based on certificateless public key cryptosystem, and has three phases that are coming in the following.

B. Bilinear Pairings

Let \( G_1 \) be a cyclic additive group generated by \( P \), whose order is a prime \( n \) and \( G_2 \) is a cyclic multiplicative group of the same order \( n \). Let the discrete logarithm problem (DLP) in both \( G_1 \) and \( G_2 \) is hard. An bilinear pairing is a map \( e: G_1 \times G_1 \rightarrow G_2 \), which satisfies the following three properties:

- **Bilinear:**
  \[ e(P + P', Q) = e(P, Q)e(P', Q), \]
  \[ e(P.Q_1 + Q_2) = e(P, Q_1)e(P, Q_2), \]
  \[ e(aP, bQ) = e(P, Q)^{ab} \text{ where } a, b \in \mathbb{Z}_q^*, \]

- **Non-Degenerate:** There exists \( P \in G_1 \) and \( Q \in G_2 \) such that \( e(P, Q) \neq 1 \).

- **Computable:** There is an efficient algorithm to compute \( e(P, Q) \) for all \( P, Q \in G_1 \).

Security of pairing based cryptosystems depends on the intractability of solving Bilinear Diffie-Helman Problem that introduced by Boneh and Franklin [6] as follows:

- **Bilinear Diffie-Helman problem (BDHP):** For bilinear pairing \( e: G_1 \times G_1 \rightarrow G_2 \), given \( p, aP, bP, cP \), compute \( e(P, P)^{abc} \), where \( a, b, c \) are randomly chosen from \( \mathbb{Z}_q^* \).

The BDH problem is no harder than either the elliptic curve Diffie-Helman problem (ECDHP) or the finite field Diffie-Helman problem (DHP) (Lemma IX.23. of [4]).

II. THE PROPOSED SCHEME

In this section we will propose a novel multiple key agreement protocol that, same to Harn-Lin scheme, doesn't use of hash function for construction of digital signature. The proposed scheme also is based on certificateless public key cryptosystem, and has three phases that are coming in the following.

A. Setup Phase

In this phase, Key Generation Center (KGC) chooses below items:

- Elliptic curve \( E \) over finite field \( F_q \), where \( q \) is a power of a prime number,
- Subgroup \( G_1 \) with prime order \( n \) and generator \( P \), of group \( E(F_q) \),
- Multiplicative group \( G_2 \) with prime order \( n \),
- Bilinear pairing, \( e: G_1 \times G_1 \rightarrow G_2 \),
- Map-to-point, \( H: 0^1 \rightarrow G_1 \).

Then, KGC chooses a random number \( x_{KGC} \in \mathbb{Z}_n^* \) as master-key and computes the public key \( P_{KGC} = x_{KGC}P \). In the end of the setup phase, KGC publishes the system parameters \( \{G_1, G_2, n, P, e, H, P_{KGC}\} \).

B. Key Extraction Phase

In this phase each user \( U_i \) with identity \( ID_i \) obtains a long-term private/public key pair as follows:

- \( U_i \) sends his/her identity \( ID_i \) to KGC and request a partial-long term private key.
- KGC after verifying the user's identity, computes \( Q_i = H(ID_i) \) and the partial user's long-term private key \( D_i = x_{KGC}Q_i \) then sends \( D_i \) to \( U_i \) via a secure channel.
- \( U_i \) after receiving \( D_i \), checks if \( e(D_i, P) = e(H(ID_i), P_{KGC}) \) , chooses a random number \( x_i \in \mathbb{Z}_n^* \) securely and computes its corresponding public value \( P_i = x_iP \).
- Finally the user saves \( X_i = \{D_i, x_i\} \) as his/her long-term private key and \( Y_i = \{ID_i, P_i\} \) as his/her long-term public key.

By the way, user \( A \) obtains the long-term private key \( X_A = \{D_A, x_A\} \) and the long-term public key \( Y_A = \{ID_A, P_A\} \) and user \( B \) also, obtains \( X_B = \{D_B, x_B\} \) and \( Y_B = \{ID_B, P_B\} \) as his long-term private and public key respectively.

C. Key Agreement Phase

In this phase, entities \( A \) and \( B \) that have private/public key pairs, \( \{X_A = \{D_A, x_A\}\}, Y_A = \{ID_A, P_A\} \) and \( \{X_B = \{D_B, x_B\}\}, Y_B = \{ID_B, P_B\} \) respectively, execute protocol 1. Description of the protocol is as follows:

- **Entity A**, as initiator of the protocol, chooses two random numbers, \( r_{A1}, r_{A2} \in \mathbb{Z}_n^* \), and computes \( T_{A1} = r_{A1}P \) and \( T_{A2} = r_{A2}P \) such that \( k_{A1} \cdot k_{A2} \neq 0 \text{ mod } n \), where \( k_{A1} \) and \( k_{A2} \) are \( x \)-coordinates of points \( T_{A1} \) and \( T_{A2} \) respectively. Then A signs points \( T_{A1} \) and \( T_{A2} \) as follows:
  \[ S_A = (k_{A1} \cdot k_{A2})x_Ak_AD_A + D_A + (k_{A1}r_{A1} + k_{A2}r_{A2})Q_A \]
  Where \( k_A \) is \( x \)-coordinate of public key \( P_A \). At the end of the step, A sends quantities \( T_{A1}, T_{A2}, S_A, Y_A = \{ID_A, P_A\} \) to B.
- **Entity B**, also, same to A, chooses \( r_{B1}, r_{B2} \in \mathbb{Z}_n^* \) and computes \( T_{B1} = r_{B1}P \) and \( T_{B2} = r_{B2}P \) where, \( k_{B1} \cdot k_{B2} \neq 0 \text{ mod } n \). Then he computes
  \[ S_B = (k_{B1} \cdot k_{B2})x_Bk_BQ_B + D_B + (k_{B1}r_{B1} + k_{B2}r_{B2})Q_B \]
Where \( k_B \) is x-coordinate of public key \( P_B \). At the end of the step, B sends quantities \((T_{B1}, T_{B2}, S_B, Y_B = \langle ID_B, P_B \rangle)\) to A.

- A, upon receiving the messages from B, checks that \( k_{B1}, k_{B2} \neq 0 \mod n \), if it holds, he computes \( Q_B = H(ID_B) \) and verifies signature \( S_B \) as follows:

\[
e(P, S_B) = e(k_{a1}T_{A1} + k_{a2}T_{A2} + (k_{a1} \cdot k_{a2}) \langle k_B P_B + P_{GCC} \rangle, Q_B)
\]

If it does not hold, then A terminates the execution.

Otherwise, A computes session keys as follows:

\[
K_1 = r_{a1}T_{B1} = r_{a1}r_{B1}P \\
K_2 = r_{a1}T_{B2} = r_{a1}r_{B2}P \\
K_3 = r_{a2}T_{B1} = r_{a2}r_{B1}P \\
K_4 = r_{a2}T_{B2} = r_{a2}r_{B2}P
\]

- Entity B also, upon receiving messages from A, checks that \( k_{B1}, k_{B2} \neq 0 \mod n \), if it holds, he computes \( Q_A = H(ID_A) \) and verifies signature \( S_A \) as follows:

\[
e(P, S_A) = e(k_{a1}T_{A1} + k_{a2}T_{A2} + (k_{a1} \cdot k_{a2}) \langle k_B P_B + P_{GCC} \rangle, Q_A)
\]

If it does not hold, then B terminates the execution. Otherwise, B computes session keys as follows:

\[
K_1 = r_{B1}T_{A1} = r_{B1}r_{A1}P \\
K_2 = r_{B2}T_{A1} = r_{B2}r_{A1}P \\
K_3 = r_{B1}T_{A2} = r_{B1}r_{A2}P \\
K_4 = r_{B2}T_{A2} = r_{B2}r_{A2}P
\]

The soundness of the signature is shown in the below equation:

\[
e(P, S_A) = e(k_{a1}T_{A1} + k_{a2}T_{A2} + (k_{a1} \cdot k_{a2}) \langle k_B P_B + P_{GCC} \rangle, Q_A)
\]

III. SECURITY ANALYSES OF THE PROPOSED SCHEME

- Known-Key Security: This property says that the adversary who has obtained one or more session keys is unable to compute the next session keys. In the proposed key agreement protocol, suppose that the adversary knows the session keys of a session, \( K_{ij} = r_{ai}T_{Bj} = r_{ai}r_{Bj}P \) for \( i, j = 1, 2 \). It does not give to the adversary any useful information to compute the next session keys. Because for computing the session keys short-term private keys \( r_{ai} \) and \( r_{Bj} \) that be changed in each session are used. So the proposed protocol is secure against Known-Key attack.

- Unknown Key Security: In the section 1 we defined this attack. For executing this attack on the proposed protocol, the adversary C intercepts the sent message \((T_{A1}, T_{A2}, S_A, Y_A = \langle ID_A, P_A \rangle)\) from A. Then he...
must sign the values \((T_{A1}, T_{A2})\) by using his private key as follow:

\[
S_C = (k_{A1} + k_{A2})x_C k_C Q_C + D_C + (k_{A1}r_{A1} + k_{A2}r_{A2})Q_C
\]

It is clear that the adversary cannot make this signature because he does not know the random values \(r_{A1}\) or \(r_{A2}\) and solving discrete logarithm problem is requirement to obtain \(r_{A1}\) or \(r_{A2}\). This problem is a hard problem, so the proposed protocol is resistant to Unknown key attack.

- **Key Compromise Impersonation Attack:** In this attack the active adversary C who knows A’s long-term private key wants to impersonate B to A. In the proposed key agreement protocol if the adversary who knows \(X_A = \{D_A, x_A\}\) wants to execute this attack, he should make the following signature on the \((T_{B1}, T_{B2})\):

\[
S_B = (k_{B1} + k_{B2})(x_B k_B Q_B + D_B) + (k_{B1}r_{B1} + k_{B2}r_{B2})Q_B
\]

Because he does not know B’s private key, \(X_B = \{D_B, x_B\}\), it is clear that he cannot compute the signature \(S_B\). So the proposed multiple key agreement protocol is not vulnerable to key compromise impersonation attack.

- **Perfect forward secrecy:** This property emphasizes that the previous session key should not be expose by revealing the long-term private key of both entities. In the proposed protocol the adversary who knows both long-term private keys \(X_A = \{D_A, x_A\}\) and \(X_B = \{D_B, x_B\}\) cannot compute the previous session keys because the session keys are computed using the random values \(r_{A1}\) and \(r_{B1}\), and having the private keys doesn’t help to the adversary to find the random values \(r_{A1}\) or \(r_{B1}\). So under the intractability of the discrete logarithm problem assumption, the proposed protocol satisfies perfect forward secrecy.

- **Key Escrow:** The key escrow property in the certificateless key agreement protocols means that the Key Generation Center (KGC) who has the part of the users’ private key cannot obtain session keys established between the users. In the proposed scheme the session keys are made only using random numbers, so KGC cannot obtain these keys whereas he knows the part of the users’ private key.

The computations of the proposed protocol for every entity are shown in the Table I. Our protocol is designed to establish \(n^2\) session keys for \(n\) random numbers.

### REFERENCES


http://uwspace.uwaterloo.ca/bitstream/10012/4156/1/Swanson_Colleen.pdf


Mohammad sabzinejad Farash received the B.Sc. degree in Electronic Engineering from Shahid Chamran College of Kerman in 2006, and the M.Sc. degree in Communication Engineering from I.Hussein University in 2009. Currently, He is a Ph.D. candidate in Applied Mathematics at the Department of Mathematics and Computer Sciences of Tarbiat Moallem University in Iran. His research interests are Security Protocols and Provable Security Models.

Mahmoud Ahmadian Attari received the combined B.Sc. and M.Sc. degree in Electrical Engineering from University of Tehran, Iran in 1977. He also received the Ph.D. degree in Digital Communication Systems from University of Manchester in 1997. He currently is an Assistant Professor at Electrical Faculty of K.N.Toosi University of Technology. His research interests are Coding Theories and Cryptography.

Majid bayat is a Ph.D. candidate in the Department of Mathematics and Computer Sciences at Tarbiat Moallem university in Tehran, Iran. He is presently a Research Assistant of Tarbiat Moallem university and Information Systems and Security Lab (ISSL) of Sharif University in Tehran, Iran. His research interests include Information Security and Cryptography.