# Two improved authenticated multiple key exchange protocols

Feng LIU

School of Mathematics & Information, Ludong University, Yantai 264025, China

*E-mail: liufeng@ldu.edu.cn (2010-05)*

*tel.: +86 535 6659585*

**Abastract:** Many authenticated multiple key exchange protocols were published in recent years. In 2008, Lee et al. presented an authenticated multiple key exchange protocol based on bilinear pairings. However, Vo et al. demonstrated an impersonation attack on the protocol , and it failed to provide authenticity and perfect forward secrecy as they had claimed. Later, Vo et al. proposed their enhancement protocol conforming which conforms to all desirable security properties. But, Vo's protocol required any party had held the public key each other, which required a large amount of storage. In this paper, we propose two new authenticated multiple key exchange protocols based on Lee's protocol, and makes them immune against Vo et al.'s attacks.

**Keywords:** Cryptography;authentication;key exchange;security;Bilinear pairing

## 1 Introduction

A key exchange protocol allows two or more parties to establish a shared key which can be used for encrypting communications over an insecure network. A two-party key agreement protocol is used to establish a common session key between two parties. Both parties contribute some information to derive the shared session key. The first key agreement protocol was proposed by Diffie and Hellman in 1976 [1]. However, the protocol does not enable authentication of the two parties and thus is susceptible to the man-in-the-middle attack. To solve the problem, Al-Riyami[2]presented several protocols some of which use pairing. Their protocols assure authenticity through use of certificates issued by a Certificate Authority (CA). The session keys are generated by both short-term keys and long-term keys. The signature of the CA assures that only the entities which are in possesion of the static keys are able to compute the session keys. Still, in a certificate system the participants must firstly verify the certificates before using the public key of a user, which requires a large amount of computing time and storage.

Authenticated key agreement protocols provide authentication of the participating parties and thus are attractive for practical implementation.In 2001, Harn and Lin[3]proposed an authentication key exchange protocol which employs the digital signature technique to achieve user authentication and does not require a one-way hash function.In the protocol,two parties generate multiple shared keys after running the key agreement protocol.More precisely, if two parties compute and transmit $n$ public keys of Diffie–Hellman protocol to each other, then $n^2-1$ session keys are shared between them. Later, Hwang et al. [4]proposed an efficient authentication key exchange protocol requiring less computation than Harn and Lin's scheme [3]. Nevertheless, the scheme [4] was broken by Lee and Wu [5]by the modification attack. Recently, Lee et al. [6]proposed two authenticated multiple key exchange protocols: one is based on ECC and the other is based on bilinear pairings. These protocols let two parties share not only one but also four session keys in authenticated manner.However, Vo et al.[7]demonstrated an impersonation attack on Lee's pairing-based authenticated key exchange protocol. They also showed that, using a long-term public key of an entity only, any attacker can impersonate the party to agree some

session keys with another party. Consequently, Lee et al.'s protocol fails to provide authenticity as they had claimed. Furthermore, they indicated that perfect forward secrecy of Lee's protocol was not guaranteed. Thus, Vo et al. proposed a simple modification to the protocol which could withstand their own attacks.

In this paper we examine the two-party authenticated key exchange protocols using pairing operations from[5,7]. Then, we propose two new authenticated multiple key exchange protocols based on Lee et al.'s [6]protocol. In contrast to the original protocol, the proposed protocols are immune against Vo et al.'s key compromise impersonation attack, while being more efficient.

The rest of the paper is organized as follows: Section 2 briefly explains preliminary concepts, i.e. bilinear maps and the associated computational problems. Section 3 reviews Lee's multiple key exchang protocol,the attack on the protocol proposed by Vo et al., as well as the weakness of the Vo's protocol. and analyzes their security. Our proposed protocols are described in Section 4 with the corresponding security and efficiency discussion. In Section 5, the efficiency comparison of the proposed protocols and competitive protocol is conducted. Finally, a conclusion is drawn in Section 6.

## 2 Preliminaries

In this section, we briefly describe preliminaries which are needed later in the paper. We give the basic definition and properties of bilinear pairings, the computational problems which are fundamental when discussing authenticated key agreement protocols.

### 2.1 Bilinear Pairings

Let $G_1$ be an additive group generated by $P$, whose order is a prime $q$, and $G_2$ be a multiplicative group of the same order $q$; a bilinear pairing is a map

$$e: G_1 \times G_1 \to G_2$$

with the following properties :

- Bilinear: for all $P, Q \in G_1$ and $e(c_1P, c_2Q) = e(P, Q)^{c_1c_2}$.
- Non-degenerate: there exists $P \in G_1$ such that $e(P, P) \neq 1$.
- Computable: given $P, Q \in G_1$, there is an efficient algorithm to compute $e(P, Q)$.

### 2.2 Computational problems

- Computational Diffie-Hellman (*CDH*) problem: given a triple

  $$(P, c_1P, c_2P) \in G_1$$

  for $c_1, c_2 \in Z_q^*$, find the element $c_1c_2P$.

- Decision Diffie-Hellman (*DDH*) problem: given a quadruple

  $$e(P, c_1P, c_2P, c_3P) \in G_1$$

  for $c_1, c_2, c_3 \in Z_q^*$, decide whether $c_3 = c_1c_2 \bmod q$ or not.

- Gap Diffie-Hellman (*GDH*) problem: a class of problems where the *CDH* problem is hard but the *DDH* problem is easy.

Groups where the *CDH* problem is hard but the *DDH* problem is easy are called *GDH* groups.

## 3. Lee's authenticated key exchange protocol based on bilinear pairings

This section briefly reviews the two-party multiple protocol developed by Lee[5], and Vo et al.'s key compromise impersonation attack on it, and explicates the weaknesses of Vo et al.'s protocol. Let $\mathcal{A}$ and $\mathcal{B}$ be two communication parties.

### 3.1 Lee's two-party multiple protocol from pairings

We firstly review Lee's multiple key exchange protocol based on bilinear pairings.

*Initiate*

Let $X_{\mathcal{U}} \in Z_q^*$ and $Y_{\mathcal{U}} (= X_{\mathcal{U}} P)$ be $\mathcal{U}$ 's long-term private key and long-term public key, $Cert(Y_{\mathcal{U}})$ be the certificate of $\mathcal{U}$ 's long-term public key signed by a trusted party ($\mathcal{TP}$)

*Ex-massage*

$\mathcal{A}$ :chooses $a_1, a_2 \in \mathbb{Z}_q^*$,and computes $T_{\mathcal{A}1} = a_1 P, T_{\mathcal{A}2} = a_2 P, S_{\mathcal{A}} = (a_1 K_{\mathcal{A}1} + a_2 K_{\mathcal{A}2}) T_{\mathcal{A}1} + X_{\mathcal{A}} T_{\mathcal{A}2}$;

$\mathcal{A} \to \mathcal{B}$ : $\{T_{\mathcal{A}1}, T_{\mathcal{A}2}, S_{\mathcal{A}}, Cert(Y_{\mathcal{A}})\}$ ,where $K_{\mathcal{A}i}$ is the $x$-coordinate value of $T_{\mathcal{A}i}$.

$\mathcal{B}$ : chooses $b_1, b_2 \in \mathbb{Z}_q^*$,and computes $T_{\mathcal{B}1} = b_1 P, T_{\mathcal{B}2} = b_2 P, S_{\mathcal{B}} = (b_1 K_{\mathcal{B}1} + b_2 K_{\mathcal{B}2}) T_{\mathcal{B}1} + X_{\mathcal{B}} T_{\mathcal{B}2}$;

$\mathcal{B} \to \mathcal{A}$ : $\{T_{\mathcal{B}1}, T_{\mathcal{B}2}, S_{\mathcal{B}}, Cert(Y_{\mathcal{B}})\}$ ,where $K_{\mathcal{B}i}$ is the $x$-coordinate value of $T_{\mathcal{B}i}$.

*Co-keys*

$\mathcal{A}$ : $e(S_{\mathcal{B}}, P) \stackrel{?}{=} e(K_{\mathcal{B}1} T_{\mathcal{B}1} + K_{\mathcal{B}2} T_{\mathcal{B}2}, T_{\mathcal{B}1}) e(T_{\mathcal{B}2}, Y_{\mathcal{B}})$

$K_{11} = e(a_1 T_{\mathcal{B}1}, Y_{\mathcal{A}} + Y_{\mathcal{B}})$;

$K_{12} = e(a_1 T_{\mathcal{B}2}, Y_{\mathcal{A}} + Y_{\mathcal{B}})$;

$K_{21} = e(a_2 T_{\mathcal{B}1}, Y_{\mathcal{A}} + Y_{\mathcal{B}})$;

$K_{22} = e(a_2 T_{\mathcal{B}2}, Y_{\mathcal{A}} + Y_{\mathcal{B}})$.

$\mathcal{B}$ : $e(S_{\mathcal{A}}, P) \stackrel{?}{=} e(K_{\mathcal{A}1} T_{\mathcal{A}1} + K_{\mathcal{A}2} T_{\mathcal{A}2}, T_{\mathcal{A}1}) e(T_{\mathcal{A}2}, Y_{\mathcal{A}})$

$K_{11} = e(T_{\mathcal{A}1} b_1, Y_{\mathcal{A}} + Y_{\mathcal{B}})$;

$K_{12} = e(T_{\mathcal{A}1} b_2, Y_{\mathcal{A}} + Y_{\mathcal{B}})$;

$K_{21} = e(T_{\mathcal{A}2} b_1, Y_{\mathcal{A}} + Y_{\mathcal{B}})$;

$K_{22} = e(T_{\mathcal{A}2} b_2, Y_{\mathcal{A}} + Y_{\mathcal{B}})$.

### 3.2 Vo et al.'s key-compromise impersonation attack

Vo et al.[7] demonstrated an impersonation attack on Lee's protocol.[5]. And,they showed that, using a long-term public key of an party only, any attacker could impersonate the party to agree some session keys with another party. For example, they analyzed $S_{\mathcal{A}}$ as follows:

$$S_{\mathcal{A}} = (a_1 K_{\mathcal{A}1} + a_2 K_{\mathcal{A}2}) T_{\mathcal{A}1} + X_{\mathcal{A}} T_{\mathcal{A}2} = (a_1 K_{\mathcal{A}1} + a_2 K_{\mathcal{A}2}) T_{\mathcal{A}1} + a_2 Y_{\mathcal{A}}$$

Checking the final equation,any attacker who wants to impersonate Alice could compute $S_{\mathcal{A}}$ directly from Alice's long-term public key without knowing Alice's long-term private key. Consequently, Lee et al.'s protocol fails to provide authenticity as they have claimed.

Furthermore,Vo indicated that perfect forward secrecy of their protocol was not guaranteed. When attackers know long-term private keys of $\mathcal{A}$ and $\mathcal{B}$ , $x_{\mathcal{A}}$ and $x_{\mathcal{B}}$ , respectively, the attackers easily compute the previous session keys as follows:

$$K_{ij} = e(a_i T_{\mathcal{B}j}, Y_{\mathcal{A}} + Y_{\mathcal{B}}) = e(T_{\mathcal{B}j}, a_i (X_{\mathcal{A}} + X_{\mathcal{B}}) P) = e(T_{\mathcal{B}j}, (X_{\mathcal{A}} + X_{\mathcal{B}}) T_{\mathcal{A}i})$$

### 3.3 The weakness of Vo's authenticated protocol from pairings

Based on their observation Vo et al. had just made about why the attacks were feasible, they proposed that their revised protocol should be modified in a minimal way to Lee's protocol. Unfortunately, in Vo's enhanced protocol each participant must firstly verify the certificates before using the public key of a user, which required a large amount of computing time and storage[8].

## 4. Proposed multiple key agreement protocols

We note that a distinctive feature of Lee's protocol is that no secure channels between $\mathcal{TP}$ and the participants are assumed. All communication is done over (authenticated) public channels using public key signature. And, the initialization of Lee is done without any interaction between the $\mathcal{TP}$ and the participants. In fact, participants may enter or leave theprotocol *dynamically*, the only requirement is that a participant holds a registered public key. And, compared to Vo's protocol, we try to decrease the requirement for storing public keys.

Based on the above observations, we propose that our enhanced protocols should be modified in a hybrid way to avoid Vo's attacks.

We now describe the revised protocols,as follows:

### 4.1. Protocol 1

*Initiate*

Let $X_{\mathcal{U}} \in Z_q^*$ and $Y_{\mathcal{U}} (= X_{\mathcal{U}} P)$ be $\mathcal{U}$ 's long-term private key and long-term public key, $Cert(Y_{\mathcal{U}})$ be the certificate of $\mathcal{U}$ 's long-term public key signed by a trusted party( $\mathcal{TP}$ ).

*Ex-massage*

$\mathcal{A}$ :chooses $a_1$ , $a_2 \in \mathbb{Z}_q^*$ ,and computes

$$T_{\mathcal{A}1} = a_1 Y_{\mathcal{A}} \text{ , } T_{\mathcal{A}2} = a_2 Y_{\mathcal{A}} \text{ , } S_{\mathcal{A}} = (a_1 K_{\mathcal{A}1} + a_2 K_{\mathcal{A}2}) T_{\mathcal{A}1} + X_{\mathcal{A}} T_{\mathcal{A}2} \text{ ;}$$

$\mathcal{A} \to \mathcal{B}$ : $\{T_{A1}, T_{A2}, S_A, Cert(Y_A)\}$ ,where $K_{Ai}$ is the $x$-coordinate value of $T_{Ai}$.

$\mathcal{B}$ : chooses $b_1, b_2 \in \mathbb{Z}_q^*$,and computes

$T_{B1} = b_1 Y_B$ , $T_{B2} = b_2 Y_B$ , $S_B = (b_1 K_{B1} + b_2 K_{B2}) T_{B1} + X_B T_{B2}$ ;

$\mathcal{B} \to \mathcal{A}$ : $\{T_{B1}, T_{B2}, S_B, Cert(Y_B)\}$ ,where $K_{Bi}$ is the $x$-coordinate value of $T_{Bi}$.

*Co-keys*

$\mathcal{A}$ : $e(S_B, Y_B) \stackrel{?}{=} e(K_{B1} T_{B1} + K_{B2} T_{B2}, T_{B1}) e(T_{B2}, Y_B)$

$K_{11} = e(a_1 X_A T_{B1}, Y_A + Y_B)$ ;

$K_{12} = e(a_1 X_A T_{B2}, Y_A + Y_B)$ ;

$K_{21} = e(a_2 X_A T_{B1}, Y_A + Y_B)$ ;

$K_{22} = e(a_2 X_A T_{B2}, Y_A + Y_B)$ .

$\mathcal{B}$ : $e(S_A, Y_A) \stackrel{?}{=} e(T_{A1}, K_{A1} T_{A1} + K_{A2} T_{A2},) e(T_{A2}, Y_A)$

$K_{11} = e(T_{A1} X_B b_1, Y_A + Y_B)$ ;

$K_{12} = e(T_{A1} X_B b_2, Y_A + Y_B)$ ;

$K_{21} = e(T_{A2} X_B b_1, Y_A + Y_B)$ ;

$K_{22} = e(T_{A2} X_B b_2, Y_A + Y_B)$ .

### 4.1.1. Security analysis

The security of Protocol 1 is based on the difficulty of computing the discrete logarithm problem and the Diffie – Hellman protocol. We will firstly discuss that an adversary is not able to derive the secret keys using the transmitted messages $\{T_{A1}, T_{A2}, S_A, Cert(Y_A)\}$ and $\{T_{B1}, T_{B2}, S_B, Cert(Y_B)\}$ . An adversary would have to separately compute $X_A$ and $a_i$ from $Y_A (= X_A P)$ and $T_{Ai} (= a_i Y_A)$ which would be equivalent to solving the discrete logarithm problem. The same applies when the adversary tries to find private key from $S_A$ .

Additionally, we will show that Protocol 1 satisfies the security properties described in Section 3.2 and thus keeps merits of the original protocol.

*Key-compromise impersonation.* Let us consider the following scenario: $\mathcal{A}$ 's secret key is disclosed, an adversary obtains the secret key and tries to impersonate $\mathcal{B}$ to $\mathcal{A}$ . She would have to compute $K_{ij} = e(a_i b_j X_A Y_B, Y_A + Y_B)$ to impersonate $\mathcal{B}$ and it must be computed using $\mathcal{A}$ 's shotr-term

key $a_i$ which the adversary is not able to compute from $T_{\mathcal{A}i}(=a_i Y_{\mathcal{A}})$ since she would have to solve the CDH problem.

*Perfect Forward Secrecy.* Let us assume the secret keys $X_{\mathcal{A}}$ and $X_{\mathcal{B}}$ are disclosed and the adversary tries to compute the key $K_{ij}=e(a_i b_j X_{\mathcal{A}} X_{\mathcal{B}} P, Y_{\mathcal{A}}+Y_{\mathcal{B}})$ .In order to be able to compute the key, the adversary would have to compute $a_i X_{\mathcal{A}} T_{\mathcal{B}j}$ or $b_j X_{\mathcal{B}} T_{\mathcal{A}i}$ .For this purpose she would have to know $a_i$ or $b_j$ which she cannot derive as it would be equal to solving the bilinear discrete logarithm problem. Therefore the proposed protocol provides perfect forward secrecy.

### 4.2. Protocol 2

In this section we describe the second proposed protocol, namely Protocol 2.

*Initiate*

Let $X_{\mathcal{U}} \in Z_q^*$ and $Y_{\mathcal{U}}(=X_{\mathcal{U}} P)$ be $\mathcal{U}$ 's long-term private key and long-term public key, $Cert(Y_{\mathcal{U}})$ be the certificate of $\mathcal{U}$ 's long-term public key signed by a trusted party ($\mathcal{TP}$)

*Ex-massage*

$\mathcal{A}$ :chooses $a_1, a_2 \in \mathbb{Z}_q^*$,and computes $T_{\mathcal{A}1}=a_1 P, T_{\mathcal{A}2}=a_2 P, S_{\mathcal{A}1}=a_1 X_{\mathcal{A}}+a_2, S_{\mathcal{A}2}=a_2 X_{\mathcal{A}}+a_1$;

$$\mathcal{A} \rightarrow \mathcal{B}: \left\{T_{\mathcal{A}1}, T_{\mathcal{A}2}, S_{\mathcal{A}1}, S_{\mathcal{A}2}, Cert(Y_{\mathcal{A}})\right\}.$$

$\mathcal{B}$ : chooses $b_1, b_2 \in \mathbb{Z}_q^*$,and computes $T_{\mathcal{B}1}=b_1 P, T_{\mathcal{B}2}=b_2 P, S_{\mathcal{B}1}=b_1 X_{\mathcal{B}}+b_2, S_{\mathcal{B}2}=b_2 X_{\mathcal{B}}+b_1$;

$$\mathcal{B} \rightarrow \mathcal{A}: \left\{T_{\mathcal{B}1}, T_{\mathcal{B}2}, S_{\mathcal{B}1}, S_{\mathcal{B}2}, Cert(Y_{\mathcal{B}})\right\}.$$

*Co-keys*

$$\mathcal{A}: \quad e(P,(S_{\mathcal{B}1}+S_{\mathcal{B}2})P-(T_{\mathcal{B}1}+T_{\mathcal{B}2})) \stackrel{?}{=} e(Y_{\mathcal{B}}, T_{\mathcal{B}1}+T_{\mathcal{B}2})$$

$$K_{11}=e(a_1 X_{\mathcal{A}}(S_{\mathcal{B}1}P-T_{\mathcal{B}2}), Y_{\mathcal{A}}+Y_{\mathcal{B}});$$

$$K_{12}=e(a_1 X_{\mathcal{A}}(S_{\mathcal{B}2}P-T_{\mathcal{B}1}), Y_{\mathcal{A}}+Y_{\mathcal{B}});$$

$$K_{21}=e(a_2 X_{\mathcal{A}}(S_{\mathcal{B}1}P-T_{\mathcal{B}2}), Y_{\mathcal{A}}+Y_{\mathcal{B}});$$

$$K_{22}=e(a_2 X_{\mathcal{A}}(S_{\mathcal{B}2}P-T_{\mathcal{B}1}), Y_{\mathcal{A}}+Y_{\mathcal{B}}).$$

$$\mathcal{B}: e(P,(S_{\mathcal{A}1}+S_{\mathcal{A}2})P-(T_{\mathcal{A}1}+T_{\mathcal{A}2})) \stackrel{?}{=} e(Y_{\mathcal{A}}, T_{\mathcal{A}1}+T_{\mathcal{A}2})$$

$$K_{11}=e((S_{\mathcal{A}1}P-T_{\mathcal{A}2})X_{\mathcal{B}} b_1, Y_{\mathcal{A}}+Y_{\mathcal{B}});$$

$$K_{12} = e((S_{A1}P - T_{A2})X_B b_2, Y_A + Y_B);$$

$$K_{21} = e((S_{A2}P - T_{A1})X_B b_1, Y_A + Y_B);$$

$$K_{22} = e((S_{A2}P - T_{A1})X_B b_2, Y_A + Y_B).$$

#### 4.2.1. Security analysis

As in Protocol 1, the security of Protocol 2 is based on the difficulty of computing the bilinear discrete logarithm problem and the Diffie-Hellman protocol. First let us show that it is impossible for an adversary to derive the secret key if she eavesdrops on the transmitted messages $\{T_{A1}, T_{A2}, S_{A1}, S_{A2}\}$ and $\{T_{B1}, T_{B2}, S_{B1}, S_{B2}\}$ .As $S_{Ai} = a_i X_A + a_j$ has two unknown variables $a_1$ and $a_2$ which are determined by $A$ , the adversary would have to separately compute $a_1$ and $a_2$ from $T_{A1} = a_1 P$ and $T_{A2} = a_2 P$ . Hence, it is equivalent to solving the bilinear discrete logarithm problem.

Noticing $S_{Ai}P - T_{Aj} = a_i Y_A$ and $S_{Bi}P - T_{Bj} = a_i Y_B$ , the protocol 2 is actually identical with the protocol 1.So, the protocol 2 has identical security as the protocol 1.

## 5. Efficiency comparison

In this section we compare the efficiency of the proposed improved protocols and Lee's authenticated key agreement protocol.

The efficiency comparison is summarized in Table 1. The comparison includes operations which have to be carried out by each party and is divided into the following groups:

• Modular data addition( $Da$ )and modular point addition( $Pa$ ) are computationally less expensive.

• Modular data-point multiplications( $DPm$ ) and pairing computation( $e$ ) are more expensive and thus have greater impact on the efficiency of the protocol.

*Table 1 Computation effort per user*

| Step | Lee[6] | Our protocol 1 | Our protocol 2 |
|---|---|---|---|
| Short-term public keys | $2DPm$ | $2DPm$ | $2DPm$ |
| Verification | $3e + 2Pa + 2DPm$ | $3e + 2Pa + 2DPm$ | $2e + 3Pa + DPm$ |
| Key computation (iff one key) | $e + DPm + Pa$ | $e + DPm + Pa$ | $e + DPm + 2Pa$ |

From Table 1 we can observe that Protocol 1 is the same efficient as Lee et al.'s original protocol, and it avoids Vo et al.'s attacks. Protocol 2 is even more efficient the Lee's protocol; i.e. any user has to compute 2 bilinear pairing computation.

## 6. Conclusion

We have proposed two new authenticated multiple key exchange protocols: the first protocol denoted as Protocol 1 is based on CDH problem from bilinear pairings and sanitizes the weakness

which leads to Vo et al.'s key compromise impersonation attack.We have shown that Protocol 1 is the same efficient and at the same time conforms to all the desirable security properties. Furthermore we have proposed an efficiently improved protocol based on Lee's protocol denoted as Protocol 2. It is more efficient than the original protocol, while keeping all the security merits. The efficiency advantages of both proposed protocols are considerable, while conforming to all the desirable security properties for authenticated key exchange protocols.

**Acknowledgements:**

This work was supported by the Ludong University Research Program under Grant NO. L20082702

**References**

[1] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Transactions on Information Theory, 22 (1976) 644 – 654.

[2] S S Al-Riyami and K G Paterson. Tripartite authenticated key agreement protocols from pairings. Cryptology eprint Archive 2002, Report 2002/035.

[3] Harn L, Lin H-Y. Authenticated key agreement without using one-way hash function. Electron Lett ,2001;37(10):629-630.

[4] Hwang R J, Shiau S H, Lai C H. An enhanced authentication key exchange protocol. Advanced information networking and applications, 2003. In: Proceedings of the 17th international conference on AINA 2003; p. 202 – 205.

[5] Lee N-Y, Wu C-N. Improved authentication key exchange protocol without using one-way hash function. ACM Operat Syst Rev, 2004,38(2):85-92.

[6] Lee N-Y, Wu C-N, Wang C-C. Authenticated multiple key exchange protocols based on elliptic curves and bilinear pairings. Comput Electr Eng, 2008,34(1):12 – 20.

[7] Vo D-L, Lee H, Yeun C-Y, Kim K. Enhancements of authenticated multiple key exchange protocol based on bilinear pairings.Computers and Electrical Engineering,36 (2010) 155-159

[8] F. Liu. One-round and authenticated three-party multiple key exchange protocol from parings. Cryptology eprint Archive 2010, Report 2010/239.