Security of Multiple-Key Agreement Protocols and Propose an Enhanced Protocol

Mohammad Sabzinejad Farash[1], Mahmoud Ahmadian Attari[2] and Majid Bayat[1]

[1]Department of Mathematics and Computer Sciences, Tarbiat Moallem University, Tehran, Iran

[2]Faculty of Electrical and computer Engineering, K.N. Toosi University of Technology, Tehran, Iran

Email: sabzinejad@tmu.ac.ir, mahmoud@eetd.kntu.ac.ir, bayat@tmu.ac.ir

Abstract

Multiple key agreement protocols produce several session keys instead of one session key. Most of the multiple key agreement protocols do not utilize the hash functions in the signature schemes used for identification. Not using hash function in these protocols causes that the protocols do not satisfy some requirement security properties. In this paper we review the multiple key agreement protocols and perform attacks on some of them. Then we introduce a new multiple key agreement protocol and show that the proposed protocol is more secure than the existent multiple key agreement protocols.

Keywords: Key agreement protocols, Multiple-key agreement protocols, Signature schemes

1. Introduction

Cryptography helps us to make a secure communication in public networks. The secret key plays an essential role in the cryptosystems such that revealing the secret key causes the cryptographic system to be compromised. Therefore how to exchange the secret key is very important in cryptographic applications. One of the considerable methods for secret key exchanging is key agreement protocols. These protocols enable two or more users of any public networks to share a secret common key together.

The first key agreement protocol was introduced by Diffiee and Hellman [4]. But this protocol is vulnerable to man in the middle attack because two participants of the protocol do not verify identity of each other. Typical remedy for this problem is to use public key cryptosystems such as public key infrastructure or identity based cryptography. Menezes et. al. [10] proposed a key agreement protocol named as MQV that does not use a hash function for making digital signature. This protocol was standardized in ANSI X9.42 [1], ANSI X9.63 [2] and IEEE [9]. Harn and Lin [5] introduced the first multiple key agreement protocol that is based on the idea of using the signature without hash functions same as MQV protocol. In multiple key agreement protocols multiple keys are agreed instead of one key in typical key agreement protocols. Multiple key agreement protocols are considerable because cost of computation and communication is less than usual key agreement protocols for one shared key.

Yen and Joye [15] showed that the Harn-Lin protocol is insecure against forgery attack and introduced an improved protocol. Next Wu et. al. [16] showed that the Yen-Joye protocol is as insecure as Harn-Lin protocol and then introduced an enhanced protocol which used hash function in contradiction to Harn-Lin Protocol; nevertheless, the problem still remained in their protocol. Harn and Lin [6] again proposed an improved protocol to overcome the posed weaknesses of their first protocol. After that Zhou et. al. [17] claimed that the second Harn-Lin protocol is insecure against forgery attack.

In this paper we review most of the multiple key agreement protocols that have been introduced until now and perform attacks on some of them. Finally we introduce a new multiple key agreement protocol and show that the proposed protocol is more secure than the existent multiple key agreement protocols.

There are some security properties that are recommended for key agreement protocols [3]. Here we review them as follows. Let A and B are two participants that are intended to share a common secret key by 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:

- *Implicit key confirmation*: A key agreement protocol has this property if the both participants are assured that only the other participant can compute the secret common key.
- *Explicit key confirmation*: This means that the both participants are assured that the other participant have computed the secret common key.

2. Review of the multiple key agreement protocols

In this section we review certificate based multiple key agreement protocols. In these protocols two participants authenticate each other after sending and receiving a message and agree on multiple secret common keys. The notations used through this paper are presented in **Table 1**. Because the weaknesses of the key agreement protocols reviewed in this paper are arisen from the utilized digital signature schemes, we only study the digital signature scheme according to **Table 2**. Four columns are represented in **Table 2**, from left hand side, the firs column is the protocol name, the second column is user's short-term public keys, the third column shows the digital signature and the signature verification equation is presented in column 4. In **Table 3** weaknesses and the number of shared keys of each protocol are represented. Some of these weaknesses notated by ,*, are introduced by the authors of this paper which are explained in the next sections.

Table 1. The Notations

Notation	Description
g	Generator of multiplicative group *G* with large prime order *q*
x_A, x_B	Long-term private keys for participants *A* and *B*.
y_A, y_B	Long-term public keys for participants *A* and *B*.
r_A, r_B	Short-term private keys that is generated in each session.
t_A, t_B	Public-term private keys that is generated in each session.
$H()$	One-way hash function
$K_{AB} = g^{x_A x_B}$	Long-term private common Diffie-Hellman key.
K	Session key

Table 2.Comparison of multiple key agreement protocols

Protocol	Number of session keys	Weaknesses
HL98 [5]	n^2 - 1	Foraged signature [15]
YJ [15]	n^2 - 1	Foraged signature [16]
WHH [16]	n^2 - 1	Foraged signature [14]
HL01 [6]	n^2 - 1	Foraged signature [17]
ZFL [17]	n^2 - 1	Foraged signature [14]
YSH [14]	n^2 - 1	Unknown key attack [14]
Tseng [13]	n^2	Foraged signature [11] and Key compromise impersonation attack [*]
Shao [11]	n^2	Unknown key attack [12]
HC [7]	n^2	Key compromise impersonation attack [*]
HCH [8]	n^2	Long-term private keys and one of the four session keys give the other three session keys [*]

Table 3. Summarization of multiple key agreement protocols

Protocol	Signature	Short-term public key	Verification
HL98 [5]	$s_A = x_A - g^{t_{A1}t_{A2}}(r_{A1} + r_{A2})$	$t_{A1/2} = g^{r_{A1/2}}$	$y_A \overset{?}{=} g^{s_A} \cdot (t_{A1} \cdot t_{A2})^{g^{t_{A1} \cdot t_{A2}}}$
YJ [15]	$s_A = x_A - (t_{A1} \cdot t_{A2})(r_{A1} + r_{A2})$	$t_{A1/2} = g^{r_{A1/2}}$	$y_A \overset{?}{=} g^{s_A} \cdot (t_{A1} \cdot t_{A2})^{t_{A1} \cdot t_{A2}}$
WHH [16]	$s_A = x_A - H(t_{A1} \cdot t_{A2})(r_{A1} + r_{A2})$	$t_{A1/2} = g^{r_{A1/2}}$	$^{**}\ y_A \overset{?}{=} g^{s_A} \cdot (t_{A1} \cdot t_{A2})^{h(t_{A1} \cdot t_{A2})}$
HL01 [6]	$s_A = x_A - t_{A1}r_{A1} - t_{A2}r_{A2}$	$t_{A1/2} = g^{r_{A1/2}}$	$y_A \overset{?}{=} g^{s_A} \cdot t_{A1}^{t_{A1}} \cdot t_{A2}^{t_{A2}}$
ZFL [17]	$s_A = x_A - (t_{A1} + t_{A2})(r_{A1} + r_{A2})$	$t_{A1/2} = g^{r_{A1/2}}$	$y_A \overset{?}{=} g^{s_A} \cdot (t_{A1} \cdot t_{A2})^{(t_{A1} + t_{A2})}$
YSH [14]	$s_A = x_A - (t_{A1} \oplus t_{A2})(r_{A1} + r_{A2})$	$t_{A1/2} = g^{r_{A1/2}}$	$y_A \overset{?}{=} g^{s_A} \cdot (t_{A1} \cdot t_{A2})^{(t_{A1} \oplus t_{A2})}$
Tseng [13]	$s_A \cdot t_A = x_A - t_{A1}(r_{A1} + r_{A2})$	$t_A = g^{r_{A1} + r_{A2}}$ $t_{A1/2} = y_B^{r_{A1/2}}$	$t_A = t_{A1}^{x_B^{-1}} \cdot t_{A2}^{x_B^{-1}}$, $y_A \overset{?}{=} t_A^{t_{A1}} \cdot g^{s_A t_A}$
Shao [11]	$s_A = x_A \cdot t_A - (t_{A1} + t_{A2})(r_{A1} + r_{A2})$	$t_A = g^{r_{A1} + r_{A2}}$ $t_{A1/2} = y_B^{r_{A1/2}}$	$t_A = t_{A1}^{x_B^{-1}} \cdot t_{A2}^{x_B^{-1}}$, $y_A^{t_A} \overset{?}{=} t_A^{(t_{A1} + t_{A2})} \cdot g^{s_A}$
HC [7]	$s_A \oplus K_{AB} = (t_{A1} - t_{A2})x_A$ $- (t_{A1} \oplus t_{A2})(r_{A1} + r_{A2})$	$t_{A1/2} = g^{r_{A1/2}}$	$y_A^{(t_{A1} - t_{A2})} = g^{s_A \oplus K_{AB}} \cdot (t_{A1} \cdot t_{A2})^{(t_{A1} \oplus t_{A2})}$
HCH [8]	$h_A = (y_B)^{r_{A1}} \cdot t_{A2}$ $s_A = x_A - h_A(r_{A1} + r_{A2})$	$t_{A1/2} = g^{r_{A1/2}}$	$h_A = (t_{A1})^{x_B} \cdot t_{A2}$ $y_A \overset{?}{=} g^{s_A} \cdot (t_{A1} \cdot t_{A2})^{h_A}$

3. The proposed key compromise impersonation attack on Tseng's protocol

For key compromise impersonation attack on Tseng's protocol [13], the adversary computes $\left(t_{B_1}, t_{B_2}, s_B\right)$ as follows:

$$t_{B2} = \left(y_B^{t_{B1}^{-1}}\right)^{x_A} \quad , \quad t_{B1} = y_A \quad , \quad s_B \cdot t_B = -t_{B1} \quad , \quad t_B = g \cdot y_B^{t_{B1}^{-1}}$$

Then he sends $\left(t_{B_1}, t_{B_2}, s_B\right)$ to *A*. After receiving these values, *A* verifies the signature as follow,

$$t_{b1} = t_{B1}^{x_A^{-1}} = y_A^{x_A^{-1}} = g \quad , \quad t_{b2} = t_{B2}^{x_A^{-1}} = \left(\left(y_B^{t_{B1}^{-1}}\right)^{x_A}\right)^{x_A^{-1}} = y_B^{t_{B1}^{-1}}$$

$$t_B = t_{b1} \cdot t_{b2} = g \cdot y_B^{t_{B1}^{-1}}$$

$$y_B = t_B^{t_{B1}} \cdot g^{s_B t_B} = \left(g \cdot y_B^{t_{B1}^{-1}}\right)^{t_{B1}} \cdot g^{s_B t_B} = g^{t_{B1}} \cdot y_B \cdot g^{-t_{B1}} = y_B$$

As we observed in the above equations the user *A* verifies message $\left(t_{B_1}, t_{B_2}, s_B\right)$ have generated by user *B* while he has not a role in the protocol. Finally *A* computes the session keys as follows:

$$K_1 = t_{b1}^{r_{A1}} = g^{r_{A1}} \quad , \quad K_3 = t_{b2}^{r_{A1}} = \left(y_B^{t_{B1}^{-1}}\right)^{r_{A1}} = t_{A1}^{t_{B1}^{-1}}$$

$$K_2 = t_{b1}^{r_{A2}} = g^{r_{A2}} \quad , \quad K_4 = t_{b2}^{r_{A2}} = \left(y_B^{t_{B1}^{-1}}\right)^{r_{A2}} = t_{A2}^{t_{B1}^{-1}}$$

Because the adversary knows $\left(t_{B1}, t_{A1}, t_{A2}\right)$, he can easily compute K_3 and K_4. So Tseng's multiple key agreement protocol is vulnerable to key compromise impersonation attack.

4. The propose key compromise impersonation attack on HC protocol

The adversary, for key compromise impersonation attack on protocol HC [7], can select $\left(t_{B_1}, t_{B_2}\right)$ such that $t_{B_1} = t_{B_2}$. So $t_{B_1} - t_{B_2} = 0$, $t_{B_1} \oplus t_{B_2} = 0$ and the verification equation is as follow:

$$\begin{aligned} s_B \oplus K_{AB} &= (t_{B1} - t_{B2})x_B - (t_{B1} \oplus t_{B2})(r_{B1} + r_{B2}) \\ &= (0)x_B - (0)(r_{B1} + r_{B2}) = 0 \end{aligned}$$

Therefore the adversary can sign the equal values $\left(t_A, t_B\right)$ by $s_B = K_{AB}$ signature and whereas he knows *A*'s long-term private key, it is not difficult for him to compute $s_B = K_{AB} = y_B^{x_A}$. So the HC protocol is insecure against key compromise impersonation attack.

4.1. Review of HCH protocol

As we showed in **Table 2**, the HCH protocol [8] is the most secure multiple key agreement protocol. But in the following we show that HCH protocol has the same weakness as what Shim [12] proposed on Shao protocol [11]. Let the adversary has obtained long-term private key of the both participants, so he can easily compute the following values:

$$(r_{A1} + r_{A2}) = \left(x_A - s_A\right)\left(t_{A1}^{x_B} \cdot t_{A2}\right)^{-1}$$
$$(r_{B1} + r_{B2}) = \left(x_B - s_B\right)\left(t_{B1}^{x_A} \cdot t_{B2}\right)^{-1}$$

Then he computes the following equations:

$$t_{B1}^{(r_{A1}+r_{A2})} = g^{r_{A1}r_{B1}} \cdot g^{r_{A2}r_{B1}} = K_1 \cdot K_3 \qquad (1)$$
$$t_{B2}^{(r_{A1}+r_{A2})} = g^{r_{A1}r_{B2}} \cdot g^{r_{A2}r_{B2}} = K_2 \cdot K_4 \qquad (2)$$
$$t_{A1}^{(r_{B1}+r_{B2})} = g^{r_{A1}r_{B1}} \cdot g^{r_{A1}r_{B2}} = K_1 \cdot K_2 \qquad (3)$$
$$t_{A2}^{(r_{B1}+r_{B2})} = g^{r_{A2}r_{B1}} \cdot g^{r_{A2}r_{B2}} = K_3 \cdot K_4 \qquad (4)$$

So if the adversary can obtain one of the four session keys, he can compute the other three session keys. For example if the adversary knows K_1 he can obtain K_2 and K_3 from (1,3) and then compute K_4 from (2 or 4).

5. The proposed protocol

5.1. Description of the proposed protocol

The utilized signature in the proposed multiple key agreement protocol is based on the signature scheme of HCH [8]. Description of the proposed protocol showed in **Fig. 1** is as follows:

- *A* generates two random numbers r_{A1} and r_{A2} and computes the short-term public keys $t_{A1} = g^{r_{A1}}$, $t_{A1} = g^{r_{A1}}$ and t_A . Then she signs t_{A1} and t_{A2} as follows:

  $$s_A = x_A - \left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(r_{A1} + r_{A2}\right)$$

  She sends $\left(t_{A1}, t_{A2}, t_A, s_A\right)$ to *B*.

  Also *B* executes the same computation as *A* and sends $\left(t_{B1}, t_{B2}, t_B, s_B\right)$ to *A*.

- *A* upon receiving the message from *B* verifies *B*'s signature by checking the following equation:

$$y_B \overset{?}{=} g^{s_B} \cdot \left(t_B \cdot t_{B2}\right)^{\left(t_{B1}^{x_A} \cdot t_{B2}\right)}$$

  If the above equation verification fails *A* terminates the execution, otherwise she computes the session keys $K_{ij} = {t_{Bi}}^{r_{Aj}} = g^{r_{Aj} r_{Bi}}\ for\ \ i,j \in \{1,2\}$

- Also *B* upon receiving the message from *A* verifies *A*'s signature as follows:

$$y_A \overset{?}{=} g^{s_A} \cdot \left(t_A \cdot t_{A2}\right)^{\left(t_{A1}^{x_B} \cdot t_{A2}\right)}$$

  If the above equation verification fails *A* terminates the execution, otherwise she computes the session keys $K_{ij} = {t_{Aj}}^{r_{Bi}} = g^{r_{Aj} r_{Bi}}$ for $i,j \in \{1,2\}$

$$\begin{array}{ccc}
A & & B \\
\left(x_A, Y_A\right) & & \left(x_B, Y_B\right) \\
r_{A1}, r_{A2} \in_R Z_q^* & & r_{B1}, r_{B2} \in_R Z_q^* \\
t_{A1} = g^{r_{A1}},\ t_{A2} = g^{r_{A2}} & & t_{B1} = g^{r_{B1}},\ t_{B2} = g^{r_{B2}} \\
s_A = x_A - \left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(r_{A1} t_{A1} + r_{A2} t_{A2}\right) & & s_B = x_B - \left(y_A^{r_{B1}} \cdot t_{B2}\right)\left(r_{B1} t_{B1} + r_{B2} t_{B2}\right) \\
 & \xrightarrow{\left(t_{A1}, t_{A2}, s_A\right)} & \\
 & \xleftarrow{\left(t_{B1}, t_{B2}, s_B\right)} & \\
y_B \overset{?}{=} g^{s_B} \cdot \left(t_{B1}^{t_{B1}} \cdot t_{B2}^{t_{B2}}\right)^{\left(t_{B1}^{x_A} \cdot t_{B2}\right)} & & y_A \overset{?}{=} g^{s_A} \cdot \left(t_{A1}^{t_{A1}} \cdot t_{A2}^{t_{A2}}\right)^{\left(t_{A1}^{x_B} \cdot t_{A2}\right)} \\
K_{ij} = {t_{Bi}}^{r_{Aj}} = g^{r_{Aj} r_{Bi}} & & K_{ij} = {t_{Aj}}^{r_{Bi}} = g^{r_{Aj} r_{Bi}} \\
i,j \in \{1,2\} & & i,j \in \{1,2\}
\end{array}$$

Fig. 1. The proposed multiple key agreement protocol

5.2. Security analysis of the proposed protocol

In the following we discuss security analysis of the proposed protocol to show that it is more secure than the existent multiple key agreement protocols.

- *Known-Key security:* This 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, $K_{ij} = {t_{Bi}}^{r_{Aj}} = g^{r_{Aj} r_{Bi}}$ for

$i, j = 1,2$. It does not give adversary any useful information to compute the next session keys. Because for computing the session keys short-term private keys $r_{A_{1/2}}$ and $r_{B_{1/2}}$ that be changed in each session are used. So the proposed multiple key agreement protocol is secure against Known- Key attack.

- *Unknown key security*: In the section1 we illustrated this attack. The adversary C for executing this attack on the proposed protocol intercepts the sent message from A. Then he must sign the values (t_{A1}, t_{A2}) by using his private key as follow:

$$s_C = x_C - \left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(r_{A1}^2 + r_{A2}\right)$$

 It is clear that the adversary cannot make this signature because he does not know the random values r_{A_1} or r_{A_2} and solving discrete logarithm problem is requirement to obtain r_{A_1} or r_{A_2}. 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 wants to execute this attack, he should make the a signature on the (t_{B1}, t_{B2}) as:

$$s_B = x_B - \left(y_A^{r_{B1}} \cdot t_{B2}\right)\left(r_{B1}^2 + r_{B2}\right)$$

 Because he does not know B's private key $, x_B$, 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 exposed by revealing the long-term private key of both participants. In the proposed protocol the adversary who knows both long-term private keys x_A and x_B cannot compute the previous session keys because computing the session keys depends on knowing one of the short-term private keys of participants and this is equal to solving discrete logarithm problem. In addition the adversary by using both long-term private keys x_A and x_B cannot obtain the random values r_{A_i} or r_{Bj} from s_A. The equation of used digital signature scheme is represented in (5).

$$s_A = x_A - \left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(r_{A1} + r_{A2}\right) \tag{5}$$

 The adversary who knows the values $(t_{A1}, t_{A2}, x_A, x_B, s_A)$ transforms (5) to (6).

$$\left(t_{A1}^{x_B} \cdot t_{A2}\right)^{-1}\left(x_A - s_A\right) = \left(r_{A1}^2 + r_{A2}\right) \tag{6}$$

Left hand side of (6) is an obvious value for adversary but obtaining r_{A_1} or r_{A_2} is equal to exhaustive search in the group G and this is equal to solving discrete logarithm problem. So under the intractability of the discrete logarithm problem assumption, the proposed protocol satisfies perfect forward secrecy.

5.3. More precise analysis of the proposed protocol

Let adversary in our protocol multiplies S_A and S_B as follows:

$$\begin{aligned} x_A x_B &= \left(s_A - \left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(r_{A1}t_{A1} + r_{A2}t_{A2}\right)\right) \\ &\qquad \cdot\left(s_B - \left(y_A^{r_{B1}} \cdot t_{B2}\right)\left(r_{B1}t_{B1} + r_{B2}t_{B2}\right)\right) \\ &= s_A s_B - \left(s_A t_{B1} r_{B1} + s_A t_{B2} r_{B2}\right)\left(y_A^{r_{B1}} \cdot t_{B2}\right) \\ &\quad - \left(t_{A1} r_{A1} s_B + t_{A2} r_{A2} s_B\right)\left(y_B^{r_{A1}} \cdot t_{A2}\right) \\ &\quad + \begin{pmatrix} t_{A1} r_{A1} t_{B1} r_{B1} + t_{A1} r_{A1} t_{B2} r_{B2} \\ + t_{A2} r_{A2} t_{B1} r_{B1} + t_{A2} r_{A2} t_{B2} r_{B2} \end{pmatrix}\left(y_B^{r_{A1}} \cdot t_{A2}\right)\left(y_A^{r_{B1}} \cdot t_{B2}\right) \end{aligned}$$

Then we have the following equation:

$$\begin{aligned} g^{x_A x_B} = K_{AB} = g^{s_A s_B} \cdot \left(t_{B1}^{r_{B1}} \cdot t_{B2}\right)^{-s_A \cdot \left(t_{B1}^{x_A} \cdot t_{B2}\right)} \cdot \left(t_{A1}^{r_{A1}} \cdot t_{A2}\right)^{-s_B \left(t_{A1}^{x_B} \cdot t_{A2}\right)} \\ \cdot \left(K_1^{r_{A1} r_{B1}} \cdot K_2^{r_{A1}} \cdot K_3^{r_{B1}} \cdot K_4\right)^{\left(t_{B1}^{x_A} \cdot t_{B2}\right)\left(t_{A1}^{x_B} \cdot t_{A2}\right)} \end{aligned} \tag{7}$$

Sides of equation (7) are dependent to the both participant's private key. Therefore if the adversary can obtain all four session keys of a session he cannot compute K_{AB} without owning one of the both participant's long-term private key and this means that the participants are authorized to use all four(n^2 in a general case) session keys.

Let the adversary has obtained the both participant's long-term private key and wants to make the discussed attack on HCH protocol in section 2.3. In this case he computes the following equations:

$$\begin{aligned} (r_{A1}^2 + r_{A2}) &= \left(x_A - s_A\right)\left(t_{A1}^{x_B} \cdot t_{A2}\right)^{-1} \\ (r_{B1}^2 + r_{B2}) &= \left(x_B - s_B\right)\left(t_{B1}^{x_A} \cdot t_{B2}\right)^{-1} \end{aligned}$$

Then he computes the following equations:

$$t_{B1}^{(r_{A1}^2+r_{A2})} = g^{r_{A1}^2 r_{B1}} \cdot g^{r_{A2} r_{B1}} = K_1^{r_{A1}} \cdot K_3$$
$$t_{B2}^{(r_{A1}^2+r_{A2})} = g^{r_{A1}^2 r_{B2}} \cdot g^{r_{A2} r_{B2}} = K_2^{r_{A1}} \cdot K_4$$
$$t_{A1}^{(r_{B1}^2+r_{B2})} = g^{r_{A1} r_{B1}^2} \cdot g^{r_{A1} r_{B2}} = K_1^{r_{B1}} \cdot K_2$$
$$t_{A2}^{(r_{B1}^2+r_{B2})} = g^{r_{A2} r_{B1}^2} \cdot g^{r_{A2} r_{B2}} = K_3^{r_{B1}} \cdot K_4$$

According to the above equations the adversary who knows the three session keys cannot compute the fourth session key. Therefore the proposed protocol is more secure than HCH protocol and is the most secure multiple key agreement protocols (See **Table 2**). Note that in the proposed protocol each party generates two random numbers same as the previous multiple key agreement protocols and the added computation only is computing of t_A and t_B for *A* and *B* respectively.

6. Conclusion

In this paper we reviewed multiple key agreement protocols and made attacks on some of them. Then we introduced a new and efficient multiple key agreement protocol and we showed that the proposed protocol is the most secure and efficient multiple key agreement protocols. At the end we concluded that all key agreement protocols that use digital signature schemes without hash function do not completely satisfy all security properties and the proposed protocol that is the best multiple key agreement protocols still has a partial weakness.

References

[1] ANSI X9.42, "Agreement of Symmetric Algorithm Keys Using Diffie–Hellman," Working Draft, May 1998.

[2] ANSI X 9.63, "Elliptic Curve Key Agreement and Key Transport Protocols," Working Draft, July 1998.

[3] S. Blake-Wilson, D. Johnson, and A. Menezes, "Key agreement protocols and their security analysis," In Proc. of Sixth IMA International Conference on Cryptography and Coding, pages 30 – 45. Cirencester, UK, 1997.

[4] W. Diffiee, M. Hellman, "New Directions in Cryptography. In IEEE Transaction on Information Theory," IT-22 (6), pp. 644-654, 1976.

[5] L. Harn, H.-Y. Lin, "An authenticated key agreement protocol without using one-way function," In: Proceedings of eighth information security conference, Taiwan, May 1998; p. 155–60.

[6] L. Harn, H.-Y. Lin, "Authenticated key agreement without using one-way hash function," Electron Lett 2001; 37(10):629–30.

[7] H. Huang and C. Chang, "Enhancement of an Authenticated Multiple-Key Agreement Protocol Without Using Conventional One-Way Function," In CIS 2005, Part II, LNAI 3802, pp. 554 – 559, 2005. Springer-Verlag (2005).

[8] C.-J. Huang, S.-H. Chang and W.-H. Hsu, "Authenticated Key Agreement Protocol for Exchanging n2 Keys without Using One-way Hash Function," In NCS 全國計算機會議 DSpace at FCUniversity, available in:

(dspace.lib.fcu.edu.tw/bitstream/2377/3190/3/ce07ncs001999000185.pdf), (2006).

[9] IEEE P1363, "Standards Specifications for Public-Key Cryptosystems," Working Draft, July 1998.

[10] A.J. Menezes, M. Qu, and S.A. Vanstone, "Some key agreement protocols providing implicit authentication," In: Proceeding of the second workshop on selected areas in cryptography (SAC'95), 1995; pp. 22–32.

[11] Z. Shao, "Security of Robust Generalized MQV Key Agreement Protocol Without Using One-way Hash Functions," Computer Standards and Interfaces, Vol. 25, (2003) 431–436.

[12] K.-A. Shim, "Vulnerabilities of generalized MQV key agreement protocol without using one-way hash functions," In: Computer Standards & Interfaces, 29, (2007), 467–470.

[13] Y.-M. Tseng, "Robust Generalized MQV Key Agreement Protocol without Using One-way Hash Functions," Computer Standards and Interfaces, Vol. 24, (2002) 241–246

[14] H.-T. Yeh, H.-M. Sun, T. Hwang, "Improved authenticated multiple-key agreement protocol," in: Proceedings of the 11th National Conference on Information Security, TaiNan, Taiwan, May 2001, pp.229–231.

[15] S.-M. Yen, M. Joye, "Improved authenticated multiple-key agreement protocol," Electronics Letters 1998;34 (18):1738–1739

[16] T.-S. Wu, W.-H. He, C.-L. Hsu, "Security of authenticated multiple-key," Electronics Letters 35 (5) (1999) 391–392.

[17] H.-S. Zhou, L. Fan and J.-H. Li, "Remarks on unknown key-share attack on authenticated multiple-key agreement protocol," In Electronics Letters 2003; 39 (17):1248–1249.