Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 13, Number 7 (2017), pp. 2951–2964
© Research India Publications
http://www.ripublication.com/gjpam.htm

Password Based Group Key Exchange Protocol via Twin Diffie Hellman Problem

Shruti Nathani and B.P. Tripathi
Department of Mathematics,
Govt. N.P.G. College of Science,
Raipur (C.G.), India.

Abstract

In 2012, Yuan et al. proposed a novel and efficient password based group key exchange protocol that mutually trusted on key generation center (KGC) using secret sharing technology. In 2013, R.F. Olimid mounted an insider attack against Yuan et al. protocol. In this paper we improve Yuan's password based group key exchange protocol by using twin Diffie Hellman problem. We also give security analysis of the proposed scheme and show that R.F. Olimid's insider attack against Yuan et al. is not applicable in our proposed scheme.

AMS subject classification: 94A60.
Keywords: Group key exchange, twin Diffie Hellman problem, password, KGC.

1. Introduction

With the development of computer and network technologies, network communications have become a part for many people's daily lives. To communicate over an open channel secretly, all the participants need to share a same knowledge to differ from the outsiders such as common session key and other secrets.To prevent an adversary from gaining access to the sensitive content of communications, a session key can be used for encryption/decryption. Therefore before exchanging communications message, a key exchange protocol must be used to construct the session keys for legitmate participants in the communication [2].

The key exchange protocol should guarantee the common session key computed by all participants is same and no person outside these participants can get this key even

in the conditions that all messages between these participants are eavesdropped by the outsiders. Furthermore, if all the messages between the participants are intercepted and replaced, each participator can detect it and drop this session key [17].

As we know that, the first practical key exchange protocol is introduced by Diffie and Hellman [18] in 1976, is the most commonly used protocol for constructing a common session key between two parties and its security is based on the discrete logarithm problem.This protocol is simple and efficient. The information encrypted by the public key can only be decrypted by the private key. The eavesdropper can not get any information about the private key, because it is never transmitted on the network. But the private key x is a big number and impossible to be remembered by normal person. Then many efforts [13, 16, 14] have been made to construct more user friendly schemes, such as password-based protocols.

The password based key exchange protocols require users only to remember a human memorable low entropy password and hope to provide the comparative intensity with the public keys system by internal cryptogrammic operations.The most password based key exchange protocols originated from the basic DH (Diffie-Hellman) protocol and its security basis discrete logarithm.

The DH key agreement protocol is not suitable for group communication, such as an e-conference, e-learning and multiuser games, which has more than two participants. Therefore a group key establishment protocol is needed for group communications. Even though group key agreement protocols have more flexibility to generate the group key, these protocols usually have heavy communication cost to construct the group key [2].

In 1996, Steiner et al. [12] proposed a key agreement protocol based on the natural extension of the DH key agreement protocol for the group communication. In 2001, Steiner's protocol has been enhanced with the property of authentication by Bresson et al. [4]. In 2006, Bohli [6] proposed a framework for robust group key agreement that provides security against malicious insiders and active adversaries in public point-to-point network. However, most DH based group key agreement protocols do not scale well and, in particular, require $O(n)$ rounds. In 2007, Katz and Yung [7] proposed the first constant-round and fully scalable group key exchange protocol to reduce the message transmission overhead.

There are other group key establishment protocols based on non-DH key agreement approach. In 2002, Tzeng [19] proposed a secure group key agreement protocol with fault-tolerant. With this ability, the protocol can detect malicious participants and prevent them from joining the group communications. However, each participant needs to maintain n-degree polynomials, where the parameter n depends on the number of participants. In fact, this is a serious problem to system overhead. In 2007, Tseng [21] demonstrated that Tzeng's protocol does not provide forward secrecy and then proposed a secure group key agreement protocol based on the decisional Diffie-Hellman (DDH) problem. In 2009, Huang et al. [10] proposed a group key protocol based on DLP to enhance the performance of Tzeng's scheme. In 2010, Zhao et al. [9] proposed a similar protocol based on RSA cryptosystem and also claim that their scheme can achieve the ability of fault-tolerant.Secret sharing schemes have been used to establish the group key

in recent years. In 2010, Harn and Lin [11] proposed a secure group key transfer protocol based on Shamir's (t, n) secret sharing [1], denoted as $(t, n) - SS$. They also modified Shamir's $(t, n) - SS$ as modulus N, a composite integer, to withstand the insider attack. In 2011, Nam et al.'s [8] proved that Harn Lin [11] protocol is vulnerable to reply attack. In 2012, Yuan et al.'s [17] apply the secret sharing technology to key exchange in combination with the traditional password based key exchange schemes. In 2013, R.F. Olimid [15], proved that, Nam et al.'s [8] reply attack against Harn Lin's [11] protocol may also apply to Yuan et al.'s [17] protocol and also R.F. Olimid [15] introduces the insider attack that remains valid evenif the protocol stands against the reply attack.

In 2009, Cash, Kiltz and Shoup [3] proposed a new computational problem called twin Diffie-Hellman problem which has the following interesting properties:

- The twin Diffie-Hellman problem can easily be employed in many cryptographic constructions where one would usually use the ordinary Diffie-Hellman problem, without imposing a terrible efficiency penalty.
- The twin Diffie-Hellman problem is hard, even given access to a corresponding decision oracle, assuming the ordinary Diffie-Hellman problem (without access to any oracles is hard).

Thus in 2013, motivated by Cash et al.'s [3], Pathak and Sanghi [5] proposed two simple three party password authenticated secure key exchange protocol based on twin Diffie-Hellman problem.

In this paper we can also use the concept of Cash et al.'s [3], twin Diffie-Hellman problem and propose a password authenticated group key exchange protocol via twin Diffie-Hellman problem. We first explain Yuan et al.'s [17] password based group key exchange protocol and R.F. Olimid's [15] insider attack against yuan et al.'s [17] protocol then we give our proposed protocol. At last, we give security analysis of the proposed scheme also show that the proposed protocol is also resistent to R.F. Olimid's insider attack.

2. Preliminaries

In this section, we give some fundamental backgrounds.

2.1. Factoring Problem [11]

Let us choose two large safe primes p and q (i.e. primes such that $p' = \frac{p-1}{2}$ and $q' = \frac{q-1}{2}$ are also primes) and compute $n = pq.n$ is made publicly known. Factoring problem is defined to compute factors p and q such that $n = pq$.

2.2. Factoring Assumption [11]

It is computationally intractable to solve the Factoring Problem.

Table 1: Characteristics of some group key exchange schemes

Schemes	Steiner et al.'s (1996)[12]	Bresson et al.'s (2001)[4]	Tzeng's (2002)[19]	Bohil's (2006)[6]	Katz Yung's (2007)[7]	Tseng's (2007)[21]	Huang et al.'s (2009)[10]	Zhao et al.'s (2009)[9]	Harn Lin's (2010)[11]	Yuan et al.'s (2012)[17]
Security principle	CDH assumption	DDH+ CDH assumptions	DLP+ polynomial interpolation	Sigature +Session identifiers	DDH assumption	DDH - assumption	DLP	Factorization problem	Shamir's(t,n) SS+factorization problem	Shamir's(t,n) SS+factorization problem
Advantages	Natural extensions of DH key agreement for n-Party	Authenticated key exchange /Mutual authentication	Without an online KGC /Fault-tolerace	Prevent insider and outsider attacks	Constant-round and fully scable	without an online KGC /Fault-tolerace	without an online KGC /Fault-tolerace	without an online KGC	Efficient	Efficient password based GKE
Drawbacks	High computation and communication costs	High communication cost	High system'sover-head/ Do not provide forward secrecy	Session identifires must be preshared	High computation cost	High computation and communication costs	High communication cost	High communication cost/ can not exclude adversaries completely	The online KGC is required/secrets must be preshared	The online KGC is required/password must be preshared/ vulnerable to insider attack

2.3. Twin DH Problem:

Cash, Kiltz and Shoup [3] suggested a new computational problem called twin Diffie-Hellman (DH) problem which is closely related to the usual (computational) DH problem and can be used in many of the same cryptographic constructions that are based on the DH problem. Moreover, the twin DH problem is atleast as hard as the ordinary DH problem [5].

Definition 2.1. Let G be a cyclic group of prime order q with a generator g. Define $dh(X, Y) := Z$ where $X = g^x$, $Y = g^y$ and $Z = g^{xy}$.

Given random $X, Y \in G$, the problem of computing $dh(X, Y)$ is the DH problem. The DH assumption asserts that it is hard to compute $dh(X, Y)$ with random choice $X, Y \in G$.

Define $2dh : G^3 \rightarrow 'G^2$. $(X_1, X_2, Y) \rightarrow (dh(X_1, Y), dh(X_2, Y))$, which is called the twin DH function. The twin DH assumption states that it is hard to compute 2dh (X_1, X_2, Y), given random $(X_1, X_2, Y) \in G$.[5]

2.4. Shamir's Secret Sharing

A secret sharing scheme is a method to split a secret into multiple shares that are distributed to participants via secured channels. The secret can be recovered only when the members of an authorized set of users combine their shares together [15].

Yuan et al. [17] base their protocol on Shamir''s secret scheme [1], which we describe next.

Let m be the number of users, $U_1, \ldots, U_m$ the users, S the secret to be shared and q a large prime number $(q > mandq > S)$. A dealer:

1. chooses m distinct and public elements $x_1, x_2, \ldots, x_m \in \mathbb{Z}_q$ ' for the participants $(x_i for U_i, i = 1, \ldots, m)$.

2. picks a $t - 1$ degree random polynomial

$$f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1} (mod q)$$

   such that $a_0 = S$ and $a_i \in \mathbb{Z}_q, i = 1, \ldots, t - 1$.

3. transmits the share $f(x_i)$ to the participant $U_i, i = 1, \ldots, m$, via a secure channel.

The reconstruction is based on polynomial interpolation: given at least $t$points $(x_i, f(x_i))$ with distinct $x_i's$, the unique polynomial $f(x)$ can be found as:

$$f(x) = \sum_{i=1}^{t} f(x_i) \prod_{1 \le j \le t, i \ne j} \frac{x - x_j}{x_i - x_j}$$

The secret S is evaluated as $f(0)$. Notice that any t or more participants can recover the secret, but less than t users obtain no information [15].

3. Yuan et al.'s Protocol [17]

Yuan et al. recently introduced a password-based GKT protocol [17] that uses Shamir's secret sharing scheme [1]. The protocol uses the following notations: m the number of possible users, $\{U_1, \ldots, U_m\}$ the set of all users, $\{U_1, \ldots, U_t\}$ the set of participants to a given session (after reordering), h_1 and h_2 two collision-resistant hash functions, $'\leftarrow^R$ a random choice from a specified set of values, $\|$ the string concatenation.

Initialization
The KGC selects two large primes p and q and computes $n = pq$;

Users Registration
Each user U_i, $i = 1, \ldots, m$, shares a long-term secret password $pw_i = pw_{ix} \| pw_{iy}$ with the KGC.
Round 1
User U_1:

1.1 chooses $k_1 \leftarrow^R \mathbb{Z}_n$;

1.2 computes $K_1 = pw_{1x} + k_1$ and $M_1 = h_1(U_1, \ldots, U_t, k_1)$;

1.3 sends a key generation request:
$U_1 \rightarrow KGC : (U_1, U_1, \ldots, U_t, K_1, M_1)$.

Round 2
The KGC:

2.1 computes $k_1 = K_1 - pw_{1x}$;

2.2 checks if $M_1 = h_1(U_1, \ldots, U_t, k_1)$;
if the equality does not hold, he quits;

2.3 broadcasts:
$KGC \rightarrow^*: \{U_1, \ldots, U_t\}$.

Round 3
Each User U_i, $i = 2, \ldots, t$:

3.1 chooses $k_i \leftarrow^R \mathbb{Z}_n$;

3.2 computes $K_i = pw_{ix} + k_i$ and $M_i = h_1(U_1, \ldots, U_t, k_i)$;

3.3 sends:
$U_i \rightarrow KGC : (U_i, \{U_1, \ldots, U_t\}, K_i, M_i)$.

Round 4
The KGC:

4.1 computes $k_i = K_i - pw_{ix}, i = 2, 3, \ldots, t$;

4.2 checks if $M_i = h_1(U_1, \ldots, U_t, k_i), i = 2, \ldots, t$;
if atleast one equality does not hold, he quits;

4.3 selects two random numbers x_{ta} and y_{ta} of lengths equal to pw_{ix} and pw_{iy};

4.4 generates the polynomial $f(x)$ of degree t that passes through $t + 1$ points $(x_{ta}, y_{ta}), (pw_{1x}, pw_{1y} + k_1), \ldots, (pw_{tx}, pw_{ty} + k_t)$;

4.5 Computes t additional points $P_1, \ldots, P_t$ of $f(x)$;

4.6 Computes the verification messages $V_i = h_2(U_1, \ldots, U_t, P_1, \ldots, P_t, k_i), i = 1, 2, \ldots, t$;

4.7 sends,$i = 1, \ldots, t$:
$KGC \rightarrow: U_i : (P_1, P_2, \ldots, P_t, V_i)$.

Key Computation
Each user $U_i, i = 1, \ldots, t$:

5.1 checks if $V_i = h_2(U_1, \ldots, U_t, P_1, \ldots, P_t, k_i)$;
if the equality does not hold, he quits;

5.2 computes the group key $f(0)$ by interpolating the points $P_1, P_2, \ldots, P_t$ and $(pw_{ix}, pw_{iy} + k_i)$.

4. Insider Attack Against Yuan et al. protocol proposed By R.F. Olimid [15]

Let $U_a, 1 \leq a \leq m$ be an attacker whose goal is to reveal the long-term secret password of a user $U_i, 1 \leq i \leq m, i \neq a$. This knowledge further permits him to compute all session keys U_i is qualified for. In this attack U_a behaves honestly and does not impersonate U_i. The adversary initiates two sessions (s_1) and (s_2) of the protocol, requesting to share a key with U_i. So, he is qualified to recover the coefficients of the polynomials $f(x)_{(s_1)}$ and $f(x)_{(s_2)}$ by interpolation (Key Computation Phase):

$$\begin{aligned} f(x)_{(s_1)} &= a_{(s_1)}x^2 + b_{(s_1)}x + c_{(s_1)} \\ f(x)_{(s_2)} &= a_{(s_2)}x^2 + b_{(s_2)}x + c_{(s_2)} \end{aligned} \tag{4.1}$$

Since $(pw_{ix}, pw_{iy} + k_{i_{(s_1)}})$ and $(pw_{ix}, pw_{iy} + k_{i_{(s_2)}})$ are valid points of $f(x)_{(s_1)}$ and respectively $f(x)_{(s_2)}$, Equations (4.1) become

$$\begin{aligned} pw_{iy} + k_{i_{(s_1)}} &= a_{(s_1)}pw_{ix}^2 + b_{(s_1)}pw_{ix} + c_{(s_1)} \\ pw_{iy} + k_{i_{(s_2)}} &= a_{(s_2)}pw_{ix}^2 + b_{(s_2)}pw_{ix} + c_{(s_2)} \end{aligned} \tag{4.2}$$

We emphasize that the attacker does not know $k_{i_{(s_1)}}$ and $k_{i_{(s_2)}}$, but he may eavesdrop on $k_{i_{(s_1)}}$ and $k_{i_{(s_2)}}$ (step 3.3). The following hold from the definition of K_i (step 3.2)

$$\begin{aligned} pw_{iy} + k_{i_{(s_1)}} &= pw_{iy} + K_{i_{(s_1)}} - pw_{ix} \\ pw_{iy} + k_{i_{(s_2)}} &= pw_{iy} + K_{i_{(s_2)}} - pw_{ix} \end{aligned} \tag{4.3}$$

Replacing Equations (4.3) in Equations (4.2) leads to:

$$\begin{aligned} pw_{iy} &= a_{(s_1)} pw_{ix}^2 + (b_{(s_1)} + 1) pw_{ix} + c_{(s_1)} - K_{i_{(s_1)}} \\ pw_{iy} &= a_{(s_2)} pw_{ix}^2 + (b_{(s_2)} + 1) pw_{ix} + c_{(s_2)} - K_{i_{(s_2)}} \end{aligned} \tag{4.4}$$

We eliminate pw_{iy} from Equations (4.4) and introduce the following notations:

$$\begin{aligned} A_{(s_1 s_2)} &= a_{(s_1)} - a_{(s_2)} \\ B_{(s_1 s_2)} &= b_{(s_1)} - b_{(s_2)} \\ C_{(s_1 s_2)} &= c_{(s_1)} - c_{(s_2)} - (K_{i_{(s_1)}} - K_{i_{(s_2)}}) \end{aligned} \tag{4.5}$$

It follows that:

$$A_{(s_1 s_2)} pw_{ix}^2 + B_{(s_1 s_2)} pw_{ix} + C_{(s_1 s_2)} = 0 \tag{4.6}$$

The adversary follows the same strategy as before for two other sessions (s_3) and (s_4) to acquire:

$$A_{(s_3 s_4)} pw_{ix}^2 + B_{(s_3 s_4)} pw_{ix} + C_{(s_3 s_4)} = 0 \tag{4.7}$$

The attacker now finds pw_{ix} as the solution of the equation system formed by Equation (4.6) and Equation (4.7):

$$pw_{ix} = \frac{A_{(s_1 s_2)} C_{(s_3 s_4)} - A_{(s_3 s_4)} C_{(s_1 s_2)}}{A_{(s_3 s_4)} B_{(s_1 s_2)} - A_{(s_1 s_2)} B_{(s_3 s_4)}} (mod n) \tag{4.8}$$

once U_a obtains pw_{ix} he computes pw_{iy} as:

$$pw_{iy} = f(pw_{ix})_{(s_j)} - K_{i_{(s_j)}} + pw_{ix} \tag{4.9}$$

for any $j = 1, \ldots, 4$. In conclusion, the attacker achieves his goal: he exposes the long-term secret password of the user U_i.

5. Proposed Protocol

Initialization The KGC selects two large primes p and q and computes $n = pq$.

User Registration: Each user $U_i, i = 1, 2, \ldots, m$ partially shares a long term secret password pw_i with KGC.
$pw_i \rightarrow$ password of each user i, partially shared with S.
Let pw_i be the password partially shared between user i and server S, which is a arbitrary

bit string.
Here each user stores the password (pw_{ix}, pw_{iy}). Each user shares pw_{ix} with the server S and kept secret pw_{iy}. That is each user partially share the password pw_i.
In the proposed protocol, let us use a_i in place of pw_{iy}, which is kept secret by each user.

Round 1
User U_1:

1.1 chooses $k_1 \leftarrow^R \mathbb{Z}_n$;
Then $v_1 = {k_1}^{a_1}$.

1.2 computes $K_1 = pw_{1x} + v_1$ and $M_1 = h_1(U_1, \ldots, U_t, v_1)$;

1.3 sends a key generation request:
$U_1 \rightarrow KGC : (U_1, U_1, \ldots, U_t, K_1, M_1)$.

Round 2
The KGC:

2.1 computes $v_1 = K_1 - pw_{1x}$;

2.2 checks if $M_1 = h_1(U_1, \ldots, U_t, v_1)$;
if the equality does not hold, he quits;

2.3 broadcasts:
$KGC \rightarrow^*: \{U_1, \ldots, U_t\}$.

Round 3
Each User $U_i, i = 2, \ldots, t$:

3.1 chooses $k_i \leftarrow^R \mathbb{Z}_n$; Then $v_i = {k_i}^{a_i}$

3.2 computes $K_i = pw_{ix} + v_i$ and $M_i = h_1(U_1, \ldots, U_t, v_i)$;

3.3 sends:
$U_i \rightarrow KGC : (U_i, \{U_1, \ldots, U_t\}, K_i, M_i)$.

Round 4
The KGC:

4.1 computes $v_i = K_i - pw_{ix}, i = 2, 3, \ldots, t$;

4.2 checks if $M_i = h_1(U_1, \ldots, U_t, v_i), i = 2, \ldots, t$;
if atleast one equality does not hold, he quits;

4.3 selects two random numbers x_{ta} and y_{ta}, both of lengths equal to pw_{ix};

4.4 generates the polynomial $f(x)$ of degree t that passes through $t + 1$ points $(x_{ta}, y_{ta}), (pw_{1x}, v_1), \ldots, (pw_{tx}, v_t)$;

4.5 Computes t additional points $P_1, \ldots, P_t$ of $f(x)$;

4.6 Computes the verification messages $V_i = h_2(U_1, \ldots, U_t, P_1, \ldots, P_t, v_i), i = 1, 2, \ldots, t$;

4.7 sends, $i = 1, \ldots, t$:
$KGC \rightarrow: U_i : (P_1, P_2, \ldots, P_t, V_i)$.

Key Computation
Each user $U_i, i = 1, \ldots, t$:

5.1 checks if $V_i = h_2(U_1, \ldots, U_t, P_1, \ldots, P_t, v_i)$;
if the equality does not hold, he quits;

5.2 computes the group key $f(0)$ by interpolating the points $P_1, P_2, \ldots, P_t$ and (pw_{ix}, v_i).

6. Security analysis

The security analysis of the proposed protocol mainly relies o the difficulty of twin Diffie Hellman problem. In our proposed protocol the password of the clients are only partially shared with the KGC. This prevents most of the attacks.

Theorem 6.1. The proposed protocol is secured from most of the following attacks:

(a) Trivial attack

(b) On-line Password Guessing Attack

(c) Off-line Password Guessing Attack

(d) Perfect Forward Secrecy.

Proof.

(a) **Trivial Attack:** An attacker may directly try to compute the passwords and/or the group key $f(0)$ from the transmitted message $[U_i \rightarrow KGC : (U_i, \{U_1, \ldots, U_t\}, K_i, M_i), i = 1, \ldots, t]$ But due to difficulties of the discrete logarithm problem, twin Diffie Hellman problem and one way ness of hash function, the trivial attack is not possible in the proposed protocol.

(b) **Online Password Guessing Attack:** In on-line guessing attack an attacker tries to confirm a guessed password in an on-line transaction.In our proposed protocol the password of each clients are only partially and not completely shared with the server (KGC). The part pw_{iy} of the password pw_i are kept secret by the user U_i and is not transmitted through any message.Even if an attacker or a malicious user B tries to guess $U_1{}'s$ password then he can only guess b_1, and

choose any random $k_1^{'} \in^R \mathbb{Z}_n$ and get $\nu_1{}^{'} = (k_1{}^{'})^{b_1}$ and send it in online transaction. But to verify the correctness of his guessed password he has to compute $M_1 = h_1(U_1, \ldots, U_t, \nu_1) = h_1(U_1, \ldots, U_t, k_1{}^{a_1})$,which is impossible since he requires the value of $k_1{}^{a_1}$. Therefore, Online guessing attack is not possible in the proposed protocol.

(c) **Off-line Password Guessing Attack:**Assume that an attacker tries to mount off-line password guessing attack to guess the password. Suppose he intercepts the message M_1 by $M_1{}^{'}$ of user U_1, but still he can not verify his guessed password due to the difficulty of getting the value of a_1,which is kept secret by user U_1 and one-way ness of hash function. Hence off-line password guessing attack is impossible in the proposed scheme.

(d) **Perfect Forward Secrecy:** In our proposed scheme each user $U_i, i = 1, \ldots, m$ partially shared a long term password ($pw_i = pw_{ix} \parallel pw_{iy}$) with the KGC. Each user shares pw_{ix} with the KGC and kept secret the part pw_{iy} of the password pw_i. Even if the passwords pw_{ix} of each user U_i are compromised, the attacker can not calculate the group key as the part pw_{iy} is unknown.This value remains unknown even to the KGC (server) and so there is no chance of any compromise. ■

Theorem 6.2. (Outsider attack) Assume that an attacker who impersonate a group member for requesting a group key service, than the attacker can neither obtain the group key nor share a group key with any group member.

Proof. Although any attacker can impersonate a group member to issue a service request to KGC without being detected and KGC will respond by sending group key information accordingly ;however the group key can only be recovered by any group member who shares a secret with KGC. In our proposed protocol each member partially shared his/her password pw_i with KGC. This security feature is information theoretically secure.

If the attacker tries to reuse a compromised group key by replying previously recorded key information from KGC, this attack can not succeed in sharing this compromised group key with any group member because in the proposed scheme each group member partially shared his/her password ($pw_i = pw_{ix} \parallel pw_{iy}$) that means pw_{iy} shared with KGC and kept secret pw_{iy} and hence the compromised group key can not be reused since group key $f(0)$ is a function of shared secret between KGC and each member, pw_{ix} and ν_i, where $\nu_i = k_i{}^{a_i}$, and the computation of ν_i needs a_i. We can use a_i to denote pw_{iy}, the second part of the password which is kept secret by each user U_i. Thus the outsider attack is not possible in the proposed scheme. ■

Insider attack: In 2013, R. F. OLIMID [15] gave a insider attack against Yuan et al.'s [17] protocol which we explain in section (4). In this section we can show that this type of insider attack is also not possible in our proposed protocol.

Let $U_a, 1 \leq a \leq m$ be an attacker whose goal is to reveal the long-term secret password of a user $U_i, 1 \leq i \leq m, i \neq a$. This knowledge further permits him to compute all session keys U_i is qualified for.

The adversary initiates two sessions (s_1) and (s_2) of the protocol, requesting to share a key with U_i. So, he is qualified to recover the coefficients of the polynomials $f(x)_{(s_1)}$ and $f(x)_{(s_2)}$ by interpolation (Key Computation Phase):

$$\begin{aligned} f(x)_{(s_1)} &= a_{(s_1)}x^2 + b_{(s_1)}x + c_{(s_1)} \\ f(x)_{(s_2)} &= a_{(s_2)}x^2 + b_{(s_2)}x + c_{(s_2)} \end{aligned} \tag{6.1}$$

Since $(pw_{ix}, v_{i_{(s_1)}})$ and $(pw_{ix}, v_{i_{(s_2)}})$ are valid points of $f(x)_{(s_1)}$ and respectively $f(x)_{(s_2)}$, Equations (6.1) become

$$\begin{aligned} v_{i_{(s_1)}} &= a_{(s_1)}pw_{ix}^2 + b_{(s_1)}pw_{ix} + c_{(s_1)} \\ v_{i_{(s_2)}} &= a_{(s_2)}pw_{ix}^2 + b_{(s_2)}pw_{ix} + c_{(s_2)} \end{aligned} \tag{6.2}$$

We emphasize that the attacker does not know $v_{i_{(s_1)}}$ and $v_{i_{(s_2)}}$, but he may eavesdrop on $v_{i_{(s_1)}}$ and $v_{i_{(s_2)}}$ (step 3.3). The following hold from the definition of K_i (step 3.2)

$$\begin{aligned} v_{i_{(s_1)}} &= K_{i_{(s_1)}} - pw_{ix} \\ v_{i_{(s_2)}} &= K_{i_{(s_2)}} - pw_{ix} \end{aligned} \tag{6.3}$$

Replacing Equations (6.3) in Equations (6.2) leads to:

$$\begin{aligned} a_{(s_1)}pw_{ix}^2 + (b_{(s_1)} + 1)pw_{ix} + c_{(s_1)} - K_{i_{(s_1)}} = 0 \\ a_{(s_2)}pw_{ix}^2 + (b_{(s_2)} + 1)pw_{ix} + c_{(s_2)} - K_{i_{(s_2)}} = 0 \end{aligned} \tag{6.4}$$

Then we introduce the following notations:

$$\begin{aligned} A_{(s_1s_2)} &= a_{(s_1)} - a_{(s_2)} \\ B_{(s_1s_2)} &= b_{(s_1)} - b_{(s_2)} \\ C_{(s_1s_2)} &= c_{(s_1)} - c_{(s_2)} - (K_{i_{(s_1)}} - K_{i_{(s_2)}}) \end{aligned} \tag{6.5}$$

It follows that:

$$A_{(s_1s_2)}pw_{ix}^2 + B_{(s_1s_2)}pw_{ix} + C_{(s_1s_2)} = 0 \tag{6.6}$$

The adversary follows the same strategy as before for two other sessions (s_3) and (s_4) to acquire:

$$A_{(s_3s_4)}pw_{ix}^2 + B_{(s_3s_4)}pw_{ix} + C_{(s_3s_4)} = 0 \tag{6.7}$$

The attacker now finds pw_{ix} as the solution of the equation system formed by Equation (6.6) and (6.7):

$$pw_{ix} = \frac{A_{(s_1s_2)}C_{(s_3s_4)} - A_{(s_3s_4)}C_{(s_1s_2)}}{A_{(s_3s_4)}B_{(s_1s_2)} - A_{(s_1s_2)}B_{(s_3s_4)}}(mod n) \tag{6.8}$$

Once U_a obtains pw_{ix} he computes pw_{iy}, as in insider attack proposed by R.F. Olimid, attacker puts the value of pw_{ix} in equation (6.4) to find pw_{iy}, but there is no relation between pw_{ix} and pw_{iy},because in our proposed scheme each user U_i partially shared his password pw_i,so there are no possibility to find pw_{iy} from pw_{ix}.

Thus the insider attack proposed by R.F. Olimid [15], is also not succeed in our proposed scheme.

7. Conclusion

Although so many password based key exchange protocols have been developed. Most of them are vulnerable to various attacks. With the increasing need for authentication and secure communication a password based group key exchange protocol via twin Diffie Hellman problem is introduced which resist to all the known attacks.

References

[1] A. Shamir, "How to share a secret", Commun. ACM, vol. 22, no. 11, pp. 612–613, Nov. (1979).

[2] C.Y. Lee, Z.H. Wang, L. Harn and C.C. Chang, "Secure key transfer protocol basedon secret sharing for group communications", IEICE Trans. inf. and syst. vol. E94-D, No. 11, November (2011).

[3] D. Cash, E. Kiltz and V. Shoup, "The twin Diffie-Hellman problem and applications", Journal of cryptology, vol. 22, pp. 470–504, (2009).

[4] E. Bresson, O. Chevassut, D. Pointcheval, and J.J. Quisquater, "Provably authenticated group Diffie-Hellman key exchange", Proc. 8th ACM Conference on Computer and Communications Security (CCS '01), pp. 255–264, Philadelphia, USA, Nov. (2001).

[5] H.K. Pathak and Maju Sanghi, "Simple Three Party Key Exchange Protocols via Twin Diffie Hellman Problem", Interational Journal of Network Security, vol. 15, No. 4., pp. 256–264, July (2009).

[6] J.M. Bohli, "A framework for robust group key agreement", Proc. International Conference on Computational Science and Applications (ICCSA '06), pp. 355–364, Glasgow, UK, May (2006).

[7] J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange", J. Cryptology, vol. 20, no. 1, pp. 85–113, Jan. (2007).

[8] J. Nam, M. Kim, J. Paik, W. Jeon, B. Lee, D. Won, "Cryptanalysis of a Group Key Transfer Protocol based on Secret Sharing", In: Proceedings of the Third international conference on Future Generation Information Technology, pp. 309–315. FGIT'11, Springer-Verlag, Berlin, Heidelberg, (2011).

[9] J. Zhao, D. Gu and Y. Li, "An efficient Fault-tolerant group key agreement protocol", Compu. Commun., vol. 33, no. 7, pp. 890–895, May (2010).

[10] K.H. Huang, Y.F. Chung, H.H. Lee, F. Lai and T.S. Chen, "A Conference Key agreement protocol with fault-tolerant capability", Computer Standards and Interfaces, vol. 31, no. 2, pp. 401–405, Jan (2009).

[11] L. Harn, C. Lin, "Authenticated Group Key Transfer Protocol based on Secret Sharing", IEEE Trans. Comput. 59(6), 842–846 (2010).

[12] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman key distribution extended to group communication", Proc. 3rd ACM Conference on Computer and Communications Security (CCS '96), pp. 31–37, March (1996).

[13] M.K. Boyarsky, ACMCCS 99:6th Conference on Computer and Communications Security, (1999), pp. 63–72.

[14] M. Bellare, D. Pointcheval, and P. Rogaway, "Advances in Cryptology-EUROCRYPT 2000", Lecture Notes in Computer Science 1807, (2000), pp. 139–155.

[15] R.F. Olimid, "Cryptanalysis of a password based group key exchange protocol using secret sharing", Appl. Math. Inf. Sci. 7, No. 4, 1585–1590 (2013).

[16] S. M. Bellovin and M. Merritt, IEEE Symposium on Security and Privacy, (1992), pp. 72–84.

[17] W. Yuan, L. Hu, H. Li, J. Chu, "An Efficient Password-based Group Key Exchange Protocol Using Secret Sharing", Appl. Math. Inf. Sci. 7(1), 145–150 (2012).

[18] W. Diffie and M. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, vol. 22, pp. 644–654, (1976).

[19] W.G. Tzeng, "A secure fault-tolerant conference key agreement protocol", IEEE Trans., Comput, vol. 51, no. 4, pp. 373–379, April (2002).

[20] Y. Sun, Q. wen, H. Sun, W. Li, Z. Jin, H. Zhang, "An Authenticated Group Key Transfer Protocol Based On Secret Sharing", Proceeding Engineering 29(0), 403–408 (2012).

[21] Y.M. Tseng, "A communication-efficient and fault-tolerant conference-key agreement protocol with forward secrecy", J. Syst. Softw., vol. 80, no. 7, pp. 1091–1101, July (2007).