Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/070444

Cryptanalysis of a Password-based Group Key Exchange Protocol Using Secret Sharing

Ruxandra F. Olimid[*]

Department of Computer Science, University of Bucharest, Romania

Received: 22 Jan. 2013, Revised: 20 Mar. 2013, Accepted: 22 Mar. 2013
Published online: 1 Jul. 2013

Abstract: Yuan et al. recently introduced a password-based group key transfer protocol that uses secret sharing, which they claim to be efficient and secure [9]. We remark its resemblance to the construction of Harn and Lin [1], which Nam et al. proved vulnerable to a replay attack [3]. It is straightforward that the same attack can be mount against Yuan et al.'s protocol, proving that the authors' claim is false. In the same paper, Nam et al. propose a countermeasure that may also apply to Yuan et al.'s protocol. However, we show that their protocol remains susceptible to an insider attack (even if it stands against the replay attack): any malicious participant can recover the long-term secret password of any other user and therefore becomes able to compute group keys he is unauthorized to know.

Keywords: group key transfer, secret sharing, insider attack, replay attack, cryptanalysis

1 Introduction

In order to benefit of secure group-oriented applications (such as group video conferences or group text communication), multiple users need to share a common secret cryptographic key, which is obtained as the output of a *group key establishment protocol* (GKE).

GKE protocols divide into two classes: *group key transfer* (GKT) - a privileged party called Key Generation Center (KGC) selects a key and securely distributes it to the qualified users - and *group key agreement* (GKA) - all parties collaborate to compute a common secret key.

Yuan et al. recently introduced a GKT protocol based on secret sharing [9]. A *secret sharing scheme* splits a secret into multiple shares such that only authorized set of shares may lead to the reconstruction of the secret. They are widely used for constructing GKT protocols due to the advantages they offer [5]. Such examples include the protocols of Pieprzyk and Li [5], Sáez [6], Harn and Lin [1], Hsu et al. [2] and Sun et al. [8].

Many recent papers that describe new GKT protocols lack a formal security proof, which leads to vulnerabilities. This is the case of Harn and Li's protocol that was proved susceptible to replay attacks [3] or Sun et al.'s protocol that was proved susceptible to insiders attacks and known key attacks [4].

This work enriches the list by introducing an attack on Yuan et al.'s protocol. We first remark its resemblance to the Harn and Lin's protocol [1] and highlight that it preserves the same vulnerability to replay attacks. More, we show that even if the construction is improved to stand against this kind of attacks, Yuan et al.'s protocol is susceptible to insider attacks: a malicious participant can recover group keys he is unqualified to know. Therefore, we prove that the construction is totally insecure.

The paper is organized as follows. The next section contains the preliminaries. Section 3 describes Yuan et al.'s protocol and its striking resemblance to the protocol that Harn and Lin had introduced three years before. We also notice that the replay attack mounted against the initial protocol persists for Yuan et al.'s protocol. Section 4 introduces the insider attack that remains valid even if the protocol stands against the replay attack. Section 5 concludes.

2 Preliminaries

2.1 Key Confidentiality

Confidentiality represents the main security goal of GKE protocols. It assures that the group key is available to authorized participants only and no other party can

[*] Corresponding author e-mail: ruxandra.olimid@fmi.unibuc.ro

recover it, even if the protocol runs for several times (*sessions*).

Adversaries against key confidentiality are categorized into two types: *insiders* and *outsiders*. An insider is a genuine participant who may take part to the protocol, while an outsider is always unqualified. A protocol is susceptible to insider attacks if an insider becomes able to compute session keys he is unauthorized for. Similarly, it is vulnerable to outsider attacks if an outsider is capable to reveal any session key. The main advantage of an insider is that he can honestly initiate and take part to protocol sessions.

Section 4 will introduce an insider attack against Yuan et al.'s protocol, which permits a participant to find the long-term secrets of other users that he further uses to disclose session keys he is unauthorized to know.

2.2 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.

Yuan et al. base their protocol on Shamir's secret scheme [7], 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 ($q>m$ and $q>S$). 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

$$f(x)=a_0+a_1x+\cdots+a_{t-1}x^{t-1} \pmod q \quad (1)$$

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:

$$f(x)=\sum_{i=1}^{t} f(x_i) \prod_{1\le j\le t, i\ne j} \frac{x-x_j}{x_i-x_j} \quad (2)$$

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.

3 Yuan et al.'s Protocol

3.1 Protocol Description

Yuan et al. recently introduced a password-based GKT protocol [9] that uses Shamir's secret sharing scheme [7].

Figure 1 describes the protocol and 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.

3.2 Comparison to Harn and Lin's Protocol

Yuan et al.'s proposal looks remarkably similar to a protocol introduced three years before by Harn and Lin [1]. Figure 2 describes a version of the original protocol that achieves users authentication but ignores key confirmation.

An obvious difference between the two constructions consists in the long-term secrets that are shared between the users and the KGC: Yuan et al.'s protocol requires a password $pw_i = pw_{ix}||\ pw_{iy}$, while Harn and Lin's protocol needs a secret pair (x_i,y_i) (Users Registration Phase). The computational relation between the long-term secret and the random chosen numbers also differs: pw_{ix} tries to mask k_i (steps 1.2 and 3.2), while R_i is computational unrelated to x_i or y_i and sent in clear (step 3.3). We will show in Section 4 that the relation between pw_{ix}, pw_{iy} and k_i reveals a new attack.

A second difference is that in Yuan et al.'s protocol the values M_i used to authenticate the random numbers k_i depend on the group members of the current session (steps 1.2 and 3.2), while in Harn and Lin's protocol $Auth_i$ is independent of the current session group (step 3.2). We will consider this in the next subsection, when we will briefly analyze the applicability of Nam et al.'s attack against Yuan et al.'s protocol.

A last notable difference appears in Round 4: the KGC sends t messages in case of Yuan et al.'s protocol (step 4.7), while a single broadcast message is enough in case of Harn and Lin's protocol (step 4.6). The former construction is therefore less efficient than the latter in this stage.

3.3 Nam et al.'s Replay Attack

Nam et al. prove that Harn and Lin's protocol is susceptible to a replay attack [3]: an adversary U_a, $1\le a\le m$ can disclose the long-term secret of a user U_i, $1\le i\le m$, $i\ne a$. This gives the attacker the ability to obtain the key of any session U_i participates in, even if U_a is not a qualified party for the session.

We briefly describe the attack next. The adversary eavesdrop on a protocol session for which U_i is qualified and finds $(R_i,Auth_i)$ (step 3.3). Then he initiates two sessions (s_1) and (s_2), requesting to share a key with U_i. In both sessions he performs the same actions: he impersonate the qualified user U_i by sending the eavesdropped message $(R_i,Auth_i)$ and behaves honestly in sending his own message $(R_a,Auth_a)$.

Initialization
The KGC selects 2 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,...,t;
4.2. checks if $M_i = h_1(U_1,\ldots,U_t,k_i)$, i=2,...,t;
If at least one equality does not hold, he quits;
4.3. selects 2 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 the $t+1$ points (x_{ta},y_{ta}), $(pw_{1x}, pw_{1y}+k_1)$, ... , $(pw_{tx}, pw_{ty}+k_t)$;
4.5. computes t additional points $P_1,\ldots,P_t$ of $f(x)$;
4.6. computes the verification messages $V_i = h_2(U_1,\ldots,U_t,P_1,\ldots,P_t,k_i)$, $i = 1,\ldots,t$;
4.7. sends, $i = 1,\ldots,t$:
$KGC \rightarrow U_i : (P_1,\ldots,P_t,V_i)$

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 $P_1,\ldots,P_t$ and $(pw_{ix}, pw_{iy}+k_i)$.

Fig. 1: Yuan et al.'s Group Key Transfer Protocol [9]

KGC generates two polynomials $f(x)_{(s_1)}$ and $f(x)_{(s_2)}$, one for each session:

$$\begin{aligned} f(x)_{(s_1)} &= a_{(s_1)}x^2 + b_{(s_1)}x + c_{(s_1)} \\ f(x)_{(s_2)} &= a_{(s_2)}x^2 + b_{(s_2)}x + c_{(s_2)} \end{aligned} \quad (3)$$

We stress that U_a knows the coefficients of both $f(x)_{(s_1)}$ and $f(x)_{(s_2)}$, because he is authorized for both sessions.

Since $f(x_i)_{(s_1)} = f(x_i)_{(s_2)} = y_i \oplus R_i$ and $f(x_a)_{(s_1)} = f(x_a)_{(s_2)} = y_a \oplus R_a$, x_i and x_a are two roots of the quadratic equation:

$$(a_{(s_1)} - a_{(s_2)})x^2 + (b_{(s_1)} - b_{(s_2)})x + (c_{(s_1)} - c_{(s_2)}) = 0 \quad (4)$$

Initialization
The KGC selects 2 large safe primes p and q (i.e. $p' = \frac{p-1}{2}$ and $q' = \frac{q-1}{2}$ are also primes) and computes $n = pq$;

Users Registration
Each user $U_i, i = 1, \ldots, m$, shares a long-term secret $(x_i, y_i) \in \mathbb{Z}_n^* \times \mathbb{Z}_n^*$ with the KGC;

Round 1
User U_1:
1.1. sends a key generation request:
$U_1 \rightarrow KGC : \{U_1, \ldots, U_t\}$

Round 2
The KGC:
2.1. broadcasts:
$KGC \rightarrow^* : \{U_1, \ldots, U_t\}$

Round 3
Each user $U_i, i = 1, \ldots, t$:
3.1. chooses $R_i \leftarrow^R \mathbb{Z}_n^*$;
3.2. computes $Auth_i = h(x_i, y_i, R_i)$;
3.3. sends:
$U_i \rightarrow KGC : (R_i, Auth_i)$

Round 4
The KGC:
4.1. checks if $Auth_i = h(x_i, y_i, R_i)$, $i = 1, \ldots, t$;
If at least one equality does not hold, he quits;
4.2. selects a group key $k \leftarrow^R \mathbb{Z}_n^*$;
4.3. generates the polynomial $f(x)$ of degree t that passes through the $t+1$ points $(0,k)$,$(x_1, y_1 \oplus R_1)$, ..., $(x_t, y_t \oplus R_t)$;
4.4. computes t additional points $P_1, \ldots, P_t$ of $f(x)$;
4.5. computes the authentication message $Auth = h(k, U_1, \ldots, U_t, R_1, \ldots, R_t, P_1, \ldots, P_t)$;
4.6. broadcasts:
$KGC \rightarrow^* : (P_1, \ldots, P_t, Auth)$

Key Computation
Each user U_i, $i = 1, \ldots, t$:
5.1. computes the group key $k = f(0)$ by interpolating the points $P_1, \ldots, P_t$ and $(x_i, y_i \oplus R_i)$;
5.2. checks if $Auth = h(k, U_1, \ldots, U_t, R_1, \ldots, R_t, P_1, \ldots, P_t)$;
If the equality does not hold, he quits.

Fig. 2: Harn and Lin's Group Key Transfer Protocol [1]

The adversary U_a finds the value x_i as:

$$x_i = x_a^{-1}(a_{(s_1)} - a_{(s_2)})^{-1}(c_{(s_1)} - c_{(s_2)}) \quad (5)$$

Then, he computes y_i from $f(x_i)_{(s_j)} = y_i \oplus R_i$, for any $j = 1, 2$. In conclusion, the adversary succeeds his goal and reveals the long-term secret (x_i, y_i). For more details, the reader may refer to the original paper [3].

We remark that the attack remains available for Yuan et al.'s protocol also: the attacker eavesdrops on the message $(U_i, \{U_i, U_a\}, K_i, M_i)$ (step 3.3) and then use it to impersonate U_i to the KGC. We omit the details to avoid repetition. The applicability of the replay attack apparently narrows: a session between the adversary U_a and the user U_i must previously exist, otherwise the KGC quits in step 4.2 (the authorization string M_i depends on the current group members). However, since U_a is an insider, he may initiate the needed session by himself. Compared to the original protocol, the attack only requires an extra session.

In the same paper, Nam et al.'s propose a countermeasure for the replay attack [3]: the KGC broadcasts a random value $R_0 \in \mathbb{Z}_n^*$ along with the participants list $U_1, \ldots, U_t$ (step 2.1). Then, each party uses this value (together with $\{U_1, \ldots, U_t\}$) to compute the authorization string $Auth_i$ as $Auth_i = h(x_i, y_i,$

$R_i, R_0, \{U_1, \ldots, U_t\})$, $i = 1, \ldots, t$ (step 3.2). The same approach stands for Yuan et al.'s protocol: the KGC broadcasts a random k_0 along with the participants list (step 2.3) and the users (including the initiator U_1) compute the values M_i as $M_i = h_i(U_1, \ldots, U_t, k_i, k_0)$ (step 3.2). Again, we omit the details to avoid repetition.

We claim that even if Yuan et al.'s protocol is improved to avoid replay attacks, it still remains vulnerable to insider attacks. The next session introduces an example to support our affirmation.

4 Insider Attack

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. Unlike the replay attack proposed by Nam et al. and described in the previous section, in our 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):

$$\begin{aligned} f(x)_{(s_1)} &= a_{(s_1)}x^2 + b_{(s_1)}x + c_{(s_1)} \\ f(x)_{(s_2)} &= a_{(s_2)}x^2 + b_{(s_2)}x + c_{(s_2)} \end{aligned} \tag{6}$$

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 (6) become:

$$\begin{aligned} pw_{iy} + k_{i(s_1)} &= a_{(s_1)}pw_{ix}^2 + b_{(s_1)}pw_{ix} + c_{(s_1)} \\ pw_{iy} + k_{i(s_2)} &= a_{(s_2)}pw_{ix}^2 + b_{(s_2)}pw_{ix} + c_{(s_2)} \end{aligned} \tag{7}$$

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):

$$\begin{aligned} pw_{iy} + k_{i(s_1)} &= pw_{iy} + K_{i(s_1)} - pw_{ix} \\ pw_{iy} + k_{i(s_2)} &= pw_{iy} + K_{i(s_2)} - pw_{ix} \end{aligned} \tag{8}$$

Replacing Equations (8) in Equations (7) leads to:

$$\begin{aligned} pw_{iy} &= a_{(s_1)}pw_{ix}^2 + (b_{(s_1)} + 1)pw_{ix} + c_{(s_1)} - K_{i(s_1)} \\ pw_{iy} &= a_{(s_2)}pw_{ix}^2 + (b_{(s_2)} + 1)pw_{ix} + c_{(s_2)} - K_{i(s_2)} \end{aligned} \tag{9}$$

We eliminate pw_{iy} from Equations (9) and introduce the following notations:

$$\begin{aligned} A_{(s_1s_2)} &= a_{(s_1)} - a_{(s_2)} \\ B_{(s_1s_2)} &= b_{(s_1)} - b_{(s_2)} \\ C_{(s_1s_2)} &= c_{(s_1)} - c_{(s_2)} - (K_{i(s_1)} - K_{i(s_2)}) \end{aligned} \tag{10}$$

It follows that:

$$A_{(s_1s_2)}pw_{ix}^2 + B_{(s_1s_2)}pw_{ix} + C_{(s_1s_2)} = 0 \tag{11}$$

We highlight that the attacker knows the values $A_{(s_1s_2)}$, $B_{(s_1s_2)}$ and $C_{(s_1s_2)}$. However, it is believed to be hard to compute the value pw_{ix} without solving the Equation (11) $(\text{mod } p)$ and $(\text{mod } q)$, which resumes to the factorization problem. Hence, the attacker cannot reveal the long-term secret password yet.

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

$$A_{(s_3s_4)}pw_{ix}^2 + B_{(s_3s_4)}pw_{ix} + C_{(s_3s_4)} = 0 \tag{12}$$

The attacker now finds pw_{ix} as the solution of the equation system formed by Equation (11) and Equation (12):

$$pw_{ix} = \frac{A_{(s_1s_2)}C_{(s_3s_4)} - A_{(s_3s_4)}C_{(s_1s_2)}}{A_{(s_3s_4)}B_{(s_1s_2)} - A_{(s_1s_2)}B_{(s_3s_4)}} \quad (\text{mod } n) \tag{13}$$

Once U_a obtains pw_{ix}, he computes pw_{iy} as:

$$pw_{iy} = f(pw_{ix})_{(s_j)} - K_{i(s_j)} + pw_{ix} \tag{14}$$

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.

The attack assumes that the protocol allows multiple sessions between the same parties. This is a natural assumption, since the secret key should be changed periodically for security reasons. However, if it is considered suspicious that a single user initiates the protocol multiple times with the same other participant, a coalition of insiders may mount the attack: each attacker initializes a different session with the victim U_i and finally they cooperate to disclose the long-term key password $pw_{ix}||pw_{iy}$.

5 Conclusions

Yuan et al. recently defined a password-based group key transfer protocol that uses secret sharing [9]. Unlike the authors claim that it provides key confidentiality, we prove the opposite: their construction is vulnerable to insider attacks. First, we remark its similarity to the protocol of Harn and Lin back in 2010 [1]. This makes it susceptible to the replay attack that Nam et al. initially introduced for Harn and Lin's construction [3]. Second, we propose a new attack, which shows that the protocol remains vulnerable to insiders attack even if it is improved to stand against replay attacks.

6 Acknowledgement

This paper is supported by the Sectorial Operational Program Human Resources Development (SOP HRD), financed from the European Social Fund and by the Romanian Government under the contract number SOP HDR/107/1.5/ S/82514.

References

[1] L. Harn, C. Lin, Authenticated Group Key Transfer Protocol based on Secret Sharing, IEEE Trans. Comput. **59(6)**, 842–846 (2010).
[2] C. Hsu, B. Zeng, Q. Cheng, G. Cui, A Novel Group Key Transfer Protocol, Cryptology ePrint Archive, Report 2012/043 (2012).
[3] 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).
[4] R. F. Olimid, On the Security of an Authenticated Group Key Transfer Protocol Based on Secret Sharing, In: K. Mustofa et al. (Eds.) ICT-EurAsia 2013, LNCS **7804**, pp. 399–408. IFIP International Federation for Information Processing (2013).
[5] J. Pieprzyk, C. H. Li, Multiparty Key Agreement Protocols, In: IEEE Proceedings - Computers and Digital Techniques, pp. 229–236 (2000).
[6] G. Sáez, Generation of Key Predistribution Schemes using Secret Sharing Schemes, Discrete Applied Mathematics **128(1)**, pp. 239–249 (2003).
[7] A. Shamir, How to Share a Secret, Commun. ACM **22(11)**, 612–613 (1979).
[8] Y. Sun, Q. Wen, H. Sun, W. Li, Z. Jin, H. Zhang, An Authenticated Group Key Transfer Protocol based on Secret Sharing, Procedia Engineering **29(0)**, 403 – 408 (2012).
[9] 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 (2013).

Ruxandra F. Olimid holds a Bachelor Degree in Mathematics and Computer Science - University of Bucharest (2008) - and a Bachelor Degree in Telecommunications Engineering - Politechnica University of Bucharest (2009). She received the Master Degree in Computer Science at the University of Bucharest (2010). She is currently a PhD candidate and a Teaching Assistant at the Computer Science Department of the University of Bucharest. Her main interests lie in the field of cryptography: public key cryptography, group-oriented cryptography, secret sharing, group key establishment protocols.