Security Analysis on a Group Key Transfer Protocol Based on Secret Sharing*

Mijin Kim[1], Namje Park[2], and Dongho Won[1,**]

[1] College of Information and Communication Engineering, Sungkyunkwan University,
2066 Seobu-ro, Jangan-gu, Suwon, Gyeonggi-do, 440-746, Korea
{mjkim,dhwon}@security.re.kr
[2] Department of Computer Education Teachers College,
Jeju National University, Jeju, Korea
namjepark@jejunu.ac.kr

Abstract. Group key exchange protocols are cryptographic algorithms that describe how a group of parties can communicate with their common secret key over insecure public networks. In 2013, Olimid proposed an improved group key transfer protocol based on secret sharing, and claimed that he eliminated the flaws in Sun et al.'s group key transfer protocol. However, our analysis shows that the protocol is still vulnerable to outsider and insider attacks and does not provide known key security. In this paper, we show a detailed analysis of flaws in the protocol.

Keywords: key exchange protocol, group key transfer, secret sharing, attack, confidentiality.

1 Introduction

Secure group communications over public networks require that all group participants have to share a common secret key. This shared secret key, called the session key, is used to expedite authentication, confidentiality, and data integrity services. Group key transfer protocols are designed to achieve the fundamental security goal that no one except the group participants can establish the session key. Over the years, various protocols [1,2,3,4,5,6,7,8,9] have been proposed to achieve the fundamental goal of securely distributing a session key among a group of n participants.

Recently, Sun et al. presented a group key transfer protocol based on secret sharing instead of encryption algorithm [8]. The protocol only needs the server to broadcast $n+1$ messages at once in a round of distribution and all of the legal users only need to store one secret share in all conversations regardless of new addition or someone's

* This research was supported by the MSIP(Ministry of Science, ICT&Future Planning), Korea, under the C-ITRC(Convergence Information Technology Research Center) support program (NIPA-2013-H0301-13-3007) supervised by the NIPA(National IT Industry Promotion Agency).

** Corresponding author.

James J. (Jong Hyuk) Park et al. (eds.), *Mobile, Ubiquitous, and Intelligent Computing*, Lecture Notes in Electrical Engineering 274,
DOI: 10.1007/978-3-642-40675-1_72, © Springer-Verlag Berlin Heidelberg 2014

walkout. In addition, a simple computation is enough for each user to obtain the key. However, due to a flaw in Sun et al.'s protocol design, the protocol fails to achieve the fundamental security goal. In 2013, Olimid showed that Sun et al.'s protocol is susceptible to insider attacks and violates known key security and proposed an improved version of the protocol that eliminated the flaws of the original protocol. In this work, we provide a security analysis on the improved group key transfer protocol. Our analysis shows that the protocol still has flaws in the design and can be easily attacked. We present insider attack, outsider attack and failure of known key security on the protocol.

This paper is organized as follows: Section 2 reviews Olimid's group key transfer protocol. Section 3 presents security analysis of the protocol. Finally, Section 4 concludes this work.

2 Olimid's Group Key Transfer Protocol

This section reviews an improved group key transfer protocol [9]. The protocol assumes a trusted key generation center (KGC) who provides key distribution service to its registered users, and consists of two phases: user registration, group key generation and distribution. The protocol adopts the following derivative secret sharing scheme.

Derivative Secret Sharing

Phase 1: Secret sharing

1. KGC splits S into two parts n times: $S = s_1 + s_1^{'} = s_2 + s_2^{'} = \cdots = s_n + s_n^{'}$.
2. KGC sends P_i the share $s_i^{'}$, *i=1,2,…,n,* respectively in a secure channel.

Phase 2: Reconstruction

1. KGC broadcasts the shares $s_i, i = 1,2,\ldots,n,$ at once when users want to recover the secret.
2. P_i regains S by computing $S = s_i + s_i^{'}$.

The derivative secret sharing reduces the mutual dependence on others. Detailed steps of these phases are described as follows.

2.1 Olimid's Protocol

Let U be a set of all users who can participate in the protocol. The users in any subset of U may run the protocol to establish common session key.

Phase 1: User registration: Each user is requested to login to KGC for subscribing the group key distribution service. During registration, KGC shares a long-term secret $s_i^{'}$ with each user U_i ∈U .

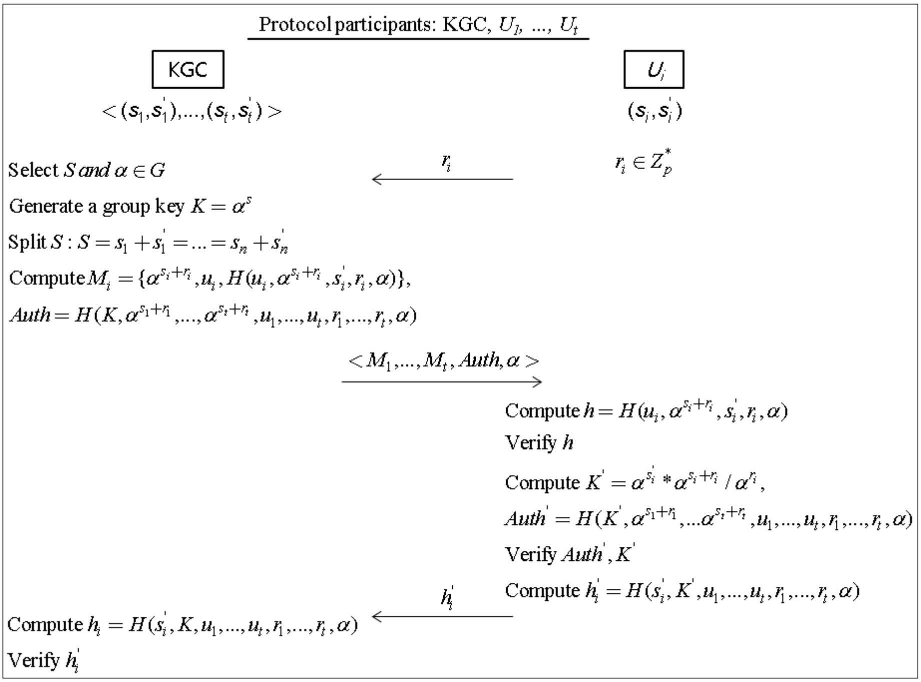
Protocol participants: KGC, U_1, ..., U_t
KGC
U_i
$<(s_1,s_1^{'}),\ldots,(s_t,s_t^{'})>$
$(s_i,s_i^{'})$
Select S and $\alpha \in G$
r_i
$r_i \in Z_p^*$
Generate a group key $K = \alpha^S$
Split $S: S = s_1 + s_1^{'} = \ldots = s_n + s_n^{'}$
Compute $M_i = \{\alpha^{s_i+r_i}, u_i, H(u_i, \alpha^{s_i+r_i}, s_i^{'}, r_i, \alpha)\}$,
$Auth = H(K, \alpha^{s_1+r_1}, \ldots, \alpha^{s_t+r_t}, u_1, \ldots, u_t, r_1, \ldots, r_t, \alpha)$
$<M_1, \ldots, M_t, Auth, \alpha>$
Compute $h = H(u_i, \alpha^{s_i+r_i}, s_i^{'}, r_i, \alpha)$
Verify h
Compute $K^{'} = \alpha^{s_i^{'}} * \alpha^{s_i+r_i} / \alpha^{r_i}$,
$Auth^{'} = H(K^{'}, \alpha^{s_1+r_1}, \ldots \alpha^{s_t+r_t}, u_1, \ldots, u_t, r_1, \ldots, r_t, \alpha)$
Verify $Auth^{'}, K^{'}$
Compute $h_i^{'} = H(s_i^{'}, K^{'}, u_1, \ldots, u_t, r_1, \ldots, r_t, \alpha)$
$h_i^{'}$
Compute $h_i = H(s_i^{'}, K, u_1, \ldots, u_t, r_1, \ldots, r_t, \alpha)$
Verify $h_i^{'}$

Fig. 1. An execution of Olimid's protocol (described from Step 3)

Phase 2: Group Key Generation and Distribution:

1. The initiator, a designated user of the group, requests for a group key distribution service by sending KGC $\{u_1, u_2, \ldots, u_t\}$, which contains the identities of the registered users $U_1, U_2, \ldots, U_t$, in current session.
2. KGC broadcasts the list of all participants according to the above received message as a response.
3. Each U_i, *i=1, … , t* sends a random challenge r_i to KGC.
4. KGC randomly selects S and α to generate the group key $K = \alpha^S$ for current service and then invokes derivative secret sharing to split *S* into two parts t times such that $S = s_1 + s_1^{'} = s_2 + s_2^{'} = \cdots = s_t + s_t^{'}$. KGC then computes: $M_i = \{\alpha^{s_i+r_i}, u_i, H(u_i, \alpha^{s_i+r_i}, s_i^{'}, r_i, \alpha)\}$ and *Auth=H(K,* $\alpha^{s_1+r_1}$, $\ldots, \alpha^{s_t+r_t}, u_1, \ldots, u_t, r_1, \ldots, r_t, \alpha)$. At last, KGC broadcasts $\{M_1, \ldots, M_t, Auth,\ \alpha\}$ to the users at once.
5. After receiving $M_i, Auth$, and *α*, U_i computes $h = H(u_i, \alpha^{s_i+r_i}, s_i^{'}, r_i, \alpha)$, where $\alpha^{s_i+r_i}$ and u_i are from $M_i, s_i^{'}$ is the shared long-term secret stored by U_i, r_i as chosen in step 3. And then U_i checks whether or not *h* is equal to the corresponding part in M_i. If any of the checks fails, U_i aborts; Otherwise, U_i computes $K^{'} = \alpha^{s_i^{'}} * \alpha^{s_i+r_i} / \alpha^{r_i}$, $Auth^{'} = H(K^{'}, \alpha^{s_1+r_1}, \ldots,$ $\alpha^{s_t+r_t}, u_1, \ldots, u_t, r_1, \ldots, r_t, \alpha)$ and checks whether or not $Auth^{'}$ is equal to *Auth*. If so, then $K^{'}$ is the correct group session key *K* which is distributed by KGC.

6. Each user U_i returns a value $h_i^{'} = H(s_i^{'}, K^{'}, u_1, \dots, u_t, r_1, \dots, r_t, \alpha)$ to KGC. KGC computes $h_i = H(s_i^{'}, K, u_1, \dots, u_t, r_1, \dots, r_t, \alpha)$ with its own $s_i^{'}$ and K, and checks whether or not $h_i^{'} = h_i$. This review confirms that every user in current session has obtained the correct group key.

3 Security Analysis

In this section, we analyze the security features of the improved group key transfer protocol based on secret sharing described in Section 2. The fundamental security goal of a key exchange protocol is to ensure that no one other than the intended users can compute the session key. But, Olimid's protocol fails to achieve this fundamental security goal. We describe this security vulnerability of the improved group key transfer protocol.

3.1 Outsider Attack

To an outside adversary, his motivation is to obtain the group key or share the group key with group participants. In the following analysis, we can see that his aim is come true.

Method 1:

1. The adversary A can grasp $r_i, M_1, \dots, M_t, Auth,$ and α from the broadcast channel between KGC and authorized users U_i.
2. Since A knows $M_i = \{\alpha^{s_i + r_i}, u_i, H(u_i, \alpha^{s_i + r_i}, s_i^{'}, r_i, \alpha)\}$ $(i = 1,2, \dots, t)$, A is able to obtain $H(u_i, \alpha^{s_i + r_i}, s_i^{'}, r_i, \alpha)$.
3. Using the grasped values $r_i, \alpha^{s_i + r_i}, u_i,$ and α, A is able to obtain $s_i^{'}$ from $H(u_i, \alpha^{s_i + r_i}, s_i^{'}, r_i, \alpha)$ by guessing attack.
4. Thus, A is able to calculate the session key $K = \alpha^{s_i^{'}} * \alpha^{s_i + r_i} / \alpha^{r_i}$.

Method 2:

1. A can grasp $r_i, M_1, \dots, M_2, Auth,$ and α from the broadcast channel between KGC and authorized users U_i.
2. From $M_i = \{\alpha^{s_i + r_i}, u_i, H(u_i, \alpha^{s_i + r_i}, s_i^{'}, r_i, \alpha)\}$ $(i = 1,2, \dots, t)$, A is able to obtain u_i and $\alpha^{s_i + r_i}$.
3. Using the grasped values *Auth*, $\alpha^{s_1 + r_1}, \dots, \alpha^{s_t + r_t}, u_1, \dots, u_t, r_1, \dots, r_t,$ and α, A is able to obtain the session key K From $Auth = H(K, \alpha^{s_1 + r_1}, \dots, \alpha^{s_t + r_t}, u_1, \dots, u_t, r_1, \dots, r_t, \alpha)$ by launching a guessing attack.

3.2 Insider Attack

Every inside user in Olimid's protocol is expected to reconstruct the group key but know nothing more extra information. However, our analyses show that malicious inside user P_i can forge the return response $H(s_j^{'}, K^{'}, u_1, \dots, u_t, r_1, \dots, r_t, \alpha)$ and impersonate P_j as following.

Method 1:

1. A can grasp $r_1, \ldots, r_t, M_1, \ldots, M_t, Auth,$ and α from the broadcast channel between KGC and authorized users U_i.
2. Since A knows $M_j = \{\alpha^{s_j+r_j}, u_j, H(u_j, \alpha^{s_j+r_j}, s_j^{'}, r_j, \alpha)\}$ $(i = 1,2,\ldots,t)$, A is able to obtain $H(u_j, \alpha^{s_j+r_j}, s_j^{'}, r_j, \alpha)$.
3. Then, A knows $r_j, u_j, \alpha^{s_j+r_j},$ and α , A is able to obtain $s_j^{'}$ from $H(u_j, \alpha^{s_j+r_j}, s_j^{'}, r_j, \alpha)$ by guessing attack.
4. Using the obtained $s_j^{'}$, malicious inside user P_i can forge the $P_j^{'}s$ response message $H(s_j^{'}, K^{'}, u_1, \ldots, u_t, r_1, \ldots, r_t, \alpha)$.

Method 2:

Let $U_a \in$U be an authorized user for a session (K_1), $s_a^{'}$ be his long-term secret, $U_{(k_1)} \subseteq$ U be the qualified set of participants of the session, $(\alpha^{s_{i(k_1)}+r_{i(k_1)}})_{U_i \in U_{(k_1)}}$ be the values that were broadcasted as part of $(M_i)_{U_i \in U_{(k_1)}}$ in step 4 , and $K_{(k_1)} = \alpha^{S_{(k_1)}}$ be the session key.

1. The participant U_a is qualified to determine (k_1) session key as: $K_{(k_1)} = \alpha^{s_a^{'}} \cdot \alpha^{s_{a(k_1)}+r_{a(k_1)}} / \alpha^{r_{a(k_1)}}$.
2. Since $\alpha^{s_{i(k_1)}+r_{i(k_1)}}$ and $r_{i(k_1)}$ are public, he is able to compute $\alpha^{s_i^{'}}$, for all $U_i \in U_{(k_1)}$: $\alpha^{s_i^{'}} = K_{(k_1)} \cdot \alpha^{r_{i(k_1)}} / \alpha^{s_{i(k_1)}+r_{i(k_1)}}$.
3. Suppose that U_a is unauthorized to recover (k_2) session key, $(k_2) \neq (k_1)$. But, he can eavesdrop the exchanged messages, then he is able to compute $\alpha^{s_{j(k_2)}} = \alpha^{s_{j(k_2)}+r_{j(k_2)}} / \alpha^{r_{j(k_2)}}$ for all $U_j \in U_{(k_2)}$, where $U_{(k_2)} \subseteq$U is the qualified set of parties of the session (k_2).
4. The inside adversary U_a can find the key $K_{(k_2)}$ of the session (K_2) as: $K_{(k_2)} = \alpha^{s_b^{'}} \cdot \alpha^{s_{b(k_2)}} = \alpha^{s_b^{'}+s_{b(k_2)}}$ where $U_b \in U_{(k_1)} \cap U_{(k_2)}$. Thus, an insider is able to compute any session key under the assumption that at least one authorized participant for both sessions exists.
5. Then, the inside adversary is able to obtain others' secret shares $s_j^{'}$ $(i = 1, \ldots, t)$.
6. Since U_a knows $(h, u_j, \alpha^{s_j+r_j}, s_j^{'}, r_j, \alpha)$, U_a is able to forge a $P_j^{'}$s response message $h_j = H(u_j, \alpha^{s_j+r_j}, s_j^{'}, r_j, \alpha)$. Thereafter, U_a is able to impersonate P_j.

Unlike to the Olimid's claim, his improved group key transfer protocol is still vulnerable to insider attacks.

3.3 Known Key Security

Suppose an adversary owns a session key $K_{(k_1)}$. We also assume that he had previously eavesdropped values $r_{i(k_1)}$ in step 3, $\alpha^{s_{i(k_1)}+r_{i(k_1)}}$ and α from the broadcasted message in step 4 of session (k_1), then he is able to compute $\alpha^{s_{i(k_1)}} =$

$\alpha^{s_{i(k_1)}+r_{i(k_1)}}/\alpha^{r_{i(k_1)}}$ for all $U_i \in U_{(k_1)}$. Because the session key $K_{(k_1)}$ is exposed, he can also compute the long term secret $\alpha^{s_i'}$, for all $U_i \in U_{(k_1)}$: $\alpha^{s_i'} = K_{(k_1)}/\alpha^{s_{i(k_1)}}$.

Let (k_2) be any previous or future session that has at least one common qualified participant U_b with (k_1), i.e. $U_b \in U_{(k_1)} \cap U_{(k_2)}$. As before, the adversary eavesdropped $r_{b(k_2)}$, $\alpha^{s_{b(k_2)}+r_{b(k_2)}}$, α and computed $\alpha^{s_{b(k_2)}} = \alpha^{s_{b(k_2)}+r_{b(k_2)}}/\alpha^{r_{b(k_2)}}$.

The adversary can now recover the key $K_{(k_2)}$:

$$K_{(k_2)} = \alpha^{s_b'} \cdot \alpha^{s_{b(k_2)}} = \alpha^{s_b'+s_{b(k_2)}}.$$

Therefore, an adversary is able to disclose any session key under the assumption that a session key has been compromised.

4 Conclusion

In 2013, Olimid proposed an improved group key transfer protocol based on a special secret sharing scheme [9]. He claimed that his improved protocol eliminated insider attack and provided known key security. However, our analysis shows that any inside/outside adversary can obtain the session key and impersonate legal users. Therefore, the improved protocol does not meet the fundamental security goal. Future work could be undertaken to remedy Sun et al. [8] and Olimid protocols.

References

1. Shamir, A.: How to share secret. Communications of the ACM 22(11), 612–613 (1979)
2. Katz, J., Yung, M.: Scalable protocols for authenticated group key exchange. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 110–125. Springer, Heidelberg (2003)
3. Nam, J., Paik, J., Kim, U.M., Won, D.: Resource-aware protocol for authenticated group key exchange in integrated wired and wireless networks. Journal of Information Sciences 177, 5441–5467 (2007)
4. Hajyvahabzadeh, M., Eidkhani, E., Mortazavi, S.A., Pour, A.N.: A new group key management protocol using code for key calculation: CKC. Information Science and Applications, 1–6 (2010)
5. Harn, L., Lin, C.: Authenticated group key transferprotocol based on secret sharing. IEEE Transactions on Computers 59(6), 842–846 (2010)
6. Nam, J., Paik, J., Won, D.: A security weakness in Abdalla et al.'s generic construction of a group key exchange protocol. Journal of Information Sciences 181(1), 234–238 (2011)
7. Nam, J., Kim, M., Paik, J., Won, D.: Security Weaknesses in Harn-Lin and Dutta-Barua protocols for group key establishment. KSII Transactions on Internet and Information Systems 6(2), 751–765 (2012)
8. Sun, Y., Wen, Q., Sun, H., Li, W., Jin, Z., Zhang, H.: An authenticated group key transferprotocol based on secret sharing. Procedia Engineering 9, 403–408 (2012)
9. Olimid, R.F.: On the security of an authenticated group key transfer protocol based on secret sharing. In: Mustofa, K., Neuhold, E.J., Tjoa, A.M., Weippl, E., You, I. (eds.) ICT-EurAsia 2013. LNCS, vol. 7804, pp. 399–408. Springer, Heidelberg (2013)