Improvements of Chang-Wu group-oriented authentication and key exchange protocols

Hung-Yu Chien
Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan, R.O.C.

Tzong-Chen Wu
Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C.

Jinn-Ke Jan
Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan, R.O.C.

Yuh-Min Tseng
Department of Information Management, NanKei College of Technology and Commerce, NanTou, Taiwan, R.O.C.

Abstract

In 1998, Chang and Wu proposed a group-oriented authentication mechanism with key exchange (Computer Communications, vol. 21). The authors show that the Chang-Wu protocols are vulnerable to the impersonation attack. They also give improvements that can not only eliminate the security flaw, but also reduce the redundant messages inherent in the original protocols.

Keywords: cryptography, group-oriented authentication; key exchange; impersonation attack

1 Introduction

In 1998, Chang and Wu [1] proposed a group-oriented authentication mechanism with key exchange. The Chang-Wu mechanism consists of three basic subprotocols: the entity-to-entity (EE), the entity-to-group (EG), and the group-to-group (GG) subprotocols, which are performed in a bottom-up approach. First of all, any two entities can invoke the EE subprotocol for authentication and key exchange with another entity to form an authenticated subgroup. Meanwhile, the remaining single entity (if it exists) can invoke the EG subprotocol for authentication and key exchange with some existing subgroup to form a new authenticated subgroup. After that, two authenticated subgroups can run the GG subprotocol to form a larger authenticated subgroup or the target group eventually. Since these protocols are performed in a bottom-up approach, and finally formed as a binary tree structure), the gain of communication cost (which is measured by the number of interactions between n entities) can be reduced from $O(n^2)$ to $O(n\log n)$ when compared to the mechanisms based on traditional end-to-end protocols. The Chang-Wu mechanism only uses one-way hashing function and exclusive-OR operations to establish the authentication (sub)protocols, and hence, it is very suitable for group-oriented applications.

In this paper, we will show that the Chang-Wu protocols are vulnerable to the impersonation attack. We also give improvements that can not only eliminate the security flow, but also reduce the redundant messages inherent in the original protocols.

2 Brief review of the Chang-Wu protocols

The Chang-Wu system requires a trusted authentication server (*AS*) that is responsible for issuing group session keys and validating the exchanged *MAC*s (the message authentication codes) during the initialization stage. It is assumed that each entity in the system shares a distinct long-term key (master key) with *AS* in advance. Besides, *AS* will publish a noticeboard and a one-way hash function to all entities in the system.

2.1 Notations and Symbols

In the following, we first give notations and symbols that will be used in the proposed subprotocols EE, EG and GG.

AS	the authentication server

NB	the public noticeboard
W, V	the temporal authenticated subgroups
G	the target group (or the temporal authenticated subgroup) eventually formed by the authenticated entities in the system
$\|X\|$	the numbers of entities in the subgroup X
GID_X	the identity for the subgroup X
U_i, ID_i	the i-th entity in group G, with the identity ID_i
U_{X_i}	the i-th entity in subgroup X, where U_{X_1} stands for the representative entity for X
f	a one-way hash function that gets input of arbitrary length and generates output of fixed length
$K_{S,i}$	the master key shared between AS and U_i
K_X	the session key for subgroup X
$K_{X,Y}$	the session key shared between the representatives of subgroups X and Y (i.e., U_{X_1} and U_{Y_1})
$K_{i,X}$	the session key shared between the entities U_i and U_{X_1}
MAC	a message authentication code, which is defined as $MAC(D_1, D_2, ..., D_k) = f(D_1 \| D_2 \| \Lambda \| D_k)$
C_i, C_X	the credentials for U_i and subgroup X, $C_i = MAC(ID_i, K_{S,i})$ respectively, defined as and $C_X = MAC(ID_X, K_X)$
N_S, N_i	the nonces issued by AS and U_i, respectively
$CKSM_{U_i \alpha X}$	the checksum value computed by U_i with respect to the authenticated subgroup W, which is defined as $f(K_{S,i}, C_X)$
$CKSM_{X \alpha U_i}$	the checksum value collaboratively computed by all entities in X with respect to U_i, which is defined as $CKSM_{X \alpha U_i} = CKSM(\|X\|) = f(K_{S,X_{(X)}}, CKSM(\|X\|-1))$ with the initial value $CKSM(1) = f(K_{S,X_1}, f(K_X, C_i))$
$CKSM_{X \alpha Y}$	the checksum value collaboratively computed by all entities in X with respect to Y, which is defined like $CKSM_{X \alpha U_i}$ with the initial value $CKSM(1) = f(K_{S,X_1}, f(K_X, C_W))$
$A \rightarrow B$	A transmits message to B
$A \quad NB$	A posts message to the noticeboard
$A \Leftarrow NB$	A downloads message from the noticeboard

2.2 Basic subprotocols

Here, we only list the basic subprotocols proposed by Chang and Wu [1]. The encouraged reader can refer the same paper for detailed explanations and the calling sequences of these subprotocols.

Subprotocol EE(U_i, U_j, W)

/* Two entities authenticate each other, share a session key, and form a temporal authenticated subgroup */

1 U_i -> AS: ID_i, ID_j, $f(K_{s,i}) \oplus N_i$

2. AS → U_i: N_s, $f(ID_i, K_{s,i} \oplus N_i) \oplus K_w$, $MAC(K_w, K_{s,i} \oplus N_i)$,$f(ID_j, K_{s,j} \oplus N_s) \oplus K_w$, $MAC(K_w, K_{s,j} \oplus N_s)$

3. U_i → U_j : Ns, $f(ID_j, K_{s,j} \oplus N_s) \oplus K_w$,

$MAC(K_w, K_{s,j} \oplus N_s)$, $f(ID_i, K_w) \oplus N_i'$, $MAC(ID_i, N_i', K_w)$

4. $U_j \rightarrow U_i$: $MAC(ID_j, N_i'+1)$

AS encapsulates the group key K_w in the messages for U_i and U_j in step 2. U_i recovers the key, and then forwards the messages prepared by AS together with his challenge to U_j. Finally, U_j extracts the key K_w and sends his response back to U_i.

Next comes the description of the subprotocol EG. There exists two phases in EG : EG1. and EG.2. In EG.1, U_i and U_{w_1} exchange their encapsulated credentials in a secret manner. The credentials C_i and C_w are encapsulated by a temporal session key $K_{i,w}$. Before the representative U_{w_1} proceeds to subprotocol EG.2, he should invoke an intra-group subprotocol to get $CKSM_{w \rightarrow U_i}$. In EG.2, U_i and U_{w_1} forward their CKSMs to AS. AS validates these CKSMs and then posts the new group session key K_v in an encapsulated manner to the noticeboard. Finally, U_i and U_{w_i} extract the new group key from the noticeboard.

Both EE and EG.1 cannot withstand the impersonation attack. We will show that an adversary impersonates another entity to exchange key later. Also some weakness should be noticed and then be improved. We will present the details in Section 3.

Subprotocol EG phase1(U_i, W, V) : exchange credential

1. U_i ->AS : $ID_i, GID_w, f(K_{s,i}) \oplus N_i$

2.AS-> U_i : $N_s, f(ID_i, N_s, K_{s,i} \oplus N_i) \oplus K_{i,w}, MAC(K_{i,w}, K_{s,i} \oplus N_i), f(GID_w, N_s, K_{s,w_1}) \oplus K_{i,w}, MAC(N_s, K_{i,w}, K_{s,w_1})$

3. U_i -> U_{w_1} : $N_s, f(GID_w, N_s, K_{s,w_1}) \oplus K_{i,w}, MAC(N_s, K_{i,w}, K_{s,w_1}), f(ID_i, K_{i,w}) \oplus N_i', f(ID_i, N_i', K_{i,w}) \oplus C_i, MAC(C_i, N_i', K_{i,w})$

4. U_{w_1} ->U_i : $f(GID_w, N_i'+1, K_{i,w}) \oplus C_w, MAC(C_w, N_i'+1)$

Calculating $CKSM_{w \rightarrow U_i}$

1. U_{w_1} **calculates** CKSM(1) as $f(K_{s,w_1}, f(K_w, C_i))$

U_{w_1} --> U_{w2} : $f(ID_{w_1}, K_w) \oplus N_{w_1}, f(ID_{w_1}, N_{w_1}, K_w) \oplus$ CKSM(1)

2. For i=2, 3, …, |W|-1

U_{w_i} **calculates** CKSM(i) as $f(K_{s,w_i}, CKSM(i-1))$

U_{w_i} --> U_{w_i+1} : $f(ID_{w_i}, K_w) \oplus N_{w1}, f(ID_{w_i}, N_{w_1}, K_w) \oplus$ CKSM(i)

3. $U_{w_{|W|}}$ **computes** CKSM(|W|) as $f(K_{s,w_{|W|}}, CKSM(|W|-1))$

$U_{w_{|W|}}$ --> U_{w_1} : $f(ID_{w_{|W|}}, K_w) \oplus N_{w_1}, f(ID_{w_{|W|}}, N_{w_1}, K_w) \oplus$ CKSM(|W|)

4. U_{w_1} takes the extracted CKSM(|W|) as $CKSM_{w \rightarrow U_i}$

Subprotocol EG Phase2(U_i, W, V) : get exchange key

1. U_i -> U_{w_1} : $ID_i, f(ID_i, K_{s,i}) \oplus N_i'', f(ID_i, N_i'', K_{s,i}) \oplus CKSM_{U_i \rightarrow w}, MAC(CKSM_{U_i \rightarrow w}, N_i'', K_{s,i})$

2. U_{w_1} -> AS : $GID_w, N_{w_1}', f(N_{w_1}', K_{s,w_1}) \oplus N_{w_1}, f(GID_w, N_{w_1}) \oplus K_w, f(GID_w, N_{w_1}, K_w) \oplus CKSM_{w \rightarrow U_i}, MAC(CKSM_{w \rightarrow U_i}, N_{w_1}, K_w), ID_i, f(ID_i, K_{s,i}) \oplus N_i'', f(ID_i, N_i'', K_{s,i}) \oplus CKSM_{U_i \rightarrow w}, MAC(CKSM_{U_i \rightarrow w}, N_i'', K_{s,i})$

3. AS=> NB : $GID_v, f(N_{w_1}, K_w) \oplus K_v, MAC(N_{w_1}, K_v), f(N_i'', K_{s,i}) \oplus K_v, MAC(N_i'', K_v)$

4. U_i <=NB : $GID_v, f(N_i'', K_{s,i}) \oplus K_v, MAC(N_i'', K_v)$

5. U_{w_i} <=NB : $GID_v, f(N_{w_1}, K_w) \oplus K_v, MAC(N_{w_1}, K_v)$

3. An attack and improvements

3.1. Impersonation attack and our improvement:

In step 3 of EE, U_i will forward the message $f(ID_j, K_{s,j} \oplus N_s) \oplus K_w$ and his challenge $f(ID_i, K_w) \oplus N_i'$ to U_j. We can see that these two messages are only linked by key K_w, so that entity U_i can legally follow step 1-2 to get the group key K_w and then impersonate another entity U_k by forging U_k's challenge message $f(ID_k, K_w) \oplus N_k'$. We show the

impersonation scenario as follows, where $U_{i\,(k)}$ means that U_i impersonates U_k.

Impersonation scenario EE(U_i, U_j, W)

1 U_i -> AS: ID_i, ID_j, $f(K_{s,i}) \oplus N_i$

2. AS → U_i: N_s, $f(ID_i, K_{s,i} \oplus N_i) \oplus K_w$, $MAC(K_w, K_{s,i} \oplus N_i)$, $f(ID_j, K_{s,j} \oplus N_s) \oplus K_w$, $MAC(K_w, K_{s,j} \oplus N_s)$

3. $U_{i\,(k)}$ → U_j : Ns, $f(ID_j, K_{s,j} \oplus N_s) \oplus K_w$, $MAC(K_w, K_{s,j} \oplus N_s)$, $f(ID_k, K_w) \oplus N_k$, $MAC(ID_k, N_k, K_w)$

4. U_j → $U_{i\,(k)}$: $MAC(ID_j, N_k+1)$

By a similar approach, an entity can impersonate another entity or group in EG.1 and GG.1 to exchange his credential. We will not list the similar attacks here, but proceed to introduce our improvement. We propose the improvement by including the identity ID_i in AS's message $f(ID_j, K_{s,j} \oplus N_s) \oplus K_w$ as follows.

2*. AS → U_i: N_s, $f(ID_i, K_{s,i} \oplus N_i) \oplus K_w$, $MAC(K_w, K_{s,i} \oplus N_i)$, ID_i, $f(ID_i \oplus ID_j, K_{s,j} \oplus N_s) \oplus K_w$, $MAC(K_w, K_{s,j} \oplus N_s)$

3*. U_i → U_j : Ns, ID_i, $f(ID_i \oplus ID_j, K_{s,j} \oplus N_s) \oplus K_w$, $MAC(K_w, K_{s,j} \oplus N_s)$, $f(ID_i, K_w) \oplus N_i$, $MAC(ID_i, N_i, K_w)$

The inclusion of ID_i in $f(ID_i \oplus ID_j, K_{s,j} \oplus N_s) \oplus K_w$ by AS will explicit inform U_j that the new session key is K_w and the authenticated entity is U_i. The message authentication code $MAC(K_w, K_{s,j} \oplus N_s)$ assures U_j that the freshness and the integrity of the messages $f(ID_i \oplus ID_j, K_{s,j} \oplus N_s) \oplus K_w$. This means that U_i has no way to forge a valid messages $f(ID_k \oplus ID_j, K_{s,j} \oplus N_s) \oplus K_w$ without knowing $K_{s,j}$ even though he can forge U_k's challenge. Now U_j can detect the impersonation attack by finding the inconsistency between the challenge and the forwarded messages.

3.2. Cutting the unnecessary responses and enhancing the robustness of AS

We can see that AS is the bottleneck and the single-point-of-failure in the mechanism. But, AS does not directly authenticate the entity who sends messages to him in the step 1 of EE, EG and GG. It implies that AS always responses in step 2 even though an entity sends an invalid message or a replay message to him. This weakness will encourage an adversary to try his attacks since AS can not validate the messages in step 1. It not only deteriorates the system performance but also weakens the robustness of the system [3]. We make the improvement by replacing step 1 as follows.

1*. U_i -> AS: ID_i, ID_j,$Timestamp_{U_i}$, $f(K_{s,i}) \oplus N_i$, $MAC(K_{s,i}, N_i, Timestamp_{U_i})$

The MAC enables AS to authenticate U_i directly. The timestamp here proves the freshness of the messages to AS as long as its value is greater than the one in the previous invocation by U_i. This mechanism requires no synchronization on a global clock. The improved protocol enables AS to authenticate U_i directly and check the freshness of messages in step 1, so that AS no longer response to invalid or replay messages in step 2. Combined with the fail-stop protocol design [3], it maybe lowers the incentives to an adversary. This improved protocols release AS from preparing responses to invalid or replay requests. It also enhances the robustness of the system.

3.3. Eliminating the redundancies in phase 2:

There are some redundancies can be eliminated in phase 2. We take EG.2 as an example for illustration. Firstly, we will point out some redundancies in step 2 of EG.2. Then we present our ideas to eliminate the redundancies and list the improved protocols. Finally, we make a comparison on the numbers of messages used.

The messages $f(N_{w_1}', K_{s,w_1}) \oplus N_{w_1}$ and $f(GID_w, N_{w_1}) \oplus K_w$ in step 2 are used to securely conveyed N_{w_1} and K_w to AS respectively. But, we see that U_{w_1} does not have to send message $f(GID_w, N_{w_1}) \oplus K_w$ just for sending K_w to AS since AS assigned and posted that value in the noticeboard previously. N_{w_1} here serves as a secret among

AS and all entities in subgroup W because U_{w_1} secretly shares this value with his group members during the $CKSM_{w-->U_i}$ calculation process. This value is used later to well protect the new session key K_v in message $f(N_{w_1}, K_w) \oplus K_v$. We can see that the temporal session key $K_{i,w}$ of EG1 can be used for the same purpose because it could become a shared secret among AS and entities in W if we substitute all N_{w_1}s with $K_{i,w}$ in the $CKSM_{w-->U_i}$ calculation process. But, U_{w_1} does not have to send it in another message because AS already knows this value. This arrangement will reduce the numbers of nonces and messages in EG2. The improved EG2* is listed as follows:

Subprotocol EG* Phase2() : get exchange key

1. U_i -> U_{w_1} : ID_i, $f(K_{i,w}, K_{s,i}) \oplus CKSM_{U_i-->w}$, $MAC(CKSM_{U_i-->w}, K_{i,w}, K_{s,i})$
2. U_{w_1} -> AS : GID_w, $f(GID_w, K_{i,w}, K_w) \oplus CKSM_{w-->U_i}$, $MAC(CKSM_{w-->U_i}, K_{i,w}, K_w)$, ID_i, $f(K_{i,w}, K_{s,i}) \oplus CKSM_{U_i-->w}$, $MAC(CKSM_{U_i-->w}, K_{i,w}, K_{s,i})$
3. AS=> NB : GID_v, $f(K_{i,w}, K_{s,i}) \oplus K_v$, $MAC(K_{s,i}, K_v)$, $f(K_{i,w}, K_w) \oplus K_v$, $MAC(K_{i,w}, K_v)$
4. U_i <=NB : GID_v, $f(K_{i,w}, K_{s,i}) \oplus K_v$, $MAC(K_{i,w}, K_v)$
5. U_{wi} <=NB : GID_v, $f(K_{i,w}, K_w) \oplus K_v$, $MAC(K_{i,w}, K_v)$

Compared with the original one, the revised EG2* eliminates N_{w_1}, N_{w_1}' and N_i". It also reduces one message in step 1 and four messages in step 2. This elimination of messages does not damage the security of the system. We can see that the new group key K_v is well protected by $K_{s,i}$, K_w and $K_{i,w}$ respectively. The revised protocols eliminate the redundancies without weakening the security of the system.

4. Conclusions

In this paper, we have shown an impersonation attack on Chang-and-Wu's group-oriented authentication with key exchange protocols. An improvement against this attack has been presented. We have also pointed out that AS does not directly validate or check freshness of those messages in phase1. This weakness encourages an adversary to try his attacks, so that it will worsen the system performance and weaken the system security. In protocols of EG2, we find that lots of redundancies can be eliminated by incorporating $K_{i,w}$ into the design of protocols in EG2. In summary, the revised protocols of phase 1 and phase2 gain more security with reduced load.

References

[1] Y. S. Chang and T. C. Wu, "Group-oriented authentication mechanism with key exchange", Computer Communications, Vol. 21, pp. 485-497, 1998

[2] W. Diffie and P.C. van Oorschot and M.J. Wiener, "Authentication and authenticated key exchanges, Designs", Codes and Cryptography, Vol. 2, No. 2, pp. 107-125, 1992

[3] Y. Ding and P. Horster, "Undetectable on-line password guessing attacks", ACM Operating Systems Review, Vol. 29, pp. 77-86, 1995