Anonymous Authentication in Dynamic Group

2001027, Jongseong Kim

Information and Communications University

1 Introduction

omitted.

2 Previous Works

omitted.

3 Proposed Anonymous Authentication Scheme

In this Section, we describe the proposed anonymous authentication scheme which allows multiple commitments in a specific session round and assures forward security.

3.1 System Set-up

The entities of our scheme consists of the group manager(GM),the verification center(VC), and m group members U_i $(i = 1, ..., m)$. The role of each entity is as follows:

GM
- He has responsibilities to manage a group and determines large prime q, primitive root a of q, and an universal hash function H.
- He generates a shared session key K_{GT} collaborating with VC and it is kept secret both GM and VC.
- He generates random numbers R_i $(i = 1, ..., m)$ which will be assigned randomly by GM to the group member participated in the current group.
- He maintains a secret database for keeping member's id U_i and it's assigned random number R_i.

VC
- She has responsibilities to verify the member participating a current session is valid or not.
- She generates a shared session key K_{GT} collaborating with GM.
- She maintains a secret database for keeping each member's last hash value $H^n(S_i)$.

Member (U_i where $i = 1, ..., m$)
- Member has to register to GM to participate in the current group.

– He can commitment himself many times using his chained hash values $H^{n-t}(S_i)$.

In general, TTP seems to be and must be trust and assumed as honest entity and it usually knows the key information about related entities. But in our scheme, VC has much less meaningful information needed to identify a group member. Without collaborating GM with VC, neither can know member's identity.

3.2 Anonymous Authentication protocol

The proposed protocol consists of the following 4 steps. Whenever GM wants to make a group for special purpose, he can execute preparation step since a group can be made temporarily. Registration of group member is only one-time in the protocol, but commitment step is executed whenever U_i commits and this step has ability to verify validity of U_i if the commitment of U_i is not first time. Finally step 4 is executed.

Step 1. Preparation :

1. GM chooses large prime q and primitive root a of q and make them public and prepares public universal hash function H.
2. GM selects random integer X_G as secret and compute $Y_G = a^{X_G} \bmod q$ and encrypts Y_G using VC's public key and sends VC it.
3. VC selects random integer X_V as secret and compute $Y_V = a^{X_V} \bmod q$ and computes a session secret key $K_{GV} = Y_G^{X_V} \bmod q$.
4. VC encrypts Y_V using K_{GV} and encrypts both it and Y_V using GM's public key and sends it to GM.
5. GM decrypts Y_V using his secret key, computes a session secret key $K_{GV} = Y_V^{X_G} \bmod q$ and verify $Y_T \stackrel{?}{=} D(Y_T)_{K_{GV}}$
6. If valid then GM generates random numbers $(R1, R_2, ..., R_m)$ and expiration date T and encrypts them using K_{GV} and sends it to VC.
7. VC decrypt it using K_{GV} and stores all them to the secure database in order.

Step 2. Member Registration:
A group member U_i who wants to commit registers to GM only once:

1. U_i encrypts his user ID using GM's public key and sends it to GM.
2. GM chooses a random number R_i randomly and encrypts it using K_{GV} and encrypts again it using U_i's public key $(E_{U_{i_{pub}}}[E(R_i)_{K_{GV}}])$.
3. U_i decrypts it using his private key to find out $E(R_i)_{K_{GV}}$, chooses a seed and hashes it n times$(H^n(S_i))$, encrypts them using VC's public key and sends it to VC.
4. VC decrypts it and verifies $R_i \stackrel{?}{\in} (R_1, R_2, ..., R_n)$ and if valid sends message "*Accept*" to the member U_i and stores $R_i, H^n(S_i)$ to the secure database.

Step 3. Commitments: A group member U_i can commit whenever he wants:

1. U_i determines t, computes hash chain value $H^{n-t}(S_i)$, encrypts it and some INFO using GM's public key and sends $E_{G_{pub}}[H^{n-t}(S_i), INFO]$ to GM. INFO can be any information (such as bidding price).
2. When the member U_i commits firstly, GM delegates member verification to VC sending by $E_{VC_{pub}}[H^{n-t}(S_i)]$, otherwise GM can verifies member validity himself.
3. VC decrypts it, verifies $H^{n-t}(S_i) \stackrel{?}{=} H^n(S_i)$. If valid then send message "OK" to GM.

Stage 4. Finding Member's Identity : When expiration date is over, GM and VC can find out the qualified group member easily.

1. GM sends $E_{VC_{pub}}[H^{n-t}(S_i)]$ to VC.
2. VC decrypts it and verifies $H^t[H^{n-t}(S_i)] \stackrel{?}{=} H^n(S_i)$.
3. If valid then VC sends R_i to GM.
4. GM bulletins U_i who has the random number R_i is just the qualified group member.

3.3 Analysis Securities and Efficiencies

4 Future Works

5 Conclusion