Proposal for Secure Group Communication using Encryption Technology

Shinya Tanada
Graduate School of Science and
Technology, Meijo University,
Aichi 468-8502, Japan
Email: shinya.tanada@wata-lab.meijo-u.ac.jp

Hidekazu Suzuki
Graduate School of Science and
Technology, Meijo University,
Aichi 468-8502, Japan
Email: hsuzuki@meijo-u.ac.jp

Katsuhiro Naito
Information Science,
Aichi Institute of Technology,
Aichi 470-0392, Japan
Email: naito@pluslab.org

Akira Watanabe
Graduate School of Science and
Technology, Meijo University,
Aichi 468-8502, Japan
Email: wtnbakr@meijo-u.ac.jp

***Abstract*—In order to realize secure information sharing, it is common to encrypt information, using a common key called ”Group Key” among group members. However, in the case of existing technologies, there is a concern that leakage of information occurs because contents of communications are preserved at the server, and also because the server administrator possesses the group key. Thus, in this Paper, we propose a method of using two kinds of random numbers of different generation origins sent through two different transmission routes in order to create a group key. With this method, group members can share the group key with security, and a secure group communication system with enhanced level of security can be realized.**

I. Introduction

With the development of network technologies, people’s interest in secure information sharing through the Internet has been increasing. The secure group communication is an important technology when members of a group share information. However, according to ”Secure Messaging Scorecard”[1] issued by ”Electronic Frontier Foundation”(EFF)[2], a non-profit organization evaluating the sufficiency of security of message applications, it is indicated that the security of current message applications is quite vulnerable. Among evaluation items set by EFF, there is an item that requires us to check if users’ data are encrypted so that even the server administrator cannot read the data. The background for including this item throught to be associated with a scandal revealed in 2013 where the ”National Security Agency”(NSA) had been obtaining information from big IT companies in the back. There exists an application named ”ChatSecure”[3] which satisfies all requirements of the security evaluation items established by EFF, but this is a one-to-one message application and cannot be used for group communication. If we have a group communication system that can be used for business, satisfying all the requirements of the items of EFF, it will be quite useful.

As a representative example of group communication, we have a chat application called LINE, which is used by many people in Japan. Although LINE is encrypted on the transmission route, it is rarely used for business matters by companies because its security is not considered to be sufficient. Moreover, since contents of exchanged messages are preserved at the server in the form of plain texts, there is a concern that they are leaked to third parties through the server.

As technologies to realize group communication with due consideration for security, ”Group Key Management Architecture”(GKMA)[4] developed by ”Multicast Security”(MSEC) Working Group as well as its improved version called ”Group Secure Association Key Management Protocol”(GSAKMP)[5] is standardized. These technologies encrypt communication contents, by using group keys.

Communication systems using group keys usually need to satisfy certain specific key management requirements. The key management requirements include such matters as procedures to invite users to a certain group and key renewal intervals. GSAKMP satisfies all of these requirements. In the case of GSAKMP, ”Group Controller Key Server”(GCKS) performs the functions of creating, delivering and renewing the group key. And it is presupposed that each of the group members and the GCKS possesses a public key certificate, and they can share a group key with security by performing mutual authentication, using the public key certificate. However, this method does not satisfy part of the evaluation items set by EFF. That is to say, it is possible for a malicious server administrator to read communication contents, because the GCKS possesses a group key.

In this Paper, we propose a new method of sharing the group key to solve the above-mentioned problems. Our proposed method is constituted of ETs (End Terminals) used by users and ”Group Management Server”(GMS). Two kinds of random numbers are created by each of the ETs and the GMS, and these two kinds of random numbers are shared by the ETs through different transmission routes, and a ”Group Key”(GK) is created using these random numbers. This proposed method can satisfy the evaluation items of EFF, because one of the random numbers exchanged among ETs directly cannot be read by the server administrator. It is also possible to satisfy the group key management requirements for group communication, by setting appropriate intervals for renewing random numbers to be generated at the GMS.

Using the items set by EFF’s evaluation and key management requirements, we obtained a result that our proposed system is more effective than existing technologies.

The remainder of this Paper is organized as follows. We first explain existing technologies and their issues in Section 2, describe our proposed method in detail in Section 3, compare our proposed method with the existing technologies in Section 4, and summarize the result of our study in Section 5.

II. Key management requirements and Existing technologies

A. Key management requirements

The objective of the group key management protocol is to provide group members with a set of secret information required for secrecy and authentication by using encryption technologies, and we have generally the following key management requirements.

- The key should be renewed periodically with predetermined intervals.
- The key data should be delivered with securely to members of the group so that they can be kept completely in secrecy, integrity-protected and verifiably obtained from authorized sources.
- The key management protocol should be secure against replay attacks[1] and ”Denial of Service”(DoS) attacks.
- The protocol should facilitate addition and removal of group members. Newly added members are denied to get access to any data existing before their joining the group (backward secrecy), and removed members cannot get access to any data after their departure (forward secrecy).

B. LINE

We describe the behaviors and security of ”LINE”, which is the most popular chat application in Japan among various group communication tools. Fig. 1 shows a method of realizing chatting by LINE. The method is constituted of end nodes used by users and the chat server. Every message created by users is sent to the server, and the server in turn transfers the message to all group members. Each user can freely invite anyone whom the user wants to have as a group member, and the number of group members is gradually expanding. Messages to invite someone to the group also go through the chat server. LINE does not have anything like a group key, and transmission data is encrypted. The setting of a password to be used for authentication is carried out at the time of the account registration of a new user. Authentication process between the chat server and the member is the same as that for ordinary client-server systems. That is to say, the authentication of the chat server is executed by the server's public key certificate and that of the member by its password. In the process of authentication, a common key is shared between the chat server and the end node, and thereafter, all transmissions are encrypted.

However, due to the reason that LINE does not have a group key, it cannot satisfy key management requirements. And furthermore, because messages are to be preserved at the server in the form of plain texts, contents could be viewed by a malicious server administrator. Accordingly, LINE's security is considered to be quite vulnerable.

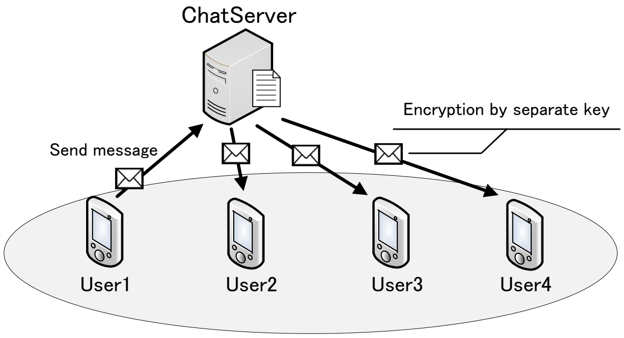


Fig. 1. Method of realizing chatting by LINE

C. GSAKMP

GSAKMP is a security framework for group communication, and this is standardized as RFC4535. It establishes a group by performing user authentication processing based on access control rules and offers a security policy of the relevant group. As an example of using GSAKMP, we can name the group communication for IETF participants.

In GSAKMP, three major components, namely the group owner, the Group Controller Key Server (GCKS), and group members share the roles of group management. In GSAKMP, not only the GCKS but also each of the group members possesses a public key certificate, and consequently, mutual authentication between the GCKS and each group members can be performed with certainty.

The group owner takes the charge of establishing and providing the security policy including such matters as key renewal intervals and ways of inviting members. Based on this security policy, the GCKS and group members are able to perform operations related to the creation of a group and the security of group communication. The GCKS is a server taking the charge of creating and delivering the group key, renewal of the key, and management of group members, based on the security policy. Group members, when group key was renewed, have to check if that was done properly based on the security policy.

Fig. 2 shows the key sharing sequence of GSAKMP. The key sharing method of GSAKMP is composed of the GCKS and users. While the group owner determines the security policy, it is not shown in the Fig. 2 as it is not directly involved in the key sharing process. It is presupposed that the users newly participating in the group are to be invited by the group owner or by any of the members already in the group. The numbers shown in Fig. 2 correspond to the following.

1) The invited user sends a ”Request to Join” to the GCKS. This is a request for participation, and in this participation message, such information as the user's own public key certificate and the group ID given to the user by the inviting member is included.

[1] ”Reply attack” is an act of making an unauthorized login, by copying and intercepting the data which users sent to the network, at the time of login operation and request to join, and by sending the data to the authentication server.

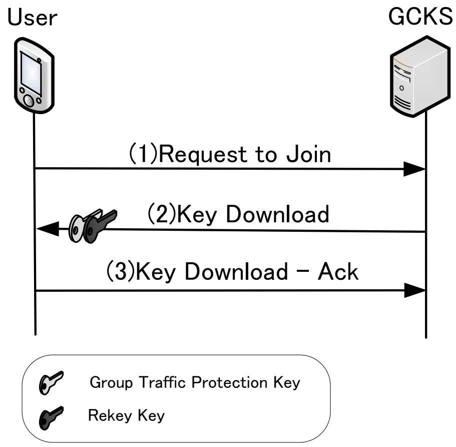


Fig. 2. Key sharing sequence of GSAKMP

2) The GCKS, after verifying the public key certificate of the user who will newly participate in the group, delivers two kinds of keys by way of ”Key Download”. One is a key called ”Group Traffic Protection Key”(GTPK), which encrypts group data, and the other is a key called ”Rekey Key”, which is used to renew the GTPK.
3) The user responds Key Download-Ack, and through this process, the invited user becomes a member of the relevant group.

For renewal of the group keys, notification of group key renewal is sent from the GCKS to each of the group members, who in turn makes the renewal of the GTPK using the Rekey Key. Because the GTPK is renewed every time a user joins the group and withdraws from the group, both of the forward secrecy and the backward secrecy are secured. GSAKMP satisfies all of the key management requirements.

Mutual authentication using the public key certificate is executed only once at the first time, because the calculation concerning the public key needs so many steps and subsequent authentication is executed by the shared GTPK and Rekey Key. By using GSAKMP, information will never be leaked from the chat server even if the chatting is made through the server, because the contents of communication are encrypted. However, GSAKMP has some problems. Because the GCKS is creating and delivering both GTPK and Rekey Key, this method does not satisfy one of the evaluation items recommended by EFF, which requires that the communication contents shall be encrypted so that even administrators cannot read them.

Another problem for this method is that each member is required to possess a public key certificate, and the acquisition and maintenance of the public key certificate incurs costs.

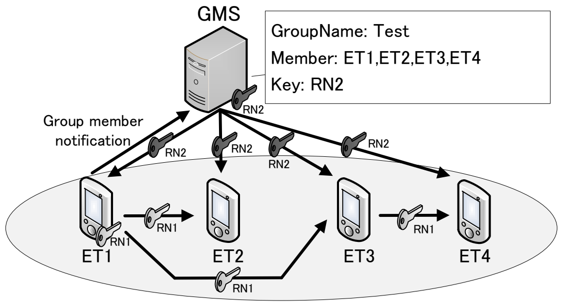


Fig. 3. System configuration of our proposed method

III. Our proposed method

In order to realize secure group communications, we propose a method of creating a group key (GK), by using two types of random numbers of different origins of generation to be delivered to group members through different communication routes. Our proposed method satisfies not only the key management requirements but also the requirements established by EFF.

A. Outline of the configuration and procedures of our proposed method

Fig. 3 shows the configuration of our proposed method focusing upon the key sharing part. Our proposed method is composed of ”Group Management Server”(GMS) and ”End Terminals”(ETs). The definition of the group can be done either by the administrator or by group members. The GMS manages the group name and members of the group. It creates/delivers/manages a random number ”RN2” and renews it at pre-determined intervals. Each of the ETs makes a group member notification to the GMS. One of the ET in the group also creates another random number ”RN1” and shares it with other members directly. After obtaining RN1 and RN2, each of the ETs creates a GK. As RN1 and GK are components that are possessed by group members only, their strict safekeeping is required.

B. Key sharing method

The special feature of our proposed method lies in the point that RN1 is shared among group members directly in a way that the administrator cannot come to know it. As concrete ways of realizing it, we can think of the following two methods. One is the way for each ET of possessing a public key certificate like the case of GSAKMP. The other is the way of using a network in which end-to-end communication is possible.

In the case of using the public key certificate, a communication server can be used for the delivery of RN1, however the acquisition and maintenance of the public key certificate incurs costs. On the other hand, in the case of the network in

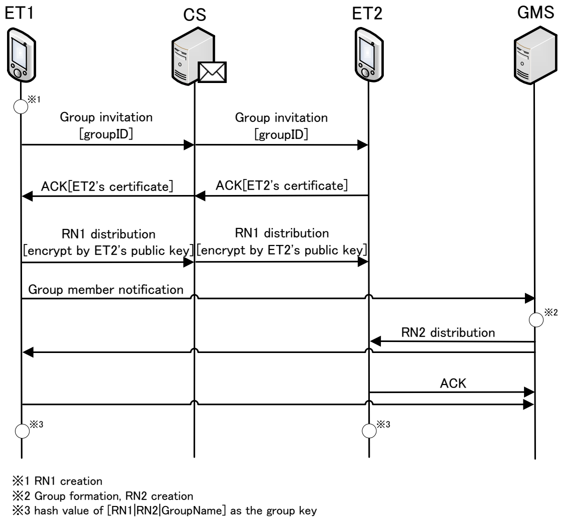

ET1
CS
ET2
GMS
※1
Group invitation [groupID]
Group invitation [groupID]
ACK[ET2's certificate]
ACK[ET2's certificate]
RN1 distribution [encrypt by ET2's public key]
RN1 distribution [encrypt by ET2's public key]
Group member notification
※2
RN2 distribution
ACK
※3
※3
※1 RN1 creation
※2 Group formation, RN2 creation
※3 hash value of [RN1|RN2|GroupName] as the group key

Fig. 4. Group forming sequence of our proposed method using the public key certificate

which end-to-end communication is possible, an end-to-end communication network needs to be arranged, although each ET does not need to possess a public key certificate.

Each ET, after sharing both RN1 and RN2, calculates the hash value of [RN1|RN2|GroupName], and regards the hash value as the GK of the group. According to this method, the GMS administrator is unable to know the content of the GK, and only the members who possess the same GK become the formal members of the group and can thereafter use the GK for mutual authentication as well as for encryption processing. In order to satisfy key management requirements, RN2 should be renewed frequency while the renewal intervals of RN1 can be relatively long. We describe two different methods of sharing the group key as below.

1) Key sharing method using the public key certificate:

Fig. 4 shows the group forming sequence of our proposed method using the public key certificate. The user who invites is defined as "inviter" and that who is invited is defined as "invitee". For communication among users, "communication server"(CS) is used. When forming a group, the first member ET1 creates RN1. The member who created RN1 can invite other users whom it wants to invite to the group as group members. Whether invitee may invite new members or not depends on the instructions from the original inviter. At the time of group invitation, the inviter sends a message of invitation to invitee, by attaching group ID to its message. The invitee sends back a response with its own public key certificate. The inviter distributes RN1 to invitee by encrypting it with the public key of invitee. Invitee decodes the encrypted RN1 with its own secret key. Through this process, the sharing of the RN1 among ETs is completed. As a sufficiently long renewal cycle of RN1 may be taken, each ET needs to carefully keep it so that it is not leaked to any non-members. Because of the encryption of RN1 by way of the public key, it is not possible even for the administrator of the CS to know the content of RN1.

Then, ET1 sends to the GMS a group member notification. The GMS, upon receiving the notification, creates a table for the management of the group. It also creates RN2 for the relevant group and distributes it among new group members. RN2 is created by the GMS with certain pre-determined renewal intervals and distributed to group members. RN2 is also renewed when any participating member withdraws from the group or when any user becomes a new member. With these procedures, both of the forward secrecy and the backward secrecy are ensured.

2) Key sharing method using end-to-end communication network:

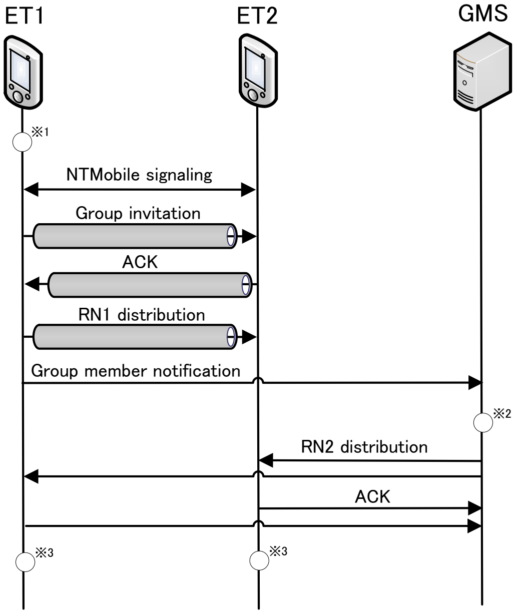
As a type of network in which end-to-end communication is possible, NTMobile[6][7] has been proposed. NTMobile is a technology by which both connectivity and mobility are realized simultaneously.

Connectivity indicates a mechanism where end nodes can initiate bidirectional communication regardless of the difference between global addresses or private addresses. Mobility indicates a mechanism where communication is maintained if a network is changed during the communication. NTMobile system is composed of NTM nodes and a "Direction Coordinator"(DC) set in the global address network. NTM nodes are the nodes with the NTMobile function installed. The DC determines the most appropriate communication route from the location of the NTM nodes and the configuration of the network, and instructs the NTM nodes to create a tunnel route. This process is hereafter called "NTMobile signaling". Thereafter, encrypted communication is realized end-to-end through the tunnel route. A common key is shared between the DC and each ET in advance using the public key certificate of the DC and user's password. Another common key is shared between the GMS and each ET in the same way.

Fig. 5 shows the group forming sequence of our proposed method using an end-to-end communication network. ETs, before initiating communication, create a tunnel route between the ETs by the NTMobile signaling, and realize the sharing by sending RN1 through the tunnel. After the sharing of RN1, a group member notification is sent from ET1 to the GMS. Like the method of using the public key certificate, the GMS creates a table for the management of the group. It also creates RN2 and delivers it to group members.

C. Renewal procedure for the group key

In order to satisfy key management requirements, it renews RN2 with a certain renewal intervals. RN2 is also renewed when any member withdraws from the group or when any new member is added, taking the forward secrecy as well as the backward secrecy into account. We describe the case where a member has withdrawn from the group and the case a new member has been added to the group.



※1 RN1 creation
※2 Group formation, RN2 creation
※3 hash value of [RN1|RN2|GroupName] as the group key

Fig. 5. Group forming sequence of our proposed method using end-to-end communication network

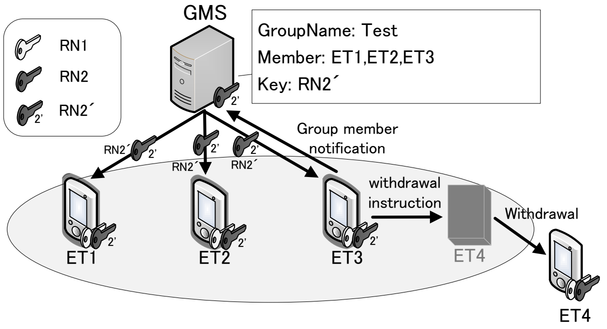


Fig. 6. Key renewal procedure when a member is withdrawn from the group

1) Case where a member has withdrawn:

Fig. 6 shows the key renewal procedure when a member is to withdraw from the group. As an example, we explain the case where ET3 has the right to give withdrawal instructions, and ET3 asks ET4 to withdraw. ET3 sends withdrawal instructions to ET4. ET4 is then forcibly withdrawn from the group. ET3 sends to the GMS ”group member notification”. The GMS creates new RN2’ and renews the member list of its own database. Thereafter, the GMS sends new RN2’ to the new members. When a GK is created at each ET using this new RN2’, the forward secrecy is ensured.

2) Case where a new member has been added:

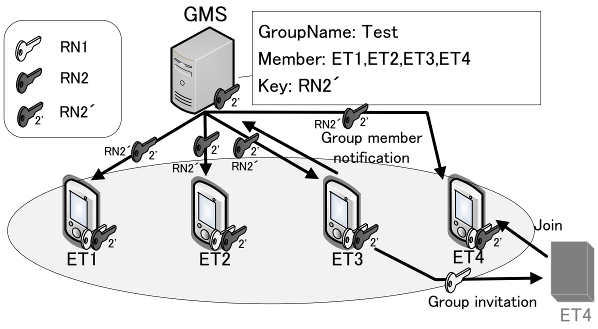


Fig. 7. Key renewal procedure when a member is newly added to the group

TABLE I
COMPARISON OF SIMILAR TECHNOLOGIES

	Item 1	Item 2	Item 3
LINE	×	×	○
GSAKMP	×	○	○
ChatSecure	○	×	×
Proposed method	○	○	○

The member who has the due authority can invite other users to become new members of the group. Fig. 7 shows the key renewal procedure when a member is newly added to the group. As an example, we explain the case where ET3 has the authority to add members, and ET3 invites ET4 to become a member. ET3 sends a group invitation notice to ET4. ET4, upon receiving the group invitation notice, sends its participation application to ET3. ET3 distributes RN1 to ET4. At this point of time, ET3 sends to the GMS ”group member notification”. The GMS, upon receiving the notice, renews the member list of its own database. Thereafter, the GMS sends new RN2’ to all members of the group. When a GK is created at each ET using the hash value, the backward secrecy is ensured.

IV. EVALUATION

TABLE I shows a comparison of similar technologies. The contents of the evaluation items are as follows:

1) Key management requirements are satisfied.
2) Encryption is provided so that the administrator cannot read the contents.
3) Group communication is feasible.

Existing technologies compared with our proposed method are LINE, GSAKMP and ChatSecure. In the case of LINE, encryption is provided on the transmission route, but because LINE does not have a group key, it cannot satisfy key management requirements. Also the administrator can read information because data are preserved at the server in the form of plain texts. GSAKMP offers security with both of the forward secrecy and the backward secrecy ensured based on mutual authentication using the public key certificate and renewal of the group key, but since keys to be used for encryption of group communications are all distributed from the key server GCKS, there is a concern that a malicious server administrator reads communication contents. ChatSecure is an application which is performing encryption so that the administrator cannot read communication contents, but since this is an application for one-to-one chatting, it is not possible to use it for group communication and does not satisfy the key management requirements.

Our proposed method satisfies key management requirements, because RN2 is renewed properly. It also satisfies the requirements set by EFF, because RN1 is exchanged in an end-to-end manner.

V. Conclusion

In existing group communication systems, the problem was that malicious key server administrators could read communication contents because the group key remains in the key server. Accordingly, in this Paper, we proposed a method of sharing two types of random numbers of different generation origins through different distribution routes, and of using these two types of random numbers to create a group key. Based on this method, secure group communication is guaranteed. We also indicated that our proposed method satisfies key management requirements. Hereafter, we will plan to implement our proposed method to actual devices and conduct performance evaluations.

References

[1] Electronic Frontier Foundation : Secure Messaging Scorecard. https://www.eff.org/secure-messaging-scorecard .

[2] Electronic Frontier Foundation https://www.eff.org/

[3] ChatSecure -Encrypted Messenger for iOS and Android. https://chatsecure.org/

[4] Multicast Security(MSEC) Group Key Management Architecture,RFC 4046,IETF (2005).

[5] GSAKMP: Group Secure Association Key Management Protocol, RFC4535, IETF (2006).

[6] K. Kamienoo, H. Suzuki, K. Naito, A. Watanabe:Implementation and Evaluation of NTMobile to Achieve IP Mobility in IPv4 and IPv6 Networks Journal of Information Processing Society of Japan, Vol.54, No.10, pp.2288-2299, Oct.2013.

[7] H. Suzuki, K. Naito, K. Kamienoo, T. Hirose and A. Watanabe NTMobile: New End-to-End Communication Architecture in IPv4 and IPv6 Networks Proceedings of the 19th Anuual International Conference on Mobile Computing and Networking (Mobicom2013), pp.171-174, Oct.2013.