International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 6 Issue: 4 190 - 193

Group Rekeying Protocol for Secure communication

Asst. Prof. P. V. Kale
Dept. Computer Science & Engineering
P. R Pote College of Engineering & Management,
Amravati, Maharashtra, India
e-mail: prachivkale@gmail.com

Dipti Patil[1], Harshali Thombare[2] , Mansi Raut[3], Mayuri Gawande[4]
Dept. Computer Science & Engineering
P. R Pote College of Engineering & Management
Amravati, Maharashtra, India
e-mail: diptiapatil@gmail.com

***Abstract*—** Group communication is one of the approaches to impart the messages effectively. Yet, security is the issue for this, and thus keys are utilized to secure the information. In this paper, the key management in group communication, an advanced rekeying approach based on the Logical Key Hierarchy (LKH) and One way Function Tree (OFT) schemes are explained. The AES crypto scheme is used to rekey the keys and the performance of the approach is better than LKH and OFT. Forward and backward security is provided by the proposed rekeying scheme. It is a novel rekeying scheme for large-scale dynamic groups that levers on logical sub-grouping and join history. On contrary other schemes, subgroups only support efficient group key management, have no application meaning and are transparent to the application layer. It misuses the historical backdrop of joining events to build up an aggregate requesting among subgroups and among nodes in every subgroup, so as to effectively recoup from collusion attacks.

Keywords- *Logical Key Hierarchy, One way Function Tree, rekeying, group communication, security.*

*****

I. INTRODUCTION

Group communication is a substantial and efficient paradigm that can be used in a range of application scenarios, from wireless sensor networks (WSNs) to large scale distribution of contents. Following to this, a node becomes member of the group by explicitly joining it. Then, that node may send/receive broadcast messages to/from other group members.

Only group members can access group communication is the very first requirement. For this, each group member secretly share a cryptographic group key to securely exchange messages in the group. Node must be prevented from deciphering previous messages even if it has recorded them (backward security). When any member leaves the group, or is forced to leave, the node must be prevented from further accessing group communication (forward security)[1]. Rekeying is guaranteed by backward and forward security. New key is distributed when node joins or leaves the group. Also, it exploits the history of joining events to establish a total ordering among subgroups and among nodes in each subgroup, in order to efficiently recover from collusion attacks and different schemes display different levels of resilience. Re-initialization of total members is required in collision recovery in many schemes, i.e. all non compromised group members have to be separately reinitialized. It follows that the recovery overhead grows linearly with the group size with negative impact on the overall system performance and scalability.

II. GREP ALGORITHM

To asses GREP in terms of storage, communication, and computing overhead of rekeying upon node joining, node leaving and recovering from collusion. It asses storage and communication overhead as the number of information items that protocol actors store and transmit/receive, respectively, and the computing overhead as the number of performed cryptographic operations. Consider a group G composed of p subgroups with m nodes each, i.e. n = p*m, this well supports heterogeneous sub-grouping, but a homogeneous one allows us to asses performance with no significant absence of generality.

A. AES Scheme

The AES encryption scheme is mostly used. It can be explained as follows:

• Key Generation chooses two big prime number p and q such that n = p.q.
• Choose a random number e, where gcd(e,(p-1) (q-1)) = 1.
• Compute another number d, where ed / 1 mod (p-1) (q-1)).

Here (n, e) is the public key and the private key is d. After establishing the AES scheme, p and q should be destroyed [2].

B. One way Function

Let's function H : {0,1}*←{0,1}* is one-way if:

• There exists a Probabilistic Polynomial Time (PPT) algorithm that on input x output H (x).
• For every PPT algorithm A there is a negligible function V_A such that for sufficiently large k [2],

$$P[H(z)=y:x\overset{R}{\leftarrow}\{0,1\};y\overset{R}{\leftarrow}H(x);z\overset{R}{\leftarrow}A(1^k,y)\leq V_A(k)$$

C. Secret Key Multiplication

There are a few proposed versatility multicast aggregate rekeying conventions. Among them are aggregate key approach, contributory key assention bolstered by Diffie Hellman calculation, and sensible key tree based approach. Among the gathering re-keying strategies said above, SKM (privileged insights keys duplication) is one strategy that does not rely upon encryption/unscrambling for its gathering re-keying process yet it isn't excessively secure than other.

 ISSN: 2321-8169
190 - 193

Normally in a safe group communication protocol, the group controller sends to the group members another key to approve new users and also plays out the group rekeying for assemble clients at whatever point the key changes.

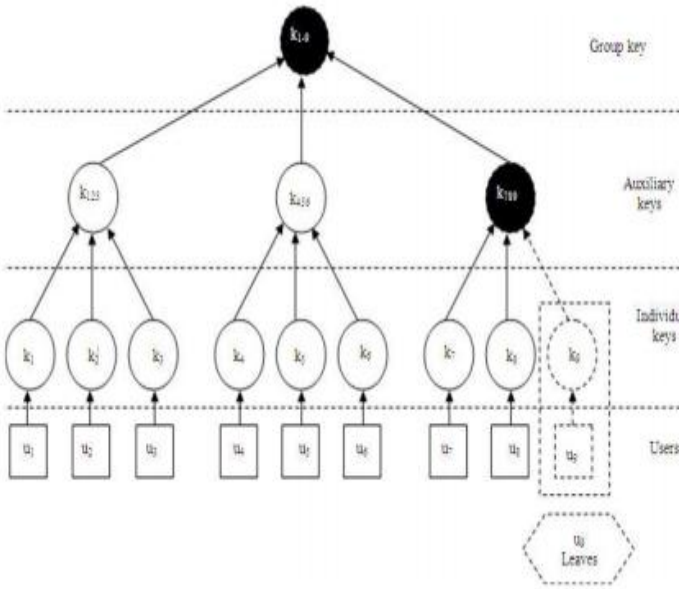

Figure 1. SKM Exapmle

The SKM protocol utilizes the consistent tree progressive system of key trade among the gathering individuals by increasing the gathering mystery keys. For information assurance, SKM protocol utilizes a secluded math which is connected to the individual key. SKM protocol utilizes mystery enter duplication in conjunction with the key tree approach. This approach is overseen by the trusted server called Gathering Controller (GC).

The user u_1 to u_9 holds individual keys as k_1 to k_9. K_{123} is the helper key offer by user u_1, u_2 and u_3. Correspondingly, k_{456} and k_{789} are shared by their users, u_4, u_6 and u_7, u_9, individually. K_{1-9} is the session key and is known to all the gathering individuals. Levels are dealt with a condition as that level-2 keys must be more prominent in esteem than level-1 and level-0 keys. Correspondingly, the level-1 keys must be more prominent in esteem than level-0 keys. To clarify the SKM aggregate re-keying protocol, expect user u_9 needs to leave the gathering. At that point the GC needs to change the mystery key which is known to u_9, and in addition different users. To deal with the keys, a re-keying process must be finished. K_{1-9} will be changed to k_{1-8}, k_{789} is changed to k_{78} and k_9 will be erased from the tree. Before produces another mystery key, the GC changes its private key from kc to kc'. In the wake of playing out the estimation as appeared by Eq. 1, the GC will multicast the qualities X and Y to whatever remains of the gathering individuals (u_1-u_8). Users u_7 and u_8 recuperate the new helper key, k_{78}, by utilizing their individual private keys k_7 and k_8 separately (Eq. 2 and 3). With the assistant key k_{78} users u_7 and u_8 can recoup the new session key, k_{1-8}, by executing Eq. 3. Similarly, users u_1-u_3 and u_4-u_6 can recover the new session key by using either the respective auxiliary keys,

K_{123} or k_{456}:

$X = k_7 \times k_8 \times kc' + k_{78}$

$Y = k_{123} \times k_{456} \times k_{78} \times kc' + k_{1-8}$ ---------- (1)

$K_{78} = X \bmod k_7$

$K_{78} = X \bmod k_8$ ---------- (2)

$K_{1-8} = Y \bmod k_{78}$ ---------- (3)

D. The proposed AES Algorithm

As mentioned above, the one-way function tree approach proposed by McGrew and Sherman is a bottom-up method. Rekeying is achieved by computing one-way function. Unlike, LKH is a top-down approach.

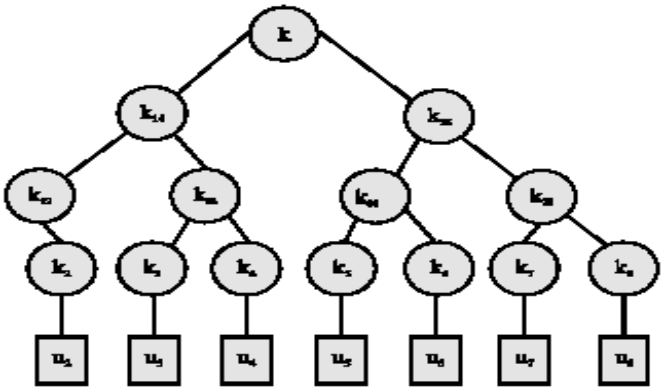

Figure 2. Key Tree

It tries to combine the two methods and establish an improved re-keying approach via AES crypto scheme. Without any loss of generality, we will describe our approach in a group consisted of eight users. Consider that there exists a Key Generation Center (KGC) in the system. KGC chooses two big prime numbers p and q such that n=p.q, a random private key d which meets gcd (e,(p-1) (q-1)=1. Then KGC computes the matching public key e to establish a AES crypto scheme. Assume that the users set is U = {u_2, u_3, u_3, u_4, u_5, u_6, u_7, u_8} and H is a cryptographic one way function. Following to the LKH approach, each user preserves the corresponding secret keys.

E. Join Operation

Whenever new user, without loss of generality, we use u_1 as the new user, joins the group, both the new user and the prior group users receive this notification. Steps for join operation are as follows:

- KGC chooses a random number X€Z (where Z ranges from 0 to n) and computes x^d,x^{d2},x^{d3},x^{d4}.
- KGC computes it and broadcast them in the group.
- The user u_2 can decrypt the cipher text and get x^{e4} via secret key k_2. Then computes $x^{d4.e} = x^d$. The keys {k_2, k_{12} ,k_{14} , k} are the rekeyed keys.

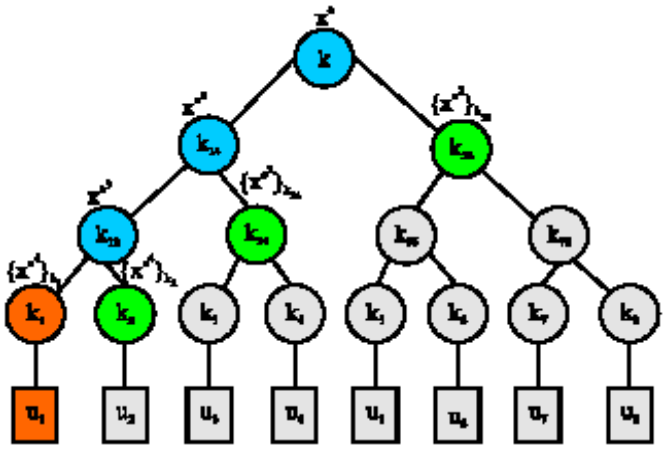

Figure 3. Join Operation

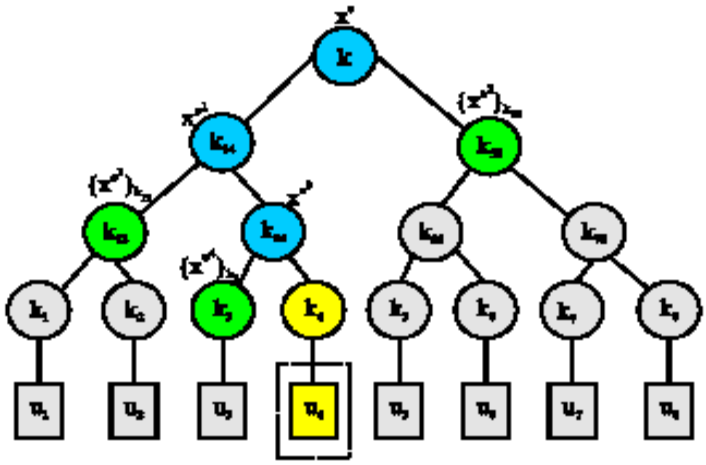

Figure 4. Leave Operation

- The user u_3 and u_4 obtain x^{d3} by the secret key k_3 and then get x^{d2} and x^d independently. Then it compute $k_{14} = H(x^{d2} \| k_{14})$ and $k= H(x^d \| k)$, respectively and preserve them as a rekeyed keys.
- x^{d2} is obtained by the users u_5,u_6,u_7,u_8 via secret key k_{58} and separately compute $x^{d2.e} = x^d$ and $k= H(x^d \| k)$.
- Random key k_1 for user u_1 is chosen and attributes $\{k_1,k_{12},k_{14},k\}$ to u_1 as it's secret key and then broadcast $\{x^{d4}\}_{k1}$.
- x^{d4} is obtained by user u_1 via secret key k_1 and then computes x^{d3}, x^{d2}, x^d. At last, user u1 computes k_{12},k_{14},k and completes the rekeying process.

This operation can be illustrated in figure 3.

F. Leave Operation

In case user u_4 will leave the group $U = \{u_1, u_2, L, u_8\}$. KGC has to update the keys related to user u_4, to provide backward security. Steps for leave operation are as follows:

- KGC chooses a random number X€Z (where Z ranges from 0 to n) and then computes x^d,x^{d2},x^{d3},x^{d4}.
- KGC computes $\{x^{d4}\}_{k4}$, $\{x^{d3}\}_{k34}$, $\{x^d\}_{k58}$ and and then broadcasts them
- The users u_1, u_2 obtain x^{d3} via secret key k_{12} and then compute $x^{d3.e}=x^{d2}$ and $x^{d2.e}=x^d$ Thereafter, they separately compute the keys and update it.
- The user u_3 decrypt and obtains x^{d4} via secret key k_3 and then computes x^{d3},x^{d2},x^d subsequently, user computes $k_{34}=H(x^{d3} \| k_3)$, $k_{14}= H(x^{d2} \| k_{14})$ and $k=H(x^d \| k)$ are the updated keys.

The user u_5, L,u_8 decrypt and and obtain x^{d2} via secret key k_{58} and then compute $x^{d2.e}=x^d$. At the end they compute $k=H(x^d \| k)$.

III. RELATED WORK

It is a novel group rekeying protocol that effectively rekeys a group with various messages which is small, consistent and autonomous of the group size. If there should be an occurrence of collusion attack, it recoups the group by misusing the historical backdrop of joining occasions. This keeps away from an aggregate part re-introduction and results in an overhead which easily develops with the gathering size, and steadily increments with attack seriousness [1]. The issue lies in creating the key management system which has a single common key for every one of the individuals from a virtual group that are basic for secure correspondence and should have accessibility and minimum excess. This normal key which is proposed as ViP-Key can be shared among the individuals from a virtual group with polynomial clustering expression, in this way making it less complex to accomplish proficient information transmission as opposed to utilizing a different key for every individual from that group [2]. Here the group is seen as a logical tree structure to diminish the overhead brought about at the group members amid the join/leave activity. The GKC makes an established adjusted tree that has the same number of leaf nodes as there are members. Each leaf node of the key tree is related with an individual from the group. Each inside node speaks to a consistent subgroup [4]. Sub-grouping approach is utilized to lessen the key updation overhead. By utilizing two keys at the transitional nodes, key updation is limited just to the particular subgroups where the participation change happened. This multicast key management protocol has less computational and correspondence cost of group rekeying than the past plans and furthermore it is compelling for expansive and very powerful multicast group [5].

IV. CONCLUSION

We have explained GREP, a novel group rekeying protocol that proficiently rekeys a gathering with various messages which is small, consistent and autonomous of the group size. If there should be an occurrence of collusion attack, GREP recuperates the group by misusing the historical backdrop of joining events. This stays away from an aggregate part re-instatement and results in an overhead which easily develops with the group size, and bit by bit increments with the attack seriousness. We have given a systematic execution assessment and demonstrated that GREP is deployable on huge scale systems of compelled gadgets.

The enhanced group rekeying come closer from AES crypto scheme in this report and portray the approach in adjust paired tree made out of eight clients. Correspondingly with LKH strategy, AES is top-down and meets forward and in reverse security. As per the overview, our approach has preferred execution over LKH and SKM.

REFERENCES

[1]. Marco Tiloca SICS Swedish ICT AB, “GREP: a Group REkeying Protocol Based on Member Join History” Security Lab Isafjordsgatan 22, Kista (Sweden), Gianluca Dini Dipartimento di Ingegneria dell’Informazione University of Pisa, Largo Lazzarino 1, Pisa (Italy),2016.

[2]. Chunbo Ma and Jun Ao, “Group Rekeying Approach for Group Communication” Information Technology Journal, pp. 1081-1084, 2008.

[3]. S. Krishnakumar and R. Srinivasan, “An Optimal Key Management Scheme for Group Key Sharing with Polynomial Expression”, Research Journal of Information Technology, pp. 12-23.

[4]. B. Parvatha Varthini and S. Valli, “Performance Analysis of Rekeying Protocols in Multicast Group Key Management”, Information Technology Journal, pp. 405-408, 2006.

[5]. V. Vijayaraghavan and R.S.D. Wahidabanu, “Effective Rekeying Architecture for Dynamic Multicast Group”, Information Technology Journal, pp. 247-250, 2007.

[6]. F. Bao, I. Chen, M. Chang and J. Cho, “Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection,” IEEE Trans. on Networkand Service Management, vol. 9, no. 2, pp. 1–15, 2012.

[7]. B. Latr´e, P. De Mil, I. Moerman, N. Van Dierdonck, B. Dhoedt and P. Demeester, “Maximum Throughput and Minimum Delay in IEEE 802.15.4,” in The First international conference on Mobile Ad-hoc and Sensor Networks, Wuhan,
China, vol. 3794. Springer, 2005, pp. 866–876.

[8]. D. Wallner, E. Harder and R. Agee, “Key Management for Multicast: Issues and Architectures”, IETF, 1999.

[9]. P. Liu, W.-C. Lee, Q. Gu and C.-H. Chu, “KTR: An Efficient Key Management Scheme for Secure Data Access Control in Wireless Broadcast Services,” IEEE Trans. on Dependable and Secure Computing, vol. 6, no. 3, pp. 188–201, 2009.

[10]. S. Rafaeli and D. Hutchison, “A Survey of Key Management for Secure Group Communication,” ACM Computing Surveys, vol. 35, no. 3, pp. 309–32.