Adaptive Secured Multicast Key Management with Re-Keying Process

Jahir Ibna Rafiq, Abdullah-Al-Omar
Dept. of Computer Science and Engineering
University of Asia Pacific,
Dhaka, Bangladesh.
jahir@uap-bd.edu, omar.cs.bd@gamil.com

Animesh Chakraborty, Albub Yusuf
Dept. of Computer Science and Engineering
University of Asia Pacific,
Dhaka, Bangladesh.
Animesh567@gmail.com, albubyusuf@gmail.com

***Abstract*— Data integrity is given top priority when exchange of information takes place. Information loses value, even becomes redundant should the intended user fail to obtain it. Modern day communications are mostly done on World Wide Web (WWW). Group communication must strike a balance between how quickly data is exchanged among users, at the same time, data security must not be compromised. Hence, protocols are in place to ensure data security in a cryptographic group by encrypting data before sending it to users. A separate transaction takes place with a key to decrypt the data. As the users grow in numbers, the key management becomes harder and complex in nature. Herein, Logical Key Hierarchy (LKH) has been proposed for a cryptographic group. A thorough analysis of the prevailing key management systems has been conducted. Few surveys have been researched and analyzed. Fundament security systems have been tested in real-life settings. A key management system needs to be in place that sharpens data security but provides users with quick service. Group Secure Association Key Management Protocol (GSAKMP) has been widely used. But online transaction and group that are geographically spread pose a different challenge. Hence, a proposal has been adopted to manage keying and rekeying for the members of a cryptographic group.**

Index Terms*— *Data Security, GSAKMP, Group Association, IPSec, Keying, Re-keying, Secure Communication.

I. INTRODUCTION

The security of data, in any circumstances, remains a vital point of contention. Modern day networks are multi-dimensional in nature, wide in size, liberal in geographical terms but one thing stays uptight – that is security of data within any perimeter of operation [1]. In theoretical terms, engineers propose prototypes. More often than not, a common endeavor to step-by-step processing is witnessed while analyzing the proposals. It is not a blunder yet it fails to withstand the demand of the industry. Hence, it is brought forward that singular way of transaction by enabling each client to receive service is neither practical nor profitable. It is more feasible to run parallel processes with agility that actually renders a service of value to the client. Therefore any multicast broadcast should be similar in nature for free unicast but there should be a barrier to prevent unsolicited intrusion. An efficient method that considers both speed and integrity, is to assign access-granting facility to the users in a group. Fundamental idea is to deliver key to the users that allows freely available data, but not meaningful without the key. So the server will send out data that has been encrypted with any existing technique [2]. Anyone within the range and perimeter of the network will get the encrypted data. Only the users with key will be able to decrypt the data – meaning a useful interpretation of the data will be shown upon presenting a valid key. Sending a key could be pictured like a rent-a-car. Cars are parked right in front of the gate. Customers pay the agreed fee and receive the key to the car. Only then they can avail themselves of the service. In a network setup the process of sending key or simply keying is absolutely easy to understand. As the number of users grows things start to change dramatically and after a certain extent of growth it becomes improbable to maintain the quality of service without a solid mechanical approach in keying. Users start to join, some leave, some fail to comply with basic terms i.e. fail to pay, abuse system etc, while some are always busy aggressively penetrating the system [3]. It is hundred times more hectic than theoretical terms and goes beyond simple rent-a-car analogy. Underneath all of this there remains one outstanding point – what sort of amalgamation of networks the users will lie beneath. Each network has to sanction transfer of data. While, Internet protocols are widely used without modification or little enhancement, challenges emerge from small network like private systems or type of user. In a library you would expect a different set of challenge than that of a hospital.

The paper is neatly organized. Introduction shines light on background and area of interest. In section II, a candid literature is put together regarding current key management protocols. The next section continues with the role of key management and the very impact it has on group communication. Our rekeying mechanism and relevant work is assembled in the next section. In section V, experiments and progress are shown. Next two sections summarize our work and establish goal for further benefit in near future.

978-1-5090-1181-0/16/$31.00 ©2016 IEEE

II. GROUP KEY MANAGEMENT PROTOCOLS

A secured multicast system depends on its group key distribution. When a key distribution system is used, it has to be ensured that the system is robust, secured, low bandwidth uses, minimal delay and computational complexity is minimized. There are two approaches to distribute the keys: static and dynamic. In static approach no key update required but the member of the system is fixed for lifetime and in dynamic approach any member can add or remove at any time. This section mainly tries to discuss about three important dynamic group key management systems.

A) Centralized b) Decentralized c) Distributed.

A. Centralized key management

This distribution system uses a single entity, which generates and distribute traffic encryption key (TEK). It provides group traffic key if any member wants to join or leave. It also generates new group traffic key for all members of a group. The following subsections give a brief review of different centralized key distribution system.

- *Group Key Management Protocol (GKMP)* – This is proposed by Harney and Muckenhirn [4]. In this approach when first member join in a group a key distributor center is generated for creating a group key packet (GKP). This distributor key also include group traffic encryption key (GTEK) and a group key encryption key (GKEK). All the members of a group knows the GKEK. When a member sends a request for joining, the distributor key sends a copy of GKP. At the time of rekeying the group controller generates new GKP which is encrypted with current *GKEK*.

- *Logical Key Hierarchy (LKH)* – This protocol is proposed independently by Wong et al. [5, 7] and Wallner et al. [6]. A set of key organized in a manner that resembles a tree, hence, a key tree, is maintained in LKH by the key server. Each member of LKH maintains a copy of keys along path from user to root of tree. A key encryption key (KEK) is assigned by a key distributor. There is a server in the background which is responsible for authenticate a new user, distribute and update keys. There are mainly three types of keys individual, subgroup and group keys. For every join or leave operation a new secure group is formed and for this server has to update existing key node and add or delete some key nodes. Then new group and subgroup key securely send with a rekey message.

B. Decentralized key management

Within decentralized management protocol, a new convention of subgroup approach is pursued. The subgroup managers manage the whole group by splitting in subgroups. The protocol efficiently distributes different parts of the work to the managers of subgroups to keep the process robust. Cryptography is imposed on its subgroup managers, encryption and decryption of the certain things is general between them. Failure of entities precedes the whole group's failure. This architecture is adept by having some characteristics-

Controllers are not centralized, which means any central manager for each case will not possess the group managers. Keys of the nodes are not dependent, each time the disclosure of the key doesn't compromise any part of the system.1-effect-n problem [8] is not present here, which means rekeying of a subgroup will not affect the whole group. The Rekeying process does not affect data communication. Communication of the group are said to be an n-to-n communication. Talking about the examples will tend us to take the name of,

Iolus[8], Scalable Multicast Key Distribution using Core Based Tree (CBT) [9], Kronos [10], Dual-Encryption Protocol (DEP) [11], Modification Tseng's key agreement protocol by Cheng and Laih [12], A non-interactive protocol by Huang, Kuo-Hsuan, et al. [13]

C. Distributive Key Management

It is exceptionally different from Decentralized key management protocol, because Group Controller have no contribution in this protocol. There are two ways by which keys could be generated; one way is all the members participate spontaneously to generate the key, which means each key would share information to generate the key. Another way around could be by a member from the group to generate the key for processing. Generating the key needs secure mechanism, which doesn't allow permitting any group member to generate the key. Robustness depends on the knowledge of the member list to each user. The characteristics that could be discussed-

Iteration number must be abated to decrease the requirements relates to processing and communication. There must be certain number of messages, which could be allowed to increase the rate of performance. Computation in setup time is a significant characteristic. Diffie-Hellman (DH) [14] allows the protocol to generate keys in contributory fashion, all the members contribute in key generation. Some example includes, Deffie-Hellman Logical Key Hierarchy [15], Octopus Protocol [16], Distributed Logical Key Hierarchy [17].

III. KEY MANAGEMENT ROLE

Multicast communication held between single or more senders and a group of receivers in each transmission. As the information or data are transmitted through various network channels it is must to secure the data also ensure that only the valid or authorized group members receive data. And because of this a group controller uses various key management schemes, which plays very important role in controlling the access of the data to the authorized members. Key management also helps in maintaining the key distribution and established a secured policy for key distribution in a group [18]. It uses some procedures and techniques to bring out essential factors for access control and secured group key distribution like – a. Identification and authentication of group members, b. Access control, c. Key generation and distribution.

A. Identification and authentication of group members

Key management provides identification and authentication of group members. Both identification and authentication is very important to prevent a trespasser to imitate any authorized

group members. It also helps to prevent the manipulation of key from an attacker. So it is must to be implemented both identification and authentication mechanism for verifying the actual legitimate group members.

B. Access Control

After recognizing the actual group members, their join operation is checked. Only the validated members get access to group communication. So access control validates the group members to give permission for access in the group communication.

C. Key generation and distribution

For securing the data, encryption is used whenever data are sent to a group. Before decrypting the data, a key is required. For better security purpose key should be changed frequently to ensure forward and backward security of information, which depends on the joining and leaving of new members in a group [19]. Forward secrecy blocked the access for future group data for those members who already left the group. In Backward secrecy all the older group data are blocked so that newly joining members cannot get access in those data. And for these reasons key should be changed whenever a member leave the group or a new member joining the group [20]. When generating the key, each key should be unique or independent from previous key so that unauthorized users fail to obtain the key. Rekeying for a small group can be easy but for thousands of members rekeying one by one can be very costly. So for rekeying a group key distributer must provide which contain mechanism for efficient rekeying and distribution.

IV. REKYEING STRATEGIES AND SECURE MULTICASTING

The Algorithms that has been used for this paper, will be described very briefly here,

A. Logical Key Hierarchy

In LKH there is a group controller who controls the whole rekeying process for all groups under it. The members are joining in a node as a leaf. These nodes are under a subgroup. Every nodes and members have key encryption keys (KEK). When any join or leave operation takes place GC needs regenerate KEK for every current nodes and members. [7]

In example, if member 16 wants to join in a node, it sends a request to the server. Server checks its authentication key. If the key is ok then GC generate new KEK encrypted with its key. So when member 16 requests for join the associated node H KEK is H_{15}, next node L is $L_{13...16,}$ next node N is $N_{9...15}$ and finally the GC node O is $O_{1...15}$. So all the nodes and members KEK needs to change when member 16 get authentication for join. For this GC generate new KEK and send a rekey message to the every member and nodes of the group. For leaving operation of a member or a subgroup same strategy followed in LKH.

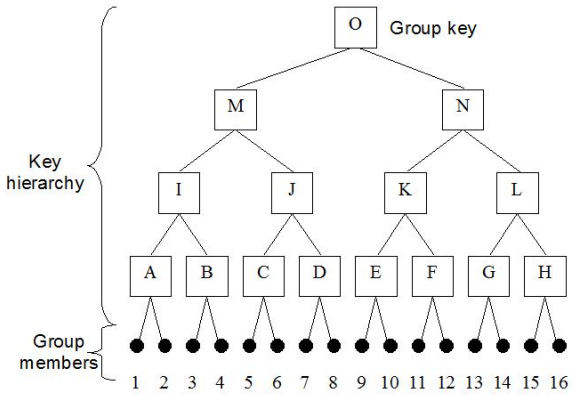

Fig. 1. Diagram of a group using LKH keying system

A. Centralized Flat Table Key Management

Waldvogel at el proposed centralized flat key management (CFKM) [22] in 1999. He introduced to assign the bits of the ID in a flat fashion (Fig. 1). In this protocol each member will have an unique identity code. If there is 2n member in group (including potential), then Key Server(KS) will construct 2n number of key encryption keys (KEKs).Every member will hold odd number of KEKs as it is constructed as (2n-1) it is maintained to give each member an unique id. This protocol gives the id in according to the code

of that member. If, a member is 01101….1. then the KEK will be (0,0) as the first bit is 0.

For second bit-1 KEK will be (1,1). The series it will follow will end in KEK((2n-1),1).

- *Leaving the group*

All the KEKs are should be renounced while a member left. The key server generates new KEKs by (2n-1) and encrypts with other (half) KEKs. Every group member, except the members on the left side, can decrypt rekeying messages. Subsequently, security during data forward is ensured.

TABLE I. SIMPLE KEY ASSIGNMENT FOR FLAT ID (WALDVOGEL AT EL) [22]

	TEK	
ID Bit #0	KEK 0.0	KEK 0.1
ID Bit #0	KEK 1.0	KEK 1.1
ID Bit #0	KEK 2.0	KEK 2.1
ID Bit #0	KEK 3.0	KEK 3.1
	Bit's Value = 0	Bit's Value = 1

- *Joining the group*

In case of join, the new member holds the certain half of KEKs to ensure backward security. KS generates new KEKs

and use one of the KEKs to encrypt. KS generates ((2n-1)) keys. New KEKs are encrypted with new TEK.

B. Security Key distribution

As cryptography is used for enhancing security in group communication access so for exchanging public key Diffie-Hellman [14] algorithm is used in multicast. As it is a one way function so it is very difficult to calculate in reverse order. Here an initiator or Group Controller creates a prime modulus 'P' and a primitive root of the prime modulus 'g' and multicast these two numbers, then all the group members generate their own random private number 'X'. After that the results (R) of DH-computation (g^X mod P) are exchanged by the members. Hence for achieving the key members again computes (R^x mod P) to access the group.

V. Experiment and Performance

The experiments are taken place in Intel corei5 3.2 GHz machine with 8 Gigabyte of RAM running windows. The analogy of the selected rekeying systems is shown in Fig 2 and Fig 3. Theoretical and practical analogy is shown in Fig 2 while relative performance of keying technique with regard to user number is shown in Fig 3.

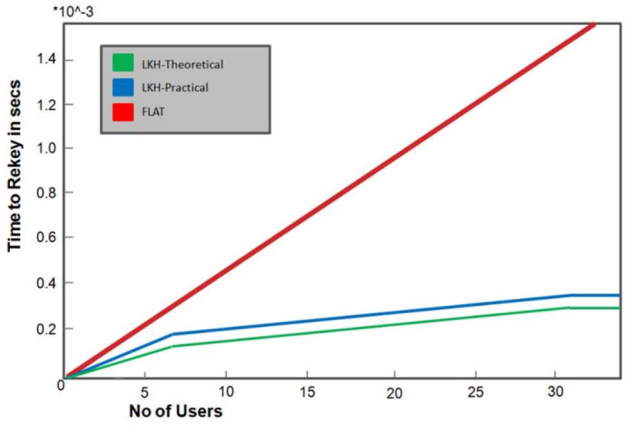

Fig. 2. Theoretical vs. Practical performance of different keying processes

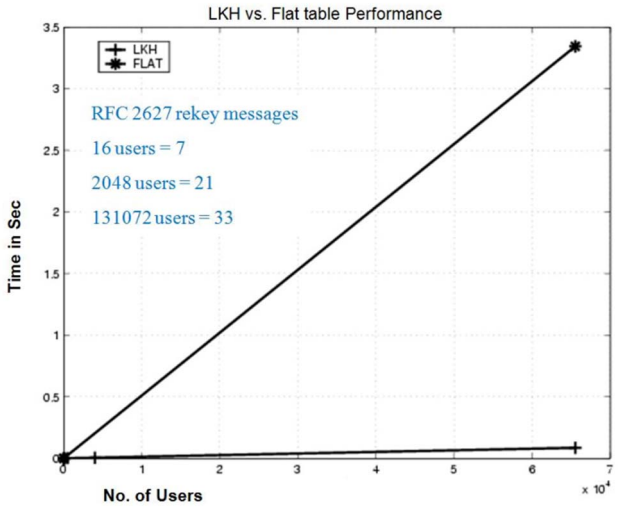

Fig. 3. LKH vs. Flat key performance for specific number of users

First diagram of this section is showing the difference of the delay of LKH (theoretical and practical) with Flat table protocol. Flat table protocol is showing inefficiency with the number of user increment. The red line (Flat) is showing linear growth of delay with the number of user, where LKH is exceptionally efficient in suppressing the delay. Though the LKH-practical is a bit delayed performance with respect to theoretical performance.

Second diagram is showing our experiment result using 0.13 million of users. Where the number of Rekeying messages are 2627. Both of the protocols are showing linear growth. The growth of the Flat table protocol is exceptionally prodigious where LKH show very low rate of delay.

A User Management Service (UMS) has been designed from the scratch. It utilizes simple Graphical User interface (GUI). The front-end is based on open framework like cinder and compatible with any operating system. The backend is java based and has the ability to work in conjunction with many data procurement computer aided tools.

Great importance is being exerted on log keeping and usually it is conducted by sticking to regular expressions to parse data, store it by conventionally available tools. A proposed plan is underway so that a test team can access test data live from the operation without interfering system operation.

Moreover, Electronic Program Guide (EPS) has been developed to visualize current status of users at any given time. The guide shall communicate with the network administrator and update data after certain interval. The data can be relayed to any conventional Display. The work is in next priority queue since our data analysis and keying strategy is still in implementation phase.

VI. Conclusion

Empirical evidence from rigorous data analysis and fact checking has convincing answers for further work. Networks are growing along with users. Data compromise cannot be

tolerated under any perceived condition. Strategic planning and proper execution can save out data integrity not only in our private life but also public facilities that are in peril. Parallel work is required to ensure that the speed does not end up so slow that clients suffer due to lethargic speed. Data integrity must not come at a cost of decreased quality in service. Hence, in house rigorous testing is required with proper automation system; log keeping and analysis of data. Successful implementation of keying will pave way for future development especially in the region of complex data like video and live feed from extra-terrestrial facility.

VII. FUTURE WORK

As stated in section VI, the successful implementation of keying strategy will pave the way of future development in this era. An analogy between the strategies will be put together that shall result in a single strategy for all type of keying strategy like Centralized, decentralized and distributed.

ACKNOWLEDGEMENT

The authors convey appreciation to Md. Firoz Mridha and Md. Habibur Rahman for their constructive suggestion during manuscript preparation.

REFERENCES

[1] Pardoe, Terry D., and Gordon Snyder. *Network Security*. Cengage Learning, 2005.

[2] Rohloff, Kurt Ryan. "SYSTEM AND METHOD FOR MERGING ENCRYPTION DATA WITHOUT SHARING A PRIVATE KEY." U.S. Patent No. 20,150,304,287. 22 Oct. 2015.

[3] Knapp, Eric D., and Joel Thomas Langill. *Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems*. Syngress, 2014.

[4] Harney, Hugh, and Carl Muckenhirn. *Group key management protocol (GKMP) architecture*. No. RFC 2094. 1997.

[5] Wong, Chung Kei, Mohamed Gouda, and Simon S. Lam. "Secure group communications using key graphs." *ACM SIGCOMM Computer Communication Review*. Vol. 28. No. 4. ACM, 1998.

[6] Wallner, Debby, Eric Harder, and Ryan Agee. *Key management for multicast: Issues and architectures*. No. RFC 2627. 1999.

[7] Wong, Chung Kei, Mohamed Gouda, and Simon S. Lam. "Secure group communications using key graphs." *IEEE/ACM transactions on networking* 8.1 (2000): 16-30.

[8] Mittra, Suvo. "Iolus: A framework for scalable secure multicasting." *ACM SIGCOMM Computer Communication Review*. Vol. 27. No. 4. ACM, 1997.

[9] Ballardie, Anthony. "Scalable multicast key distribution." (1996).

[10] Setia, Sanjeev, et al. "Kronos: A scalable group re-keying approach for secure multicast." *Security and Privacy, 2000. SandP 2000. Proceedings. 2000 IEEE Symposium on*. IEEE, 2000.

[11] Dondeti, Lakshminath R., Sarit Mukherjee, and Ashok Samal. "Scalable secure one-to-many group communication using dual encryption." *Computer Communications* 23.17 (2000): 1681-1701.

[12] Cheng, Jiin-Chiou, and Chi-Sung Laih. "Conference key agreement protocol with non-interactive fault-tolerance over broadcast network." *International Journal of Information Security* 8.1 (2009): 37-48.

[13] Huang, Kuo-Hsuan, et al. "A conference key agreement protocol with fault-tolerant capability." *Computer Standards and Interfaces* 31.2 (2009): 401-405.

[14] Diffie, Whitfield, and Martin Hellman. "New directions in cryptography." *IEEE transactions on Information Theory* 22.6 (1976): 644-654.

[15] Kim, Yongdae, Adrian Perrig, and Gene Tsudik. "Tree-based group key agreement." *ACM Transactions on Information and System Security (TISSEC)* 7.1 (2004): 60-96.

[16] Judge, Paul, and Mostala Ammar. "Security issues and solutions in multicast content distribution: A survey." *IEEE network* 17.1 (2003): 30-36.

[17] A. Ballardie, Scalable Multicast Key Distribution, RFC 1949, 1996.

[18] McDaniel, Patrick, Atul Prakash, and Peter Honeyman. "Antigone: A Flexible Framework for Secure Group Communication." *USENIX Security*. 1999.

[19] Schneier, Bruce. "Applied cryptography: protocols." *Algorithms, and Source Code in C* 2 (1996): 216-222.

[20] Kim, Yongdae, Adrian Perrig, and Gene Tsudik. "Simple and fault-tolerant key agreement for dynamic collaborative groups." *Proceedings of the 7th ACM conference on Computer and communications security*. ACM, 2000.

[21] Waldvogel, Marcel, et al. "The VersaKey framework: Versatile group key management." *IEEE Journal on selected areas in communications* 17.9 (1999): 1614-1631.