http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 7 (7): 1081-1084, 2008
ISSN 1812-5638

Group Rekeying Approach for Group Communication

Chunbo Ma and Jun Ao
School of Information and Communication,
Guilin University of Electronic and Technology, Guilin, Guangxi, 541004, People's Republic of China

Abstract: To enhance the performance of the key management in group communication, we present an improved rekeying approach based on the LKH and OFT schemes in this study. The RSA crypto scheme is used to rekey the keys and the performance of the approach is better than LKH and OFT. The proposed rekeying scheme is top-down and meets forward and backward security.

Key words: Group communication, RSA, OFT, rekeying

INTRODUCTION

How to alleviate the communication overload and improve the performance is a key issue in network communication. In some applications, we have to distribute same message to all n group members. A simple approach for achieving this goal is that the sender encrypts the message respectively for each member of the group. However, this method will generate a message O (n) long which is then transmitted to the whole group via multicast. Obviously, the cost of using the simple approach in large groups is very high. Therefore, how to efficiently distribute a message in this scenario and ensure the network security has attracted lots of attention. To due with this problem, broadcast encryption was proposed and many solutions have been presented (Boneh *et al*., 2005; Fiat and Naor, 1993; Ma *et al*., 2007; Dodis and Fazio, 2003).

Many network applications such as audio and video conference, pay TV systems, secure distribution of copyright-protected material and shared instruments need reliable and secure group communication. For group communications, we must not only consider how to effective and safe transmit information, but also should consider how to effectively carry out key management. Generally speaking, distributing the group key to valid members is a complex problem. The group may require that membership changes cause the group key to be refreshed. In other words, a secure group communication should meet forward security and backward security. Forward security is used to prevent a new member from decoding messages exchanged before it joined the group. And the backward security is used to prevent a leaving or evicted group member from accessing the group communication. The key issue to solve this problem is to establish an effective and safe rekeying scheme.

The literature presents us with several different approaches to group key management. We can divide them into following classes: Centralized group key management protocols, decentralized architectures and distributed key management protocols. The Logical Key Hierarchies (LKH) independently presented by Wong *et al*. (1998) and Wallner *et al*. (1998) is typical Centralized group key management method. To date, many solutions are based on LKH with symmetric keys organized in a tree. McGrew and Sherman (1998) presented an efficiency protocol based on One-way Function Tree (OFT). Thereafter, Canetti *et al*. (1999) improved the OFT and designed a One-way Function Chain Tree (OFCT). Additionally, Waldvogel *et al*. (1999) presented the Centralized Flat Table key management protocol. There are some other literatures on group key management (Perrig *et al*., 2001; Li and Sampalli, 2008; Khurana *et al*., 2005).

The logical key hierarchy (LKH) method was proposed by Wallner *et al*. (1998) and Wong *et al*. (1998), respectively. In this approach, a KDC maintains a tree of keys where each node represents a key encryption key. The leaves of the tree correspond to group members and each leaf holds the keys related to the nodes in the path from its leaf node to the root. The key held by the root of the tree is the group key used for encrypting message in group communications. For a balanced tree, each member stores at most $(\log_2 n)+1$ keys, where $(\log_2 n)$ is the height of the tree.

Another optimization of the logical key hierarchy approach is One-way Function Tree (OFT) proposed by McGrew and Sherman (1998). Their scheme reduces the size of the rekeying message from $2(\log_2 n)$ to only $(\log_2 n)$. The approach is bottom-up and a node's KEK is

Corresponding Author: Chunbo Ma, School of Information and Communication, Guilin University of Electronic and Technology, Guilin, Guangxi, 541004, People's Republic of China

generated rather than just attributed. The KEKs held by a node's children are blinded using a one-way function and then mixed together using a mixing function. The result of this mixing function is the KEK held by the node.

Canetti *et al.* (1999) proposed a slightly different approach that achieves the same communication overhead by using a pseudo-random-generator to generate the new KEKs rather than a one-way function. There approach is only used in the scenario of user leaving or evicted. This scheme is known as the one-way function chain tree. The pseudo-random-generator, G(x), doubles the size of its input (x) and there are two functions L(x) and R(x), that represent the left and right halves of the output of G(x), i.e., G(x) = L(x)R(x).

In this study we present an improved approach (RSA-OFT) based on McGrew and Sherman's research via RSA crypto scheme. Our approach not only meets the forward security and backward security, but also has better performance.

BACKGROUND

RSA scheme: The RSA encryption scheme is widely used. It can be described as follows:

- Key Generation Center 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 RSA scheme, p and q should be destroyed.

One way function: We say a 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 ν_A such that for sufficiently large k,

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

THE PROPOSED RSA-OFT SCHEME

As we have mentioned above, the one-way function tree approach proposed by McGrew and Sherman is a bottom-up method. The rekeying is achieved by computing one-way function. In contrast, LKH is a top-down approach. We try to combine the two methods and establish an improved rekeying approach via RSA crypto scheme.

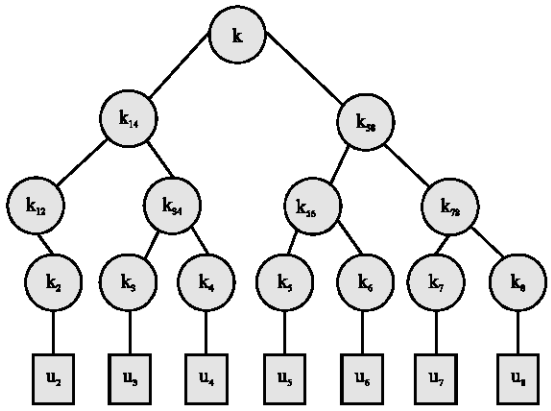

Fig. 1: Key tree

Without any loss of generality, we will describe our approach in a group consisted of eight users. Suppose that there exists a Key Generation Center 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. And then KGC computes the matching public key e to establish a RSA 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. According to the LKH approach, each user preserves the corresponding secret keys. Figure 1 shows relationship among these users.

For example, user u_3 should preserve {k_3, k_{34}, k_{14}, k} and the user u_2 holds {k_2, k_{12}, k_{14}, k}, where k is the group key.

JOIN OPERATION

When a 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. The Join operation consists of following steps.

- KGC chooses a random number $x \in \mathbb{Z}_n^*$ and computes $x^d, x^{d^2}, x^{d^3}, x^{d^4}$
- KGC computes $\{x^{d^4}\}_{k_2}, \{x^{d^3}\}_{k_{34}}, \{x^d\}_{k_{58}}$ and broadcast them in the group
- The user u_2 can decrypt the ciphertext and get x^{e^4} via secret key k_2. Then computes and $x^{d^2 \cdot e} = x^d$. Finally, computes $x^{d^4 \cdot e} = x^{d^3}$, $x^{d^3 \cdot e} = x^{d^2}$ $k'_{12} = H(x^{d^3} \| k_{12})$, $k'_{14} = H(x^{d^2} \| k_{14})$ and $k' = H(x^d \| k)$, where a||b denotes the concatenate of a and b. The keys $\{k_2, k'_{12}, k'_{14}, k'\}$ are the rekeyed keys.

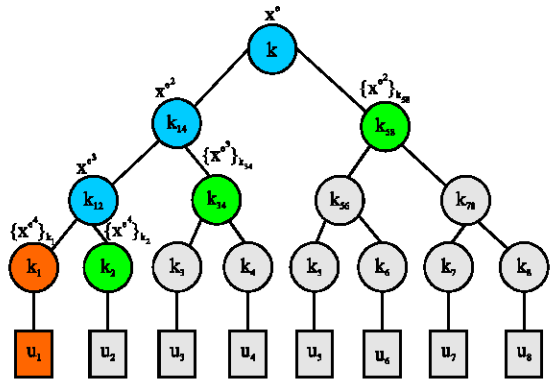

Fig. 2: Join operation

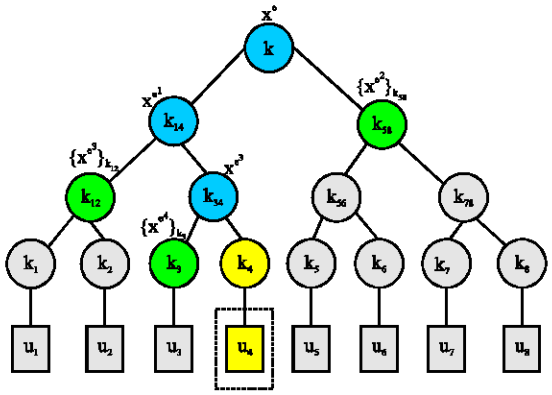

Fig. 3: Leaving operation

- The users u_3 and u_4 obtain x^{d^3} via the secret key k_{38} and then get x^{d^2} and x^d independently. Thereafter, they compute $k'_{14} = H(x^{d^2} \| k_{14})$ and $k' = H(x^d \| k)$, respectively and preserve them as rekeyed keys.
- The users u_5, u_6, u_7, u_8 obtain x^{d^2} via secret key k_{58} and then separately compute $x^{d^2 \cdot e} = x^d$ and $k' = H(x^d \| k)$.
- KGC chooses a random key k_1 for user u_1 and distributes $\{k_1, k_{12}, k_{14}, k\}$ to u_1 as his secret key and then broadcast $\{x^{d^4}\}_{k_1}$.
- The user u_1 obtains x^{d^4} via secret key k_1 and then computes x^{d^3}, x^{d^2}, x^d. Finally, user u_1 computes k'_{12}, k'_{14}, k' and completes the rekeying process.

The operation can be illustrated as Fig. 2, where the keys $\{k, k_{14}, k_{12}\}$ related to user u_1 should be changed.

LEAVING OPERATION

Suppose that the user u_4 will leave the group $U = \{u_1, u_2, L, u_8\}$. To provide backward security, KGC has to update the keys related to user u_4. The leaving operation consists of following steps.

- KGC chooses a random number $x \in \mathbb{Z}_n^*$ and then computes $x^d, x^{d^2}, x^{d^3}, x^{d^4}$
- KGC computes $\{x^{d^3}\}_{k_{12}}, \{x^{d^4}\}_{k_3}$ and $\{x^{d^2}\}_{k_{58}}$ and then broadcasts them
- The users u_1, u_2 obtain x^{d^3} via secret key k_{12} and then compute $x^{d^3 \cdot e} = x^{d^2}$ and $x^{d^2 \cdot e} = x^d$. Thereafter, they separately compute $k'_{14} = H(x^{d^2} \| k_{14})$ and $k' = H(x^d \| k)$. Here, $\{k'_{14}, k'\}$ are the updated keys.
- The user u_3 decrypts and obtains x^{d^4} via secret key k_3 and then computes x^{d^3}, x^{d^2}, x^d. Subsequently, he computes $k'_{34} = H(x^{d^3} \| k_{34})$, $k'_{14} = H(x^{d^2} \| k_{14})$ and $k' = H(x^d \| k)$ as the updated keys.
- The users u_5, L, u_8 decrypt and obtain x^{d^2} via secret key k_{58} and then compute $x^{d^2 \cdot e} = x^d$. Finally they compute $k' = H(x^d \| k)$, respectively and finish the rekeying process

The operation can be illustrated as Fig. 3, where the key $\{k_{34}, k_{14}, k\}$ related to user u_4 should be changed.

DISCUSSION

As we have mentioned, the aim of the rekeying scheme is to provide system forward and backward security. In the circumstance of user joining or leaving, we completely update the keys related to the changed user. With the approach that we have described earlier, we say the proposed rekeying meets the requirement of the security.

Present approach, which just as the LKH, is top-down. In other words, the messages were generated and distributed by KGC. The virtue of the method is that it is secure against some kinds of forgery attack in key updating. Furthermore, since only KGC holds the secret key e, nobody beside him can compute $x^d, x^{d^2}, x^{d^3}, x^{d^4}$. Additionally, the users hold the public key e and have the ability to execute reverse operation, i.e., deducing x^{d^3}, x^{d^2}, and x^d from x^{d^4}.

EFFICIENCY

We compare present approach with LKH and OFT. Table 1 evaluate the performance of our proposed approach on four aspects: Secrecy, Message Length, Storage and Style. The notation used in Table 1 and 2 is described as follows:

n = No. of member in the group
a = Degree of the tree
d = Height of the tree (for a balanced binary tree $d = \log_2 n$)
H = Hash function
X = Xor operation
E = Encryption operation
D = Decryption operation
K = Size of a key in bits

Table 1: Comparison table of group key management protocols

	Secrecy		Message: Join		Message: Leave	Storage		
Scheme	Back	Fore	Multicast	Unicast		KDC	Member	Style
LKH	Y	Y	(2d-1)K	(d+1)K	2dK	(2n-1)K	(d+1)K	Top-down
OFT	Y	Y	(d+1)K	(d+1)K	(d+1)K	(2n-1)K	(d+1)K	Bottom-up
RSA-OFT	Y	Y	(d+1)K	(d+1)K	(d+1)K	(2n-1)K	(d+1)K	Top-down

Table 2: Comparison of computation cost

	Adding a member		Evicting a member	
Scheme	GC	Member	GC	Member
LKH	3dE	(d-1)D	2dE	dD
OFT	d(H+X)+(2d+1)E	(d+1)D+d(H+X)	d(E+H+X)	D+d(H+X)
RSA-OFT	(d+1)E	(d+1)D+dH	dE	dD+dH

From the Table 1 we can see that present approach RSA-OFT has the same style with LKH, i.e., top-down. To the aspect of message length transmitted during rekeying, RSA-OFT is similar to OFT and shorter than LKH. From the Table 2 we can see that performance of RSA-OFT is much better than LKH and OFT. Additionally, note that KGC can choose a random number $x \in \mathbb{Z}_n^*$ and pre-compute $x^d, x^{d^2}, x^{d^3}, x^{d^4}$, since it is nothing to do with user's private key. This step further improves the efficiency of the RSA-OFT.

CONCLUSION

We present an improved group rekeying approach from RSA crypto scheme in this paper and describe the approach in balance binary tree composed of eight users. Similarly with LKH method, RSA-OFT is top-down and meets forward and backward security. According to the comparison described in Table 1 and 2, our approach has better performance than LKH and OFT.

ACKNOWLEDGMENT

This study is supported by the National Natural Science Foundation of China (60862001).

REFERENCES

Boneh, D., C. Gentry and B. Waters, 2005. Collusion Resistant Broadcast Encryption With Short Ciphertexts and Private Keys. In: Advances in Cryptology-CRYPTO 2005, LNCS 3621, Shoup, V. (Ed.). Springer-Verlag, Berlin, Germany. ISBN: 0302-9743 (Print) 1611-3349 (Online), pp: 258-275.

Canetti, R., J. Garay, G. Itkis, D. Micciancio, M. Naor and B. Pinkas, 1999. Multicast security: A taxonomy and efficient constructions. Proceedings of IEEE Infocom. 18th Annual Joint Conference of the IEEE Computer and Communications Societies. March 23, 1999, IEEE Press pp: 708-716.

Dodis, Y. and N. Fazio, 2003. Public key trace and revoke scheme secure against adaptive chosen ciphertext attack. Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography: Public Key Cryptography. Jan 6-8, Springer-Verlag, Berlin, Germany.-pp: 100-115.

Fiat, A. and M. Naor, 1994. Broadcast encryption. Proceedings of the 13th annual international cryptology conference on Advances in cryptology, Jan 1994, Springer-Verlag, New York, USA., pp: 480-491.

Khurana, H., R. Bonilla, A. Slagell, R. Afandi, H.S. Hahm and J. Basney, 2005. Scalable group key management with partially trusted controllers. Proceedings of ICN, April 17, 2005, Springer-Verlag, Berlin, Germany, pp: 662-672.

Li, D. and S. Sampalli, 2008. A hybrid group key management protocol for reliable and authenticated rekeying. Int. J. Network Security, 6: 270-281.

Ma, C.B., J. Ao and J.H. Li, 2007. Broadcast group-oriented encryption secure against chosen ciphertext attack. J. Syst. Eng. Elect., 18: 811-817.

McGrew, D.A. and A.T. Sherman, 1998. Key establishment in large dynamic groups using one-way function trees. Technical Report No. 0755, TIS Labs at Network Associates, Inc., Glenwood, MD.

Perrig, A., D. Song and J.D. Tygar, 2001. ELK, a new protocol for efficient large-group key distribution. IEEE Symposium on Security and Privacy. 14-16 May, IEEE Computer Society, Washington, DC., USA., pp: 247-262.

Waldvogel, M., G. Caronni, D. Sun, N. Weiler and B. Plattner, 1999. The versakey framework: Versatile group key management. IEEE J. Selected Areas Commun., 17: 1614-1631.

Wallner, D.M., E.J. Harder and R.C. Agee, 1998. Key Management for Multicast: Issues and Architectures. Internet Draft (work in progress), draft-wallner-key-arch-01.txt, Internet Engineering Task Force.

Wong, C.K., M.G. Gouda and S.S. Lam, 1998. Secure group communications using key graphs. Proceedings of the ACM. SIGCOMM. 4 Sep., ACM Press, New York, USA., pp: 68-79.