Computers & Security (2004) ■■, ■–■

ELSEVIER

Computers & Security

www.elsevier.com/locate/cose

Improvement on the flexible tree-based key management framework

Min-Shiang Hwang[a,*], Jung-Wen Lo[b,c], Chia-Hsin Liu[d]

[a]Department of Management Information System, National Chung Hsing University, 250 Kuo Kuang Road, Taichung 402, Taiwan, ROC
[b]Department of Information Management, National Taichung Institute of Technology, 129 Sec. 3, San-min Rd., Taichung 404, Taiwan, ROC
[c]Department of Computer Science, National Chung Hsing University, 250 Kuo Kuang Road, Taichung 402, Taiwan, ROC
[d]Graduate Institute of Networking and Communication Engineering, Chao Yang University of Technology, 168 Gifeng E. Rd., Wufeng, Taichung County 413, Taiwan, ROC

Received 28 April 2004; revised 2 September 2004; accepted 3 September 2004

KEYWORDS
Key management;
Multiple content distribution systems;
Tree-based

Abstract Matsuzaki et al. had proposed a flexible tree-based key management framework for a terminal to connect with multiple content distribution systems (CDSs) using a public bulletin board. In their scheme, the key management center constructs the public bulletin board by utilizing symmetric cryptosystem to protect terminal node keys. On the other hand, the terminals can obtain its node keys by decrypting the cipher which is posted on the public bulletin board of CDS. However, this method is not efficient for a large group which has a number of terminals. When a terminal changes its membership, the key managerial center needs a large amount of computation to structure the public bulletin, and the terminal cannot efficiently compute its node keys from the large pubic bulletin board. In this paper, we propose an improved scheme to structure a public bulletin board efficiently by the key management center of CDS. The improved scheme is capable of a large group of terminals and ensures that low powered equipment can efficiently obtain their node key.

* Corresponding author. Tel.: +886 422855401; fax: +886 423742337.
E-mail address: mshwang@nchu.edu.tw (M.-S. Hwang).

0167-4048/$ - see front matter © 2004 Published by Elsevier Ltd.
doi:10.1016/j.cose.2004.09.009

Introduction

In a new service environment, home users increasingly use CATV cables, radio broadcasts, and the Internet to digitally access, store, and play music, movies, or other kinds of entertainment in their lives. Recently, the most widely used technique is transferring digital data over the Internet, such as Web TV or Web movies. The providers offer high-value digital content, such as software or multimedia data distributes contents, in a public channel, and prior payment for the content cannot be bypassed. One of the basic ideas is distributing the encrypted digital content through an insecure channel. For the purpose of distributing security and performance improvement, the authorized members of a group can share a symmetric group key so that they can decrypt the group communication content.

In a realistic world, one of useful applications is the network conferences of a company. Every conference has its exclusive conference key to protect the communicative contents in the enterprise network. For example, every team member has his private key for obtaining the team conference key. He uses the team conference key to decrypt the contents of the communication and to encrypt the information he wants to share during the conference. A person who does not belong to this team cannot join the conference because he does not have the correct conference key to encrypt or decrypt the information. Another useful application is to protect the documents of a company. The documents can be encrypted with a company secret key and put on a public network for employees' use. Every employee has his private key for obtaining the company secret key, and the employee can securely retrieve the documents of his company anywhere. Therefore, if there is an efficient key management mechanism, the mechanism could be widely applied.

Several approaches with the tree-based key management have been proposed (Chang et al., 1992; Hwang 1999a; Lin et al., 2003; Shen and Chen, 2002). The logical key hierarchy was proposed by Wong et al. (2000) and the binary key tree management was proposed by Wallner et al. (1998) to reduce complexity of key management of a group. For a large group with frequent membership changed, the cost of re-keying the group is raised. Therefore, some reducible cost schemes had been proposed for achieving the best possible performance (Banerjee and Bhattacharjee, 2002; Chang et al., 1999; Chen and Chung, 2002; Chien and Jan, 2003; Hwang, 1999b; Lin et al., 2004; Snoeyink et al., 2001; Waldvogel et al., 1999; Zhong, 2002). In most cases, they only focus on the scheme which has a fixed management center in CDS. Matsuzaki et al. (2003) proposed a scheme to construct a key management framework by combining various key management schemes of CDS. A terminal which plays a home server can connect with multiple CDSs. By the public bulletin board, the terminal can compute its root key that is used to decrypt an encrypted digital content. However, the above schemes are not efficient in structuring the public board, and also not suitable for low powered equipment to obtain node keys.

In this paper, we propose an improved method to match the requirement of low powered equipment. The improved scheme can efficiently construct the public bulletin board and the terminal also can efficiently compute its node keys. Furthermore, our scheme has less calculating operations than Matsuzaki's in the framework.

The remainder of this paper is organized as follows. In next section, we briefly review the public bulletin board in framework and show why Matsuzaki et al.'s scheme is not efficient enough. Then, the improved scheme is presented. Further, we analyze the security and efficiency of our scheme. Finally, we have our conclusion in last section.

Review of Matsuzaki et al.'s framework

In this section, there are two mechanisms in the CDS_c. One is KMA (Key Management Authority) which generates $\mathrm{UK}_i^c = H(\mathrm{CK}_i \parallel c)$ for terminal i where c is an identity of the CDS_c, CK_i is the secret key of terminal i and $H(\cdot)$ is a secure one-way function. The other mechanism is KMC (Key Management Center) which determines the tree structure and calculates all node keys on the public bulletin board. Initially, KMA computes all UK_i^c and transmits them to KMC in a secure channel. The KMC uses the all UK_i^c to structure the public bulletin board, and later every terminal can obtain its node keys from the board. The following are steps for KMC building the public bulletin board.

Step 1. Determine the inner node keys K_j^c according to the tree structure in Fig. 1, where j is an inner node number.
Layer 0: $K_0^c = H(K_1^c)$,
Layer 1: $K_1^c = H(K_3^c)$, $K_2^c = H(K_5^c)$,
Layer 2: $K_3^c = H(\mathrm{UK}_0^c)$, $K_4^c = H(\mathrm{UK}_2^c)$, $K_5^c = H(\mathrm{UK}_4^c)$, $K_6^c = H(\mathrm{UK}_6^c)$.

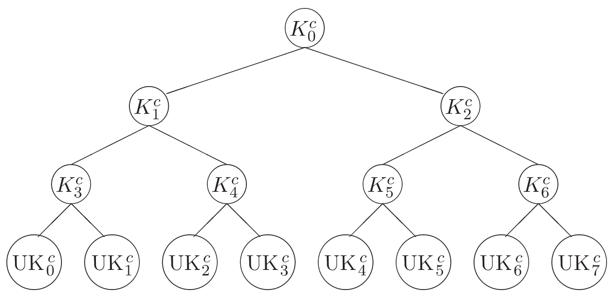

Figure 1 An example of a complete binary tree.

Step 2. Use the K_j^c to generate and publish the encrypted messages on the public bulletin board as follows:
Layer 0: $E(K_2^c, K_0^c)$,
Layer 1: $E(K_4^c, K_1^c)$, $E(K_6^c, K_2^c)$,
Layer 2: $E(\mathrm{UK}_1^c, K_3^c)$, $E(\mathrm{UK}_3^c, K_4^c)$, $E(\mathrm{UK}_5^c, K_5^c)$, $E(\mathrm{UK}_7^c, K_6^c)$.

The terminal i calculates its inner node key from the information of the public bulletin board according to its location on the complete binary tree. If the terminal i is a left child of its upper inner node, the terminal i computes $H(\cdot)$ function to obtain the upper inner node key K_j^c. Otherwise, the terminal i decrypts the cipher which is posted on the public bulletin board to obtain the key K_j^c. From this located leaf node to the upper layer inner nodes, the terminal i uses the same method until it obtains its root node key K_0^c.

For a large group of the CDS, most of the right terminals must spend more computing time to calculate the root key. Because of continuous decrypting the cipher in a high depth tree structure, the most right terminals cannot efficiently get the root key. Especially, the electric appliances which have a low computation ability will spend more time to obtain the root key. In order to access the node keys quickly, we improved Matsuzaki et al.'s scheme to suit low computing equipment.

The proposed scheme

In this section, we propose an improved method to structure the public bulletin board. Our scheme of the public bulletin board is also composed of three stages: initiative stage, structuring public bulletin board stage and obtaining keys stage.

In the initiative stage, the scheme and notations $\{i, \mathrm{CK}_i, c, \mathrm{UK}_i, j, H(\cdot)\}$ are the same as those in Matsuzaki et al.'s scheme. In structuring the public bulletin board stage, we utilize XOR(S, S') function that computes data S and S' in order to conceal secret information. Our proposal consists of the following steps:

Step 1: Determine the inner node keys K_j^c according to the tree structure in Fig. 1.
Layer 0: $K_0^c = H(K_1^c)$,
Layer 1: $K_1^c = H(K_3^c)$ and $K_2^c = H(K_5^c)$,
Layer 2: $K_3^c = H(\mathrm{UK}_0^c)$, $K_4^c = H(\mathrm{UK}_2^c)$, $K_5^c = H(\mathrm{UK}_4^c)$, and $K_6^c = H(\mathrm{UK}_6^c)$.

Step 2: Use the K_j^c and XOR operation to generate and publish the information on the public bulletin board.
Layer 0: $\mathrm{XOR}(K_0^c, H(K_2^c))$,
Layer 1: $\mathrm{XOR}(K_1^c, H(K_4^c))$ and $\mathrm{XOR}(K_2^c, H(K_6^c))$,
Layer 2: $\mathrm{XOR}(K_3^c, H(\mathrm{UK}_1^c))$, $\mathrm{XOR}(K_4^c, H(\mathrm{UK}_3^c))$, $\mathrm{XOR}(K_5^c, H(\mathrm{UK}_5^c))$, and $\mathrm{XOR}(K_6^c, H(\mathrm{UK}_7^c))$.

In obtaining the keys stage, a terminal i calculates its inner node keys according to its location in the tree. Hence, a terminal i must know its location of leaves in the tree. The terminal i can question its related information from the public bulletin board. If the terminal i is the left leaf of its upper inner node keys, the terminal i computes $H(\cdot)$ to obtain the upper node key K_j^c. Otherwise, the terminal i is the right leaf of its upper inner node so it calculates XOR operation to get the upper node key K_j^c. Until having the root node key, the terminal i executes $H(\cdot)$ or XOR operation which depends on its location on left or right side of the upper inner node. For example, terminal 3 has the leaf key UK_3^c in CDS_c, so it obtains upper K_4^c by computing $\mathrm{XOR}(H(\mathrm{UK}_3^c), \mathrm{XOR}(H(K_4^c), \mathrm{XOR}(H(\mathrm{UK}_3^c))))$.

Next, terminal 3 obtains K_1^c by computing $\mathrm{XOR}(H(K_4^c), \mathrm{XOR}(K_1^c, H(K_4^c)))$ because its location is in the right child of the position of inner node key K_1^c in this moment. Finally, terminal 3 obtains K_0^c by computing $H(K_1^c)$ because it is the left child of the inner node key K_0^c. Terminal 3 can obtain K_4^c, K_1^c, and K_0^c from its leaf to root node i according to the following pseudo code:

```
node_key = UK_i
WHILE node_key < > root_node_key
IF Position in Parent.RightChild THEN
        node_key = XOR(node_key, Position.
        CipherInformation)
ELSE
        Node_key = H(node_key)
END WHILE
```

Efficiency and security analysis

In this section, the efficiency and security of the proposed method are analyzed. Matsuzaki et al.'s scheme uses encryption algorithm to obscure the information of inner node keys in order to reach the security requirement. In generalization, an encrypted method must possess a complex algorithm to obscure a plaintext message, such as DES (Data Encryption Standard) cryptosystem which includes steps of the initial permutation, the key transformation, the expansion permutation, the S-Box substitution, the P-Box permutation and the final permutation in Menezes et al. (1996). Our scheme uses the XOR operator to replace the encrypted cryptosystem, because it is more efficient.

Compared with Matsuzaki et al.'s scheme, the number of computing times of the most right terminal *i* to obtain the root key depends on the depth *l* of the complete binary tree. In our scheme, terminal *i* can efficiently compute $l \times \mathrm{XOR}(S, S')$ for the root key. On the other hand, Matsuzaki et al.'s scheme must compute $l \times E(K, M)$ for the root key. We have performed an experiment by using a PC (2.0 GHz Pentium-4 CPU with 512 MB RAM) to test the software performance. The simulation is performed with a simulator using Java language in the UNIX operating system. In Table 1, our scheme uses the XOR operation and Matsuzaki et al.'s scheme uses a DES cryptosystem for computing the root key at a depth of 5000, 10 000, and 20 000 layers of a complete binary tree. Evidently, our scheme has a higher performance.

Next is the security analysis for our scheme. The possible attacks for breaking our scheme is an adversary from an internal or external environment. In the remainder of this section, several attacks will be raised and we demonstrate how to fight against these attacks.

Attack 1: A non-attending terminal $e \notin \mathrm{CDS}_c$ tries to obtain the root key K_0^c from the public bulletin board.

Analysis of Attack 1: A non-attending terminal $e \notin \mathrm{CDS}_c$ must have one of $\{\mathrm{UK}_0^c, \ldots, \mathrm{UK}_i^c\}$ to compute the inner node key. However, to form the valid value $\mathrm{UK}_c^i = H(\mathrm{CK}_i \parallel c)$ needs a secret key CK_i which was securely transferred from the KMA. Hence, if any non-attending terminal *e* wants to get the root node key, it must inverse the one-way function. Basically, it is hard to derive *x* from the $H(x)$ and also difficult to find another message x' satisfying $H(x) = H(x')$. For this reason, it is impossible for any outside adversary to reveal the inner node key.

Attack 2: An attending terminal $y \in \mathrm{CDS}_c$ tries to obtain the brother terminal *z*'s CK_z from the public bulletin board for free use of digital content.

Analysis of Attack 2: If an attending terminal $y \in \mathrm{CDS}_c$ tries to steal the brother terminal *z*'s CK_z^c for getting free digit content, it equally faces solving a one-way function question as we mentioned in Attack 1. Hence, terminal *y* cannot forge the valid UK_z^c without knowing CK_z^c.

Attack 3: A terminal $y \in \mathrm{CDS}_c$ should not obtain the node keys only if it uses the associated terminal leaf along the path to its root.

Analysis of Attack 3: A terminal $y \in \mathrm{CDS}_c$ efficiently obtains the node keys from its associated terminal leaf along the path to the root. If terminal *y* wants to obtain the other node keys, to calculate the inversion of a one-way function is difficult. Hence, terminal *y* cannot efficiently get the other node keys and use the node key to forge the illegitimate information.

Table 1 Comparison of executing time between Matsuzaki et al.'s scheme and our scheme

Depth of tree (layers)	Matsuzaki et al.'s scheme	Our scheme
5000	1509 ms	15 ms
10 000	2078 ms	16 ms
20 000	3657 ms	18 ms

Conclusions

We have proposed an improved flexible tree-based key management framework. When a terminal connects to the designated CDS, the terminal can efficiently compute its inner node keys from a public bulletin board, and it does not need to have a high computing ability. In other words, it could be used in mostly low powered equipment, such as being used in a wireless environment. Simultaneously, the management center of CDS can also efficiently structure a public bulletin board with less computation. Moreover, our scheme is secure against the adversary attacks from the external and internal system.

References

Banerjee Suman, Bhattacharjee Bobby. Scalable secure group communication over IP multicast. IEEE Journal on Selected Areas in Communications 2002;20(8):1511—27.

Chang CC, Hwang RJ, Wu TC. A cryptographic key management scheme for access control in a tree structure. In: Proceeding of ICS 1992. Taiwan, ROC; 1992. p. 174—8.

Chang I, Engel R, Kandlur D, Pendarakis D, Saha D. Key management for secure Internet multicast using Boolean function minimization techniques. In: INFOCOM' 99. New York; March 1999. p. 689–98.

Chen Tzer-Shyong, Chung Yu-Fang. Hierarchical access control based on chinese remainder theorem and symmetric algorithm. Computers & Security 2002;21(6):565–70.

Chien Hung-Yu, Jan Jinn-Ke. New hierarchical assignment without public key cryptography. Computers & Security 2003;22(6):523–6.

Hwang Min-Shiang. An improvement of a dynamic cryptographic key assignment schemes in a tree hierarchy. Computers & Mathematics with Applications 1999a;37(3):19–22.

Hwang Min-Shiang. An improvement of novel cryptographic key assignment scheme for dynamic access control in a hierarchy. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 1999b;82-A(3): 548–50.

Lin Iuon-Chang, Hwang Min-Shiang, Chang Chin-Chen. A new key assignment scheme for enforcing complicated access control policies in hierarchy. Future Generation Computer Systems 2003;19(4):457–62.

Lin Iuon-Chang, Ou Hsia-Hung, Hwang Min-Shiang. Efficient access control and key management schemes for mobile agents. Computer Standards & Interfaces 2004;26(5):423–33.

Matsuzaki N, Nakano T, Matsumoto T. A flexible tree-based key management framework. IEICE Transactions on Fundamentals 2003;E86-A(1):129–35.

Menezes Alfred J, van Oorschot Paul C, Vanstone Scott A. Handbook of applied cryptography: CRC Press; 1996.

Shen Victor RL, Chen Tzer-Shyong. A novel key management scheme based on discrete logrithms and polynomial interpolations. Computers & Security 2002;21(2):164–71.

Snoeyink J, Suri S, Varghese G. A lower bound for multicast key distribution. In: IEEE INFOCOM'2001; April 2001.

Waldvogel M, Caronni G, Sun D, Weiler N, Plattner B. The versakey framework: versatile group key management. IEEE Journal on Selected Areas in Communications 1999;17(9): 1614–31.

Wallner DM, Harder E, Agee RC. Key management for multicast: issues and architectures. Technical Report RFC 2627; September 1998.

Wong CK, Gouda MG, Lam SS. Secure group communications using key graphs. IEEE/ACM Transactions on Networking 2000;8(1):16–30.

Zhong Sheng. A pratical key management scheme for access control in a user hierarchy. Computers & Security 2002; 21(8):750–9.

Min-Shiang Hwang was born on August 27, 1960 in Tainan, Taiwan, Republic of China (ROC). He received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, ROC, in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984 to 1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician in the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a chairman of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999–2002. He was a professor and chairman of the Graduate Institute of Networking and Communications, CYUT, during 2002–2003. He obtained the 1997, 1998, 1999, 2000, and 2001 Distinguished Research Awards of the National Science Council of the Republic of China. He is currently a professor of the Department of Management Information System, National Chung Hsing University, Taiwan, ROC. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang has published 100 articles on the above research fields in international journals.

Jung-Wen Lo was born on February 21, 1964 in Taiwan. He received the B.S. degree in Information and Computer Engineering in 1987 from Chung Yuan Christian University, Chung-Li, Taiwan and the M.S. degree in Computer Science & Information Systems in 1994 from Texas A& M University at Commerce, Texas, USA. He is working for his Ph.D. program in the Department of Computer Science at National Chung Hsing University, Taichung, Taiwan. He was an Associate Engineer in Product Development Department of Institute for Information Industry from 1994 to 1996. Since August 1998, he has been an Instructor of the Department of Information Management at National Taichung Institute of Technology, Taichung, Taiwan. His research interests include electronic commerce, computer cryptography and computer networks.

Chia-Hsin Liu was born in Miao-Li, Taiwan, Republic of China, on August 1, 1979. He received the B.S. and M.S. degrees in Department of Information Management and Graduate Institute of Networking and Communication Engineering from Chaoyang University of Technology, Taichung, Taiwan, in 2002 and 2004, respectively. He is currently a System Engineer in OmniWise International Inc. His research interests include cryptography, information security, database security and network security.