An Improved Key-Management Scheme for Hierarchical Access Control

Wan-Yu Chao[1], Cheng-Yi Tsai[2], Min-Shiang Hwang[2,3]
(Corresponding author: Min-Shiang Hwang)

Department of Management Information Systems, National Chung Hsing University[1]
Department of Computer Science and Information Engineering, Asia University[2]
No. 500, Lioufeng Raod, Wufeng Shiang, Taichung 41354, Taiwan
Department of Medical Research, China Medical University Hospital, China Medical University[3]
No.91, Hsueh-Shih Road, Taichung 40402, Taiwan
(Email: mshwang@asia.edu.tw)

(Received Aug. 27, 2016; revised and accepted Oct. 9, 2016)

Abstract

Now, most institutions share the data through the Internet. With the rapid development of the Internet and the cloud storage, data-sharing becomes so easy that the data was stolen or destroyed easier than before. Therefore, accessing data should strictly control to avoid unauthorized access. In this paper, we propose the more efficient key management scheme for hierarchical access control than Odelu et al.'s scheme.

Keywords: Access Control; Hierarchical Access Control; Key Management

1 Introduction

The user hierarchy system is represented by a set of disjoint security class SC_i ($i = 1, 2, \cdots, N$) and a partially ordered set (POSET) $(SC, \leq)$, where $\leq$ (i.e. $<$ or $=$) is a binary partially ordered relationship in a user set [10, 11, 12]. Generally, this hierarchy system is managed by a trusted central authority (CA). CA distributes the secret key (SK_i) to each security class (SC_i) and announces the corresponding public information in a public domain. $SC_j < SC_i$ denotes that SC_i is a predecessor of SC_j, and SC_j is a successor of SC_i; and a predecessor security class can derive its successors' secret keys (SK_j). We take a user hierarchy system as an example in Figure 1 [5, 7, 13]. When $SC_2 \leq SC_1$, we can know that SC_2 is a successor of SC_1. If SC_2 encrypts files using its secret key (SK_2), SC_1 derives SC_2's secret key to access the files which are encrypted by SC_2.

In 1983, Akl and Taylor [1] proposed the cryptographic solution to the problem of access control, the first concept of the cryptographic key assignment scheme in an arbitrary POSET hierarchy. A lot of related works have been proposed to solve access control problems [2, 3, 4, 6, 8, 9, 14, 15, 17, 18, 19]. In 2013, Odelu et al. [16] proposed an efficient and secure access control scheme only based on symmetric-key cryptosystems and one-way hash functions.

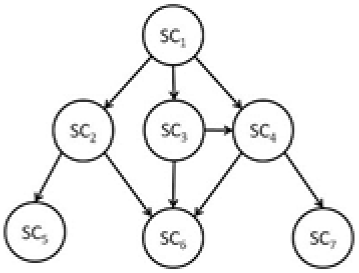

Figure 1: An example of user hierarchy system [16]

2 Review of Odelu et al.'s Schemes

In this section, we briefly introduce Odelu et al.'s scheme [16]. In Table 1, we show the notations' meaning in this paper. We just describe two phases in this scheme: key generation phase and key derivation phase.

2.1 Key Generation Phase

Firstly, CA builds the hierarchical structure for controlling access among the security classes. Let two security classes be SC_i, $SC_j \in SC$ and $SC_j \leq SC_i$ ($1 \leq i \leq N$ and $1 \leq j \leq N$).

Step 1: CA chooses $(H(\cdot),\ \Omega)$ and publishes them. Then CA selects randomly its own secret key (k_{CA})

Table 1: Notations' meanings used in this paper

Notation	Meaning
$H(\cdot)$	A one-way hash function
$E_k(\cdot)/D_k(\cdot)$	AES encryption/decryption with key k
Ω	A symmetric-key cryptosystem
$\|\|$	Concatenation
$\oplus$	Exclusive-OR
CA	The trusted central authority
ID_{CA}	An identity of CA
$ID_(SC_i)$	An identity of SC_i
T_{SHA1}	Time for hashing 512 bits message block using SHA-1
T_{AES}	Time for encrypting/decrypting 128 bits message block using AES with a 128 bits key.

and also calculates its own private key $K = H(ID_{CA}||k_{CA})$.

Step 2: CA selects randomly the secret key (SK_i), and sub-secret key (d_i) for each security class ($SC_i, 1 \leq i \leq N$) in the hierarchical structure.

Step 3: For each security class SC_i, CA computes the encryption key $EK_i = H(ID_{CA}||k_{CA}||d_i)$ and the private key $K_i = H(ID_{CA}||d_i)$.

Step 4: CA computes the following public parameters:

1) For each security class $SC_j \leq SC_i$, computes $M_{ij} = E_{EK_i}(SK_j)$.
2) For SC_i, computes $R_i = E_{K_i}(EK_i)$ and the signature $Sign_i = H(ID_{CA}||SK_i)$.
3) CA encrypts K_i as $S_i = E_K(K_i)$.

Step 5: CA finally sends d_i to each SC_i via the secure channel.

2.2 Key Derivation Phase

This phase introduces how the security class (SC_i) derives the corresponding successor security classes' (SC_j) secret key (SK_j) from the public parameters available in the public domain.

Step 1: SC_i uses its own sub-secret key to compute its private key, $K_i = H(ID_{CA}||d_i)$ and then decrypts R_i for the encryption key, $EK_i = D_(K_i)(R_i)$.

Step 2: SC_i decrypts M_{ij} to obtain the secret keys $SK_j = D_(EK_i)(M_{ij})$.

Step 3: SC_i computes the signature $Sign_j^* = H(ID_{CA}||SK_j)$ and then verifies whether $Sign_j^* = Sign_j$ or not. If yes, the derived secret key (SK_j) is legitimate.

3 The Proposed Scheme

We propose our scheme based on a hybrid cryptosystem utilizing a symmetric-key cryptosystem and a one-way hash function. At First, we present the key generation phase and key derivation phase. Finally, we show the dynamic key management.

3.1 Key Generation Phase

Like Odelu et al.'s scheme, CA builds the hierarchical structure for access control. Let two security classes be SC_i, $SC_j \in SC$ and $SC_j \leq SC_i$ ($1 \leq i \leq N$ and $1 \leq j \leq N$).

Step 1: CA selects $(H(\cdot), \Omega)$ and declares them publicly. Then CA chooses its own secret key (k_{CA}) randomly.

Step 2: CA selects randomly the secret key (SK_i), sub-secret key (d_i) for each security class (SC_i, $1 \leq i \leq N$) in the hierarchical structure.

Step 3: CA calculates the encryption key (EK_i) for each security class as $EK_i = H(k_{CA}||ID_(SC_i))$.

Step 4: CA computes the public parameters (M_{ij}, R_i) for each security class as $M_{ij} = E_(EK_i)(SK_j \oplus ID_{CA})$, and $R_i = E_(d_i)(EK_i)$.

Step 5: At last, CA sends d_i to each SC_i securely and clears all keys (SK_i, d_i, EK_i).

3.2 Key Derivation Phase

For $SC_j \leq SC_i$, we show how the security class (SC_i) derives the secret keys (SK_j) of all its successors (SC_j) and its own secret key from the public parameters, and verifies that secret key is legitimate.

Step 1: SC_i uses its sub-secret key to decrypt R_i as $EK_i = D_(d_i)(R_i)$.

Step 2: SC_i decrypts M_{ij} to obtain $SK_j \oplus ID_{CA} = D_(EK_i)(M_{ij})$.

Table 2: The storage space requirement in the hierarchy shown in Figure 1

CA (Private Domain)	Public Domain		SC_i (Private Domain)	
k_{CA}	R_1	$M_{11}, M_{12}, M_{13}, M_{14}, M_{15}, M_{16}, M_{17}$	d_1	SC_1
	R_2	M_{22}, M_{25}, M_{26}	d_2	SC_2
	R_3	$M_{33}, M_{34}, M_{36}, M_{37}$	d_3	SC_3
	R_4	M_{44}, M_{46}, M_{47}	d_4	SC_4
	R_5	M_{55}	d_5	SC_5
	R_6	M_{66}	d_6	SC_6
	R_7	M_{77}	d_7	SC_7

Step 3: If M_{ij} does not change by the attacker, we can get $SK_j = (SK_j \oplus ID_{CA}) \oplus ID_{CA}$. If not, SC_i will notify CA that M_{ij} is wrong.

3.3 Dynamic Key Management

3.3.1 Insert a New Security Class into the Existing Hierarchical Structure

Assume that a security class (SC_x) with the relationships $SC_j \leq SC_x \leq SC_i$ was inserted into an existing hierarchical structure. CA executes the following steps to manage the accessing priority:

Step 1: CA randomly selects a secret key (SK_x), a subsecret key (d_x) for the security class (SC_x).

Step 2: For SC_x, CA calculates the encryption key (EK_x) as $EK_x = H(k_{CA}||ID_{(}SC_x))$, and then computes $R_x = E_{(}d_x)(EK_x)$.

Step 3: CA computes the public parameters corresponding to SC_x's predecessors and successors including itself and declares publicly as follows:

1) For all $SC_x \leq SC_i$, CA calculates $EK_i = H(k_{CA}||ID_{(}SC_i))$ and public parameters as $M_{ix} = E_{(}EK_i)(SK_x \oplus ID_{CA})$.
2) For all $SC_j \leq SC_x$, CA computes $SK_j = D_{(}EK_j)(M_{jj}) \oplus ID_{CA}$. If it is valid, the CA computes the public parameters as $M_{xj} = E_{(}EK_x)(SK_j \oplus ID_{CA})$.

Step 4: CA sends d_x to SC_x securely.

3.3.2 Delete a Security Class from the Existing Hierarchical Structure

Suppose that an existing security class (SC_x) was deleted from the existing user hierarchical structure($SC_j \leq SC_x \leq SC_i$). In order to assure the forward security, SC_x does not have permission to access all the information which was authorized originally after being deleted from the user hierarchy system. CA performs the following steps to manage the authority to access:

Step 1: CA removes all the parameters corresponding to SC_x.

Step 2: For all the successors SC_j ($SC_j \leq SC_x$), CA renews the secret key as SK_j^*.

Step 3: For all the relationships $SC_j \leq SC_i$, CA renews the parameter $M_{ij}^* = E_{(}EK_i)(SK_j^* \oplus ID_{CA})$.

4 Analysis of the Proposed Scheme

In this section, we provide the analysis of security, storage and computational overheads required in our scheme. We assume that there are N security classes in the hierarchy system which forms a set $SC = \{SC_1, SC_2, \cdots, SC_N\}$, and each security class SC_i has v_i number of predecessors.

4.1 Storage Complexity

In Table 2, we show the storage space of private and public domains of CA and security classes for the hierarchy as shown in Figure 1. In our scheme, each key (k_{CA}, SK_i, d_i, EK_i) uses 128 bits, so CA and SC_i's private domain need 128 bits storage space. And considering the public domain for storage complexity, CA requires to store public parameters in the public domain: ID_{CA} requires 128 bits, and $NID_{(SC_i)}$ need $128N$ bits; NR_i require $128N$ bits, and finally M_{ij} need $128(\sum_{(i=1)}^{N}(v_i+1))$ bits. The total storage complexity required for the public domain is $128(\sum_{(i=1)}^{N}(v_i+1)+2N+1)$ bits. We present the comparison of the storage space between our scheme and Odelu et al.'s scheme [16] in Table 3.

4.2 Computational Complexity

For analysis of the computational complexity, because exclusive-OR operation requires very few computations, we neglect considering its computational cost in this paper. CA computes the NEK_i which require NT_{SHA1} operations, NR_i which need NT_{AES} operations, and finally M_{ij} which need $\sum_{(i=1)}^{N}(v_i+1)T_{AES}$ operations. And during the key derivation phase, all security classes SC_i in the hierarchy need to compute the followings: decrypting R_i requires NT_{AES} operations, and decrypting M_{ij} needs $\sum_{(i=1)}^{N}(v_i+1)T_{AES}$ operations. Therefore,

Table 3: Comparison of the storage space

	Private Domain		
Schemes	CA	SC_i	**Public Domain**
Odelu et al. [16]	128	128	$128(\sum_{(i=1)}^{N}(v_i+1)+3N+1)$
Ours	128	128	$128(\sum_{(i=1)}^{N}(v_i+1)+2N+1)$

Table 4: Comparison of computational costs

Schemes	Time complexity
Odelu et al. [16]	$(3N+2\sum_{i=1}^{N}(v_i+1))T_{AES}+(5N+1)T_{SHA1}$
Ours	$(2N+2\sum_{(i=1)}^{N}(v_i+1))T_{AES}+NT_{SHA1}$

the total computational complexity for the key generation and derivation phases becomes $(2N+2\sum_{(i=1)}^{N}(v_i+1))T_{AES}+NT_{SHA1}$. We also present the comparison of computational costs between our scheme and Odelu et al.'s scheme [16] in Table 4.

4.3 Security Analysis

Then we show that our scheme can resist three attacks which often occur in the user hierarchy system.

1) Contrary attack: In this attack, SC_i is a predecessor security class of a successor SC_j $(SC_j < SC_x)$. The attacker SC_j attempts to derive the secret key (SK_i) of SC_i from the public parameters available in the public domain. SC_j has $M_{ii} = E_(EK_i)(SK_i \oplus ID_{CA})$ and $R_i = E_(d_i)(EK_i)$. Because d_i is the subsecret key only known to SC_i, SC_j cannot decrypt R_i to know EK_i. Therefore, SC_j cannot decrypt M_{ii} without EK_i. In a word, it is infeasible that SC_j derives SK_i of SC_i. Our proposed scheme can resist the contrary attack.

2) Collaborative attack: Suppose that several users in security classes (SC_j, SC_x) collaborate to derive illegally the secret key (SK_i) of their predecessor (SC_i), where SC_j, $SC_x < SC_i$. The attackers cannot use their sub-secret keys (d_j, d_x) and public parameters $(M_{ii}, M_{ij}, M_{ix}, R_i)$ to compute EK_i, because d_i is the sub-secret key only known to SC_i. As a result, SC_j and SC_x cannot get SK_i through collaborative attack.

3) Forward security of successors: When we delete a security class (SC_x) from the existing hierarchical structure $(SC_j < SC_x < SC_i)$, CA replaces the secret key (SK_j) of SC_j with the renewed secret key SK_j^* and the related public parameters. Therefore, SC_x will not have chance to get the new secret key SK_j^*.

5 Conclusion

In this paper, we have proposed an improved key management scheme to solve a dynamic access problem in a user hierarchy by utilizing a symmetric-key cryptosystem and a one-way hash function. Compared with Odelu et al.'s scheme [16], both the time complexity and storage space are reduced in our scheme. Our scheme is more efficient and suitable for the hierarchy system.

References

[1] S. G. Akl and P. D. Taylor, "Cryptographic solution to a problem of access control in a hierarchy," *ACM Transactions on Computer Systems*, vol. 1, no. 3, pp. 239–248, July 1983.

[2] B. J. Cacic and R. Wei, "Improving indirect key management scheme of access hierarchies," *International Journal of Network Security*, vol. 4, no. 2, pp. 128–137, 2007.

[3] D. Giri and P. D. Srivastava, "A cryptographic key assignment scheme for access control in poset ordered hierarchies with enhanced security," *International Journal of Network Security*, vol. 7, no. 2, pp. 223–234, 2008.

[4] D. Giri and P. D. Srivastava, "An asymmetric cryptographic key assignment scheme for access control in tree structural hierarchies," *International Journal of Network Security*, vol. 4, no. 3, pp. 348–354, 2007.

[5] M. S. Hwang, "A cryptographic key assignment scheme in a hierarchy for access control," *Mathematical and Computer Modelling*, vol. 26, no. 2, pp. 27–31, 1997.

[6] M. S. Hwang, "Extension of CHW cryptographic key assignment scheme in a hierarchy," *IEE Proceedings Computers and Digital Techniques*, vol. 146, no. 4, pp. 219, 1999.

[7] M. S. Hwang, "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, vol. 82-A, no. 3, pp. 548–550, 1999.

[8] M. S. Hwang, "A new dynamic cryptographic key generation scheme in a hierarchy," *Nordic Journal of Computing*, vol. 6, no. 4, pp. 363–371, 1999.

[9] M. S. Hwang, "An asymmetric cryptographic scheme for a totally-ordered hierarchy," *International Journal of Computer Mathematics*, vol. 73, pp. 463–468, 2000.

[10] M. S. Hwang, "Cryptanalysis of YCN key assignment scheme in a hierarchy," *Information Processing Letters*, vol. 73, no. 3, pp. 97–101, 2000.

[11] M. S. Hwang and W.P. Yang, "Controlling access in large partially ordered hierarchies using cryptographic keys," *Journal of Systems and Software*, vol. 67, no. 2, pp. 99–107, 2003.

[12] Iuon-Chung Lin, Min-Shiang Hwang, and Chin-Chen Chang, "A new key assignment scheme for enforcing complicated access control policies in hierarchy," *Future Generation Computer Systems*, vol. 19, no. 4, pp. 457–462, 2003.

[13] Iuon-Chung Lin, H. H. Ou, and Min-Shiang Hwang, "Efficient access control and key management schemes for mobile agents," *Computer Standards & Interfaces*, vol. 26, no. 5, pp. 423–433, 2004.

[14] X. Liu, J. Ma, J. Xiong, and G. Liu, "Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption data," *International Journal of Network Security*, vol. 16, no. 6, pp. 437–443, 2014.

[15] J. W. Lo, M. S. Hwang, and C. H. Liu, "An efficient key assignment scheme for access control in a large leaf class hierarchy," *Information Sciences*, vol. 181, no. 4, pp. 917–925, 2011.

[16] V. Odelu, A. K. Das, and A. Goswam, "An effective and secure key-management scheme for hierarchical access control in e-medicine system," *Journal of Medical Systems*, vol. 37, no. 2, 2013. (doi: 10.1007/s10916–012–9920–5)

[17] S. F. Tzeng, C. C. Lee, and T. C. Lin, "A novel key management scheme for dynamic access control in a hierarchy," *International Journal of Network Security*, vol. 12, no. 3, pp. 178–180, 2011.

[18] Q. Zhang, Y. Wang, and J. P. Jue, "A key management scheme for hierarchical access control in group communication," *International Journal of Network Security*, vol. 7, no. 3, pp. 323–334, 2008.

[19] R. Zhou, C. Xu, W. Li, and J. Zhao, "An id-based hierarchical access control scheme with constant size public parameter," *International Journal of Network Security*, vol. 18, no. 5, pp. 960–968, 2016.

Biography

Wan-Yu Chao received the B.S. degree in Information Management from Central Police University, Taiwan, Republic of China; the M.S. degree in Graduate Institute of Management Information Systems from National Chung Hsing University. Her current research interests include applied cryptography, cloud computing, and mobile communications.

Cheng-Yi Tsai received the B.S. degree in Department of Business Administration from Chaoyang University of Technology (CYUT), Taichung, Taiwan, in 2001; the M.S. degree in Computer Science & Information Engineering from Asia University, Taichung, Taiwan, in 2005. He is currently pursuing his PHD degree in Graduate Institute of Computer Science & Information Engineering from Asia University. His current research interests include applied cryptography and mobile communications.

Min-Shiang Hwang Min-Shiang Hwang received the B.S. in Electronic Engineering from the National Taipei Institute of Technology, Taipei, Taiwan, in 1980; the M.S. in Industrial Engineering from the National Tsing Hua University, Taiwan, in 1988; and a Ph.D. in Computer and Information Science from the National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at the National Cheng Kung University, Taiwan, from 1984-1986. 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 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. He was also the 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 was a professor and chairman of the Department of Management Information Systems, National Chung Hsing University (NCHU), during 2003-2009. He was also a visiting professor of UC. Riverside and UC. Davis (USA) during 2009-2010. He was a distinguished professor of the department of Management Information Systems, NCHU, during 2007-2011. He obtained the 1997, 1998, 1999, 2000, and 2001 Excellent Research Award of National Science Council (Taiwan). He was a dean of College of Computer Science, Asia University (AU). He is currently a Chair Professor of the department of Computer Science & Information Engineering, AU. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include information security, electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang has published over 200+ articles on the above research fields in international journals.