Efficient Date-Constraint Hierarchical Key Assignment Scheme

Wei-Bin Lee, Jia-Hong Li, and Chyi-Ren Dow
Department of Information Engineering and Computer Science,
Feng Chia University

Abstract — It is well-known that there exists a trade-off between crypto-period and affected risk, so the key derivation is not only constrained by the partial-order class relation but also by the time period while handling hierarchical key assignment in dynamic enterprises. In this paper, a new date-constraint hierarchical key assignment scheme, which does not make use of public key cryptosystem, is proposed to give a cost-effective solution.

Keywords: Cryptography, HMAC, Hierarchy, Key assignment.

1 Introduction

In current society, many organizations adopt a hierarchical structure, such as in the military, government, or business companies, and each usually composed of many distinct entities with distinct privileges to access different levels of resources. In such an environment, controlling the access of data is of utmost importance. Hence, the work of key management in such a hierarchical structure to protect the secret data from leakage is obviously an important issue. Consider a hierarchical organization in which the users are divided into a set of disjointed classes, $S = \{C_i \mid 1 \le i \le n\}$, and are partially ordered by a binary relationship "$\preceq$". The relationship $C_j \preceq C_i$ means that C_i has higher privilege than C_j; that is the users in class C_i can access the data in classes C_j, while the opposite is not allowed. In 1983, Akl and Taylor [1] provided a key assignment solution for such structure, which was based on RSA [10] cryptosystem, to solve the above access control problem. Then, Sandhu [11] also proposed some cryptographic solutions for access control in a tree hierarchy. However, the above researches do not closely consider and discuss the issue of dynamic environment.

Afterward, based upon cryptographic techniques, many schemes [2, 3, 6, 8, 9] are proposed to solve the dynamic issue. These researches strive to enhance performance in key assignment and derivation to handle the essential necessities of dynamic environment where the members may join or leave the group dynamically. Beside from the fundamental issue of performance regarding key's process, there is yet another important concern, the cryptoperiod of key, which is less frequently addressed. It was not until Tzeng [13], that the

first time-bound key assignment scheme for a user hierarchy is proposed to handle the problem of time span of a specific key. With the help of time constraint, key update can automatically proceed to avoid the human-handle error or other frailty. Unfortunately, their approach was found to be computationally inefficient and some security problems were discovered in [14]. After that, Chien [4] proposed a new time-bound hierarchical key assignment scheme based on one-way hash function, nevertheless, it is not efficient enough to derive the successor's secret key. In this paper, a hash based MAC (Message Authentication Code) function is applied rather than the hash function, and the performance of computation and implementation is significantly improved.

The rest of this paper is organized as follows: Section 2 illustrates the proposed new date-constraint hierarchical key assignment scheme. Section 3 discusses the regular key change in the dynamic user hierarchy. After that, Section 4 and 5 examines the security and evaluates the performance, respectively. Finally, Section 6 concludes the summary of this paper.

2 Proposed Scheme

Considering the n disjoint classes $S = \{C_i \mid 1 \le i \le n\}$, the distinct cryptographic keys are assigned to these classes in the hierarchical tree. Without loss of generality, assume that C_i has privilege to derive the secret key K_j of C_j between time span t_1 to t_2 if and only if the relationship of $C_j \preceq C_i$ and $t_1 \le t \le t_2$ hold.

The proposed scheme is composed of two phases: the key assignment phase and the key derivation phase.

2.1 Key Assignment

Assume that the class C_i has a unique identity ID_i and the corresponding child node C_j has the identity ID_j.

Because that C_i has higher privilege, C_i undoubtedly does as follows to generate the secret key K_j for C_j:

Step 1. Generate a random number r_{ij} and compute $T_{ij} = r_{ij} \| t_1 \| t_2$.

Step 2. Calculate the secret key $K_j = HMAC_{K_i}(ID_i \| ID_j \| T_{ij})$, where K_i is the secret key of C_i, and $HMAC_{K_i}(\cdot)$ is a hash based MAC function with K_i [12].

Step 3. Distribute the secret key K_j for C_j through a secure channel.

2.2 Key Derivation

If the class C_i wants to decrypt the encrypted data in class C_j at a time span t, the following will be done with the help of the smart card owned by C_i:

Step 1. Insert the smart card into the card reader and enable the card.

Step 2. Input the public value $T_{ij} = r_{ij} \| t_1 \| t_2$, and then verify whether $t_1 \le t \le t_2$ or not.

Step 3. Derive the secret key $K_j = HMAC_{K_i}(ID_i \| ID_j \| T_{ij})$.

With the derived secret key K_j, the class C_i can decrypt the data encrypted by C_j.

3 Dynamic Access Control

In dynamic user hierarchy, a class may join or leave the group dynamically. In general, the parent is responsible for the key assignment procedure for the children. Therefore, the complexity of the key assignment for a parent is directly proportion to the degree of its children.

For easier understanding, the cases of the dynamic issue are further described.

3.1 Adding a New Class

In Figure 1, is the case of inserting a new class C_R as the right child of C_1. The secret key of C_R will be assigned from the parent class C_1. Accordingly, the secret key of C_3 will be regenerated by the new parent C_R. In the same way, the secret keys of C_6 and C_7 will be updated easily by the common parent C_3 with the new assigned key.

Therefore, the effort of key distribution is shared out among the parent node according to the partial-order classes instead of being centralized as done in a Central Authority. The total cost for key update is the directly proportion to the number of the descendants of the new inserted nodes.

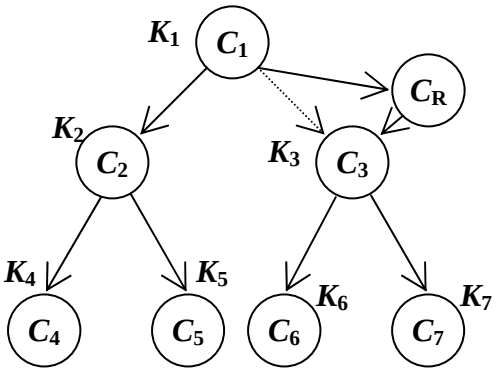

Figure 1 Adding a new inter-class.

3.2 Evicting an Existing Class

To avoid the evicted class C_j still keeping the access privilege of its original successors, the parent of C_j must regenerate the secret keys for the descendants of the class C_j. For example, in Figure 2, assume that the class C_3 is evicted, then the classes C_6 and C_7 will be the children of C_1. Accordingly, the secret keys of C_6 and C_7 will be updated by C_1.

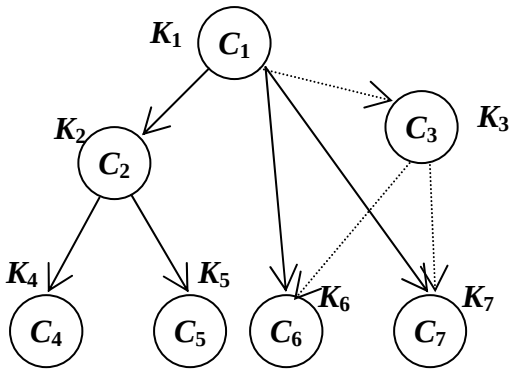

Figure 2 Evicting an existing inter-class.

Therefore, the total cost for key update is the directly proportion to the number of the evicted node's descendants.

4 Security Analysis

The proposed scheme applies a tamper-resistant device [5], which has the properties of low-cost, efficient computation, secure storage, and conditionally specific access to protect the secret data that even the owner cannot modify. Besides, a biometric verification [7], such as fingerprint, may be applied to enable the card. These procedures provide the first level security to protect the secrets stored in the card.

Except for the well physical protection of the card itself, the other major concern is whether the secret key can be derived from the usual possible methods of attack listed as follows:

Derive the secret key of parent: Assume that the class C_j tries to derive the secret key K_i of the parent class C_i from the public value $T_{ij} = r_{ij} \| t_1 \| t_2$. Fortunately, even though C_j knows his secret key K_j, it is still infeasible to derive the secret key K_i of the parent class due to the $HMAC_{K_i}(\cdot)$ based on hash function with irreversible one-way property. Therefore, there is no way to derive K_i from K_j.

Derive the secret key of sibling: Without knowing the secret key of the sibling's parent, it is very difficult to derive the certain secret key $K_j = HMAC_{K_i}(ID_i \| ID_j \| T_{ij})$ of sibling class C_j, for example, from the public parameter $T_{ij} = r_{ij} \| t_1 \| t_2$.

Derive the new children's keys by an evicted class: With the help of the key update, the new key is assigned for all descendants of the evicted node, so it is infeasible for the evicted class to derive the new children's keys according to the original old key.

Forge the legal expiration date: Because the collision-free property of the HMAC function, different T_{ij} will lead to different key. Trying to extend the key's cryptoperiod by altering the public value $T_{ij} = r_{ij} \| t_1 \| t_2$ is impossible.

5 Performance Evaluation

In our scheme, only one public value T_{ij} for each edge is necessary, therefore, the total number of public values is n-1.

Besides, the secret key is derived following the path from the parent class to the specific child node one by one, so the complexity is proportion to the path length in the hierarchical tree; that is, the edge distance. For each edge, only one HMAC operation is necessary.

Because the computation cost of one HMAC function could be regarded as one hash operation [12], assume that the cost for computing hash and HMAC function is T_h. The comparison between our and the other two latest schemes are illustrated in Table 1, which refers to Chien's literature [4]. Chien's scheme needs one hash operation to derive the decrypting key and another (t_2 - t_1) hash operations to verify the validity of the cryptperiod of key, therefore, for each edge, a total of $(t_2 - t_1 + 1)T_h$ operations are necessary to handle the key derivation. Tzeng's scheme performs (t_2 - t_1) modular exponentiation and Lucas function operations to derive key, and another hash operation to associate with time span, therefore, for each edge, a total of $(t_2 - t_1)T_e + (t_2 - t_1)T_L + T_h$ operations are needed. The other cases can be deduced likewise.

Table 1 Performance comparisons among three key assignment schemes.

	Ours	Chien [4]	Tzeng [13]
Implementation bases	HMAC function	Hash function	Public key cryptosystem
Public values	n - 1	n - 1	n + 6
Key derivation of one own class	T_h	$(t_2 - t_1 + 1)T_h$	$(t_2 - t_1)T_e + (t_2 - t_1)T_L + T_h$
Key derivation of one-edge-distance child class	T_h	$(t_2 - t_1 + 2)T_h$	$(t_2 - t_1 + r)T_e + (t_2 - t_1)T_L + T_h$
Key derivation of l-edge-distance child class	$l\,T_h$	$(t_2 - t_1 + 1 + l)T_h$	$(t_2 - t_1 + r)T_e + (t_2 - t_1)T_L + T_h$

*n: the number of classes; r: the number of the child classes C_ks, where $C_k \preceq C_i$ and C_k not $\preceq C_j$. T_h: the time complexity for one HMAC/hash operation; T_e: one modular exponentiation; T_L: one Lucas function operation.

Based on Table 1, our performance of the key derivation is significantly improved.

6 Conclusions

A new key assignment scheme with date-constraint for dynamic user hierarchy is proposed. With the attached time bound to the secret keys, it is simple to restrict the access privileges as a time-expire automatically indicates the invalidity of a private key. In addition, without public key cryptosystem, the proposed scheme enormously reduces the computational load compared with prior researches.

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, Aug. 1983.

[2] C.C. Chang, R.J. Hwang, and T.C. Wu, "Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," *Information Systems*, Vol. 17, No. 3, pp. 243-247, 1992.

[3] T.S. Chen and Y.F. Chung, "Hierarchical Access Control based on Chinese Remainder Theorem and Symmetric Algorithm," *Computers and Security*, Vol. 21, No. 6, pp. 565-570, 2002.

[4] H.Y. Chien, "Efficient Time-Bound Hierarchical Key Assignment Scheme," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 10, pp. 1301-1304, Oct. 2004.

[5] M.L. Gemplus, "Smart-Cards: A Cost-Effective Solution Against Electronic Fraud," *European Conference on Security and Detection*, No. 437, pp.81-85, Apr. 1997.

[6] L. Harn and H.Y. Lin, "A Cryptographic Key Generation Scheme for Multilevel Data Security," *Computers and Security*, Vol. 9, No. 6, pp. 539-546, 1990.

[7] A. Jain, R. Bolle, S. Pankanti, *Biometrics, Personal Identification in Networked Society*, Kluwer Academic Publishers, pp. 369-384, 1999.

[8] F.H. Kuo, V.R.L. Shen, T.S. Chen, and F. Lai, "Cryptographic Key Assignment Scheme for Dynamic Access Control in a User Hierarchy," *IEE Proceedings of Computers and Digital Techniques*, Vol. 46, No.5, pp. 235-240, Sept. 1999.

[9] C.H. Lin, "Hierarchical Key Assignment without Public-Key Cryptography," *Computers and Security*, Vol. 20, No. 7, pp. 612-619, Oct. 2001.

[10] R. Rivest, A. Shamir and L. Adleman, "A Scheme for Obtaining Digital Signature and Public-Key Cryptosystems," *Communications of the ACM*, Vol. 21, No.2, pp. 120-126, Feb. 1978.

[11] R. S. Sandhu, "On Some Cryptographic Solutions for Access Control in a Tree Hierarchy," *Proceedings of the 1987 Fall Joint Computer Conference on Exploring technology: today and tomorrow*, Dallas, Texas, United States, pp. 405-410, 1987.

[12] B. Schneier, *Applied Cryptography: Protocol, Algorithms, and Source Code in C (Second Edition)*, New York: John Wiley and Sons, 1996.

[13] W.G. Tzeng, “A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy,” *IEEE Transactions On Knowledge and Data Engineering*, Vol. 14, No. 1, pp.182-188, Jan.-Feb. 2002.

[14] X. Yi and Y. Ye, “Security of Tzeng’s Time-Bound Key Assignment Scheme for Access Control in a Hierarchy,” *IEEE Transactions On Knowledge and Data Engineering*, Vol. 15, No. 4, pp.1054-1055, July-Aug. 2003.