Comment on Wu-Chang cryptographic key assignment scheme for hierarchical access control

Tzong-Sun Wu[1], Chien-Lung Hsu[2], Han-Yu Lin[2], and Shu-Hui Kao[2]

[1] Graduate Institute of Informatics
Fo Guang University
I Lan, Taiwan 262, Republic of China

[2] Department of Information Management,
Huafan University,
Taipei, Taiwan 223, Republic of China

*Correspondence to:
Associate Professor Tzong-Sun Wu, Ph. D.
Graduate Institute of Information Science
Fo Guang University
160, Linwei Road, Jiaushi Hsiang, I Lan Hsien, 262
Taiwan, Republic of China
E-mail: tswu@cc.hfu.edu.tw
tswu@mail.fgu.edu.tw
Tel: +886-2-2663-2102 ext 4356
Fax: +886-2-2663-1119

Abstract

Wu and Chang proposed a cryptographic key assignment scheme for hierarchical access control in 2001. Based on the discrete logarithm problem, their scheme can be applied on a partially ordered set user hierarchy. However, in this paper, we will show that their scheme violates the predefined access control policy. Further, a simple improvement is given to eliminate the security leak.

Keywords: cryptanalysis, key assignment, access control, poset

1 Introduction

Recently, Wu and Chang proposed a cryptographic key assignment scheme for hierarchical access control [1]. In their scheme, the users belonging to a security class must follow the predefined partially ordered relation to have access to the information held by their successor(s). The purpose of this paper is show that the users can have access to the information without following the predefined relation, which violates Wu and Chang's claim. Furthermore, we propose an improvement to eliminate the security leak inherent in the Wu-Chang scheme.

2 Brief Review of Wu-Chang Scheme

Let $\boldsymbol{C} = \{C_1, C_2, \ldots, C_n\}$ be a set of n security classes in the hierarchy and the notation "≤" the binary partially ordered relation on $\boldsymbol{C}$. In the partially ordered set (poset) $(\boldsymbol{C}, \leq)$, $C_j \leq C_i$ means that the users in C_i have a security class higher than or equal to those in C_j. Figure 1 shows a poset access control hierarchy. We denote ID_{CA}, ID_{C_i}, and $ID_{u_{jt}}$ as the identifiers for the central authority (CA), the security class C_i, and the user u_{jt}, who is the t-th user belonging to C_j, respectively. In the system setup phase, CA selects be two large primes p and q satisfying that $|q| \geq |p| + |ID|$, and a primitive root g over $GF(p)$, where $|x|$ is the bit-length of the integer x. Then, CA chooses his private key S_{CA} such that $\gcd(S_{CA}, p - 1) = 1$, computes his public key $Y_{CA} = g^{S_{CA}} \bmod p$, and then publishes $\{p, q, g, Y_{CA}\}$. In the key generation phase, CA randomly chooses a distinct derivation key S_{C_i} such that $\gcd(S_{C_i}, p - 1) = 1$, computes its corresponding public key $Y_{C_i} = g^{S_{C_i}} \bmod p$, and then transmits S_{C_i} to each user $u_{it} \in C_i$ via a secure channel (for $C_i \in \boldsymbol{C}$). Upon receiving S_{C_i} from CA, u_{it} chooses an encryption key K_{it} over GF(p) and publishes $W_{it} = (K_{it})^{S_{C_i}^{-1}} + ID_{u_{it}} \bmod p$. In addition, CA generates the public derivation polynomial $f_i(x)$ over GF(q) for each security class C_i with a bottom-up approach for all security classes in the hierarchy. That is, CA generates a public polynomial $f_i(x)$ by interpolating the points

$$(((Y_{CA})^{S_{C_i}} \bmod p) \,||\, ID_{C_j}, S_{C_j})\text{'s} \tag{1}$$

for all $C_j \le C_i$ and

$$(((Y_{CA})^{S_{C_i}} \bmod p) \,||\, ID_{CA}, R_i) \tag{2}$$

with Lagrange interpolating formula [2]. Here, R_i is a secret number which is used for the protection purpose.

In the key derivation phase, the user $u_{ia} \in C_i$ who wants to have access to the information items held by any user u_{jb} in C_j, where $C_j \le C_i$, computes

$$S_{C_j} = f_i(((Y_{CA})^{S_{C_i}} \bmod p) \,||\, ID_{C_j}) \tag{3}$$

and derives u_{jb}'s encryption key K_{jb} as

$$K_{jb} = (W_{jb} - ID_{u_{jb}})^{S_{C_j}} \bmod p \tag{4}$$

Thereafter, u_{ia} uses K_{jb} to decrypt u_{jb}'s information items. The Wu-Chang scheme also deals with some dynamic access control problems in their paper, such as adding a new security class into the hierarchy, deleting an old security class from the hierarchy, etc. The interested reader may refer to [1] for the detailed discussion.

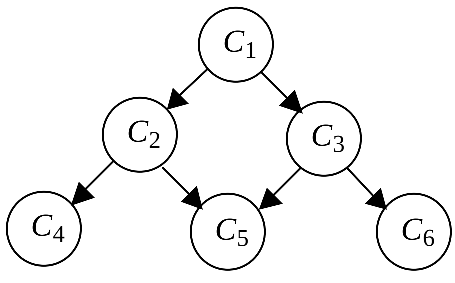

Figure 1. A poset access control hierarchy with six security classes

3 Security Leak and Improvement

According to the access policy in the poset hierarchy, users can only have access to the information held by the users with an equal or lower security classes. In the following, we will show that the Wu-Chang scheme violates the predefined requirement. For the poset hierarchy in Figure 1, the user u_{3a} knowing S_{C_3} in the security class C_3 can compute the derivation key S_{C_5} of the security class C_5 by eqn. 3, i.e, $S_{C_5} = f_3(((Y_{CA})^{S_{C_3}} \bmod p) \| ID_{C_5})$. Since the security classes C_3 and C_2 have the same immediate successor C_5, u_{3a} can use S_{C_5} to resolve the roots of the equation $f_2(x) = S_{C_5} \pmod p$ over $GF(q)$ in polynomial time [3, 4] and obtain $((Y_{CA})^{S_{C_2}} \bmod p) \| ID_{C_5}$. He further uses the shared key $(Y_{CA})^{S_{C_2}} \bmod p$ between C_2 and CA to compute the derivation key $S_{C_4} = f_2(((Y_{CA})^{S_{C_2}} \bmod p) \| ID_4)$ by eqn. 3. With this derivation key S_{C_4}, u_{3a} can obtain the encryption key of users in C_4 by eqn. 4, and thus can have access to the information held by the users in C_4.

The security leak inherent in the Wu-Chang scheme is caused by the fact that the shared key $(Y_{CA})^{S_{C_2}} \bmod p$ can be compromised and used to compute the derivation key(s) of C_2's successor(s). We can easily eliminate the weakness by using a one-way hash function h [5] to protect the shared key $(Y_{CA})^{S_{C_2}} \bmod p$ from being revealed. Note that the function h maps a string of variable length to a string of $|q|$ bits. To strengthen the Wu-Chang scheme, we replace eqns. 1 and 2 with eqns. 1* and 2*, respectively:

$$(h(((Y_{CA})^{S_{C_i}} \bmod p) \| ID_{C_j}), S_{C_j})\text{'s} \qquad (1^*)$$

$$(h(((Y_{CA})^{S_{C_i}} \bmod p) \| ID_{CA}), R_i) \qquad (2^*)$$

Consequently, eqn. 3 should also be changed to:

$$S_{C_j} = f_i(h(((Y_{CA})^{S_{C_i}} \bmod p) \| ID_{C_j})) \qquad (3^*)$$

Based on the intractability of reversing the one-way hash function h, it is computationally infeasible to compute the derivation key(s) violating the predefined access policy, since the shared key $(Y_{CA})^{S_{C_i}} \bmod p$ now is protected by h. Hence, our improvement can withstand the attack stated above.

References

1 Wu, T.C. and Chang, C.C.: 'Cryptographic key assignment scheme for hierarchical access control', *International Journal of Computer Systems Science and Engineering*, 2001, 1 (1), pp. 25-28.

2 Knuth, D.E.: 'The art of computer programming, volume 2, seminumerical algorithms' (Addison-Wesley, MA, 1981 2nd Edition)

3 Ben-Or, M.: 'Probabilistic algorithms in finite fields'. 22nd Annual Symposium on Foundations of Computer Science, IEEE FOCS'81, October 1981, Nashville, Tennessee, pp. 394-398.

4 Cohen, H.: 'A course in computational algebraic number theory' (Springer-Verlag, 1991)

5 Diffie, W. and Hellman, M.: 'New directions in cryptography', *IEEE Transactions on Information Theory*, 1976, IT-22 (6), pp. 644-654.