On the Insecurity of a Time-Bound Hierarchical Key Assignment Scheme

Alfredo De Santis, Anna Lisa Ferrara, and Barbara Masucci

Dipartimento di Informatica ed Applicazioni
Università di Salerno, 84081 Baronissi (SA), Italy

E-mail: {ads, ferrara, masucci}@dia.unisa.it

February 10, 2005

Abstract

Recently, Chien [3] has proposed a time-bound cryptographic key assignment scheme in a partially ordered hierarchy. The scheme is based on tamper-resistant devices and requires low computational load and implementation cost. Unfortunately, the scheme is not secure. In this paper we show how three malicious users can handle public and private information to misuse their tamper-resistant devices in order to compute some encryption keys that they should not be able to learn. We also show some countermeasures to withstand the weakness we have exploited.

Keywords: Access control, key assignment, collusion attack.

1 Introduction

The *access control problem* deals with the specification of users' access permission and is a fundamental issue in any system that manages distributed resources. We consider a scenario where the users of a computer system are organized in a hierarchy formed by a certain number of disjoint classes, called *security classes*. A hierarchy arises from the fact that some users have more access rights than others. In the real world there are several examples of hierarchies where an access control is required. Applications exist in business and in other areas of the private sector, for example in the management of databases containing sensitive information or in the protection of industrial secrets. Similar situations abound in other areas, particularly in the government and military.

A *hierarchical key assignment scheme* is a method to assign an encryption key and some private information to each class in the hierarchy. The encryption key will be used by each class to protect its data by means of a symmetric cryptosystem. The private information will be used by each class to compute the keys assigned to all classes lower down in the hierarchy. This assignment is carried out by a central authority, the TA, which is active only at the distribution phase. Akl and Taylor [2] first proposed an elegant hierarchical key assignment scheme. In their scheme each class is assigned a key that can be used, along with some public parameters generated by a central authority, to compute the key assigned to any class lower down in the hierarchy. Subsequently, many researchers have proposed schemes that either have better performances or allow insertion and deletion of classes in the hierarchy (e.g., [5, 6, 7, 8, 9, 11]). The problem of designing key assignment schemes for any arbitrary access control policy (i.e., not satisfying the anti-symmetric and transitive properties of a partially ordered hierarchy) has been considered in [4], where a general construction using as a building block a hierarchical key assignment scheme has been proposed.

In some situations, as electronic paper subscription and digital TV broadcasting, a user may be assigned to a certain class for only a certain period of time. In such situations the users need a different key for each time period, to encrypt their data. A straightforward implementation of such an assignment scheme requires users to handle a large amount of keys, which is not practical. The problem of reducing the space complexity of the basic straightforward solution was first considered by Tzeng [13], who proposed a *time-bound* hierarchical key assignment scheme. In his scheme the key derivation is constrained not only by the hierarchy on the classes, but also by the time period. Once the time period is expired, users in a class cannot access any subsequent class keys. However, Yi and Ye [12] showed that the scheme proposed by Tzeng was insecure against collusion attacks carried out by non-authorized users. Recently, Chien [3] has proposed a new time-bound hierarchical key assignment scheme based on tamper-resistant devices that perform only simple arithmetic operations, such as computing hash values or XOR operations. The scheme requires low computational load and implementation cost. Chien analyzed the security of his scheme against some attacks and claimed that a coalition of classes with distinct time periods cannot cooperate to compute unauthorized keys.

In this paper we show that Chien's scheme is insecure against collusion attacks carried out by non-authorized users. In particular, we show how three malicious users can handle public and private information to misuse their tamper-resistant devices in order to compute some encryption keys that they should not be able to learn. We also show some countermeasures to withstand the attack we have exploited.

The paper is organized as follows: in Section 2 we briefly describe Chien's time-bound hierarchical key assignment scheme. In Section 3 we show how three malicious users may co-operate in order to gain non-authorized data access. In Section 4 we propose some countermeasures to withstand the attack described in Section 3. Finally, in Section 5 we draw our conclusions.

2 Overview of Chien's Scheme

Consider a set of users divided into a number of disjoint classes, $C_1, \ldots, C_n$, called *security classes*. A security class can represent a person, a department, or a user group in an organization. In accordance with authority, position, or power, there is a binary relation $\preceq$ that partially orders the set of classes $\mathcal{C} = \{C_1, \ldots, C_n\}$. The poset $(\mathcal{C}, \preceq)$ is called a *partially ordered hierarchy*. For any two classes C_i and C_j, the notation $C_i \preceq C_j$ is used to indicate that the users in C_j can access C_i's data. Clearly, since C_j can access its own data, it holds that $C_j \preceq C_j$, for any $j = 1, \ldots, n$. The partially ordered hierarchy $(\mathcal{C}, \preceq)$ can be represented by a directed acyclic graph, where each class corresponds to a vertex in the graph and there is an edge from class C_j to class C_i if and only if $C_i \preceq C_j$. Further, this graph can be simplified by eliminating all self-loops and edges which can be implied by the property of the transitive closure.

The life-time of a *time-bound* cryptographic key assignment scheme for $(\mathcal{C}, \preceq)$ is divided into a certain number of time periods, starting at time 0 and ending at time z, where z is an integer. Each class C_j has several encryption keys, one for each time period. We denote by $K_{j,t}$ the key assigned to class C_j at time t. Such a key is used by the users in C_j to encrypt their data at time t. A user belonging to class C_j from time t_1 to t_2 can compute the key $K_{i,t}$ held by class C_i at time t if and only if $C_i \preceq C_j$ and $t_1 \leq t \leq t_2$.

Chien's scheme [3] is based on the following main assumptions:

1. There is a secure one-way hash function $h()$;
2. Tamper-resistant devices are available. The owner cannot access the protected data in the device.
3. There is a Trusted Agent (TA) whose task is to assign the encryption keys and the private information to the classes.
4. The system starts at time period 0 and ends at an arbitrary large time period z.

Other assumptions implicitly used by the scheme are the following:

1. An authenticated public-board is available. Everyone can read the data on the board but only the TA can update the data.
2. There is a secure channel connecting the TA to each class of users. Such a channel is used by the TA to distribute the private information to each class during the user registration phase.
3. Each class C_i has a public identity ID_i.
4. The hierarchy $(\mathcal{C}, \preceq)$ is public.

The scheme is composed of four phases: initialization, user registration, encrypting key generation and decrypting key derivation.

Initialization. The TA randomly chooses two secret values a and b and a secret key k_i, for any $i = 1, \ldots, n$. Moreover, for any two distinct classes C_i and C_j such that $C_j \preceq C_i$ and there is no class C_ℓ such that $C_j \preceq C_\ell \preceq C_i$, the TA publishes the value $r_{ij} = h(X||ID_i||ID_j||k_i) \oplus k_j$ on the authenticated public board, where X is the TA's secret key, $||$ denotes the string concatenation, and $\oplus$ denotes the bit-wise XOR operation.

User Registration. The TA sends to each class C_i the secret key k_i, chosen during the initialization phase, by using a secure channel. Any user belonging to C_i in time interval $[t_1, t_2]$ will be able to compute its encryption key $K_{i,t}$, for any $t_1 \leq t \leq t_2$, by using the secret key k_i and a tamper-resistant device issued by the TA. Such a device contains

- the TA's secret key X;
- the identity ID_i of C_i;
- the times t_1, t_2, and z;
- the hash values $h^{t_1}(a)$ and $h^{z-t_2}(b)$, where the expression $h^m(x) = h(h(\ldots(h(x))\ldots))$ denotes the application of m cascade hashing operations starting from x.

Encrypting Key Generation. The data belonging to C_i during the time period t will be encrypted by using the key $K_{i,t} = h(k_i \oplus h^t(a) \oplus h^{z-t}(b))$.

Decrypting Key Derivation. Let C_i and C_j be two distinct classes such that $C_j \preceq C_i$ and there is no class C_ℓ such that $C_j \preceq C_\ell \preceq C_i$. An user belonging to C_i in time interval $[t_1, t_2]$ is able to compute the key $K_{j,t}$ held by C_j at time t, where $t_1 \leq t \leq t_2$, by executing the following steps:

1. The user inputs to its tamper-resistant device the public value r_{ij}, the identity ID_j, the time t, and his secret key k_i.

2. The tamper-resistant device derives the secret key k_j by computing $k_j = r_{ij} \oplus h(X||ID_i||ID_j||k_i)$.

3. The tamper-resistant device computes $h^t(a) = h^{t-t_1}(h^{t_1}(a))$ and $h^{z-t}(b) = h^{t_2-t}(h^{z-t_2}(b))$.

4. The tamper-resistant device computes $K_{j,t} = h(k_j \oplus h^t(a) \oplus h^{z-t}(b))$.

The above steps can be performed when $C_j \preceq C_i$ and there is no class C_ℓ such that $C_j \preceq C_\ell \preceq C_i$. On the other hand, if $C_j \preceq C_{\ell_s} \preceq \cdots \preceq C_{\ell_1} \preceq C_i$, where $s \geq 1$, similar computations can be performed to iteratively compute the key held by any class in the path from C_i to C_j. For clarity, we explain how this happens:

1. The user inputs to its tamper-resistant device the sequential list of the public values $r_{i\ell_1}, r_{\ell_1\ell_2}, \ldots, r_{\ell_{s-1}\ell_s}, r_{\ell_s j}$ and of the identities $ID_{\ell_1}, \ldots, ID_{\ell_s}$, ID_j of the classes along the path from C_i to C_j, the time t, and his secret key k_i.

2. The tamper-resistant device derives the secret key k_{ℓ_1} by computing $k_{\ell_1} = r_{i\ell_1} \oplus h(X||ID_i||ID_{\ell_1}||k_i)$. Afterwards, for any $w = 2, \ldots, s$, it computes $k_{\ell_w} = r_{\ell_{w-1}\ell_w} \oplus h(X||ID_{\ell_{w-1}}||ID_{\ell_w}||k_{\ell_{w-1}})$. Finally, it computes $k_j = r_{\ell_s j} \oplus h(X||ID_{\ell_s}||ID_j||k_{\ell_s})$.

3. The tamper-resistant device computes $h^t(a) = h^{t-t_1}(h^{t_1}(a))$ and $h^{z-t}(b) = h^{t_2-t}(h^{z-t_2}(b))$.

4. The tamper-resistant device computes $K_{j,t} = h(k_j \oplus h^t(a) \oplus h^{z-t}(b))$.

Clearly, any user belonging to C_i in time interval $[t_1, t_2]$ is also able to compute the key $K_{i,t}$, where $t_1 \leq t \leq t_2$. This is implicit in Chien's scheme, but for clarity we explain how this happens:

1. The user inputs to its tamper-resistant device the time t and his secret key k_i.

2. The tamper-resistant device computes $h^t(a) = h^{t-t_1}(h^{t_1}(a))$ and $h^{z-t}(b) = h^{t_2-t}(h^{z-t_2}(b))$.

3. The tamper-resistant device computes $K_{i,t} = h(k_i \oplus h^t(a) \oplus h^{z-t}(b))$.

3 A Collusion Attack to Chien's Scheme

Consider the hierarchy of Figure 1 and assume that the users in C_1, C_2, C_3, and C_4 are associated with time intervals $[t_1, t_6]$, $[t_2, t_5]$, $[t_3, t_4]$, and $[t_2, t_3]$, respectively, where $0 \leq t_1 < t_2 < t_3 < t_4 < t_5 < t_6 \leq z$.

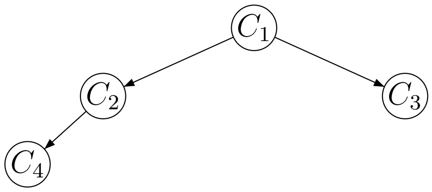

Figure 1: A partially ordered hierarchy.

During the initialization phase, the TA randomly chooses the secret keys k_1, k_2, k_3, k_4 and publishes the values $r_{12} = h(X||ID_1||ID_2||k_1) \oplus k_2$, $r_{13} = h(X||ID_1||ID_3||k_1) \oplus k_3$, and $r_{24} = h(X||ID_2||ID_4||k_2) \oplus k_4$ on the authenticated public-board.

During the user registration phase, the TA sends to each class the corresponding secret key and the tamper-resistant devices. Since $C_4 \preceq C_2$, during the decrypting key derivation phase users in C_2 can use the public value r_{24} and their tamper-resistant device to compute $K_{4,t}$, for any $t_2 \leq t \leq t_5$.

In the following we show how users in C_3 and C_4 can collude to compute a faked public value r'_{24}. Such a value can be used at any time $t_4 < t \leq t_5$ to misuse the tamper-resistant device held by users in C_2, in order to compute the unauthorized key $K_{3,t}$. Notice that, according to Chien's scheme, users in C_3 should not be able to compute $K_{3,t}$ for any $t_4 < t \leq t_5$, even if they collude with other unauthorized classes with distinct time intervals. In order to compute the faked public value, the coalition has to perform very simple operations, i.e., two XOR operations, as shown in the following:

1. A malicious user belonging to C_4 in time interval $[t_2, t_3]$ uses the public value $r_{24} = h(X||ID_2||ID_4||k_2) \oplus k_4$ and its secret key k_4 to compute the hash value $h(X||ID_2||ID_4||k_2) = r_{24} \oplus k_4$.

2. A malicious user belonging to C_3 in time interval $[t_3, t_4]$ uses its secret key k_3 and the hash value $h(X||ID_2||ID_4||k_2)$, computed by the malicious user in C_4, to compute a faked public value $r'_{24} = h(X||ID_2||ID_4||k_2) \oplus k_3$.

Now we show how a malicious user in C_2 can use the faked public value r'_{24} computed by the malicious users in C_3 and C_4 to misuse the tamper-resistant device held by users in C_2, in order to compute the unauthorized key $K_{3,t}$ at any time $t_4 < t \leq t_5$.

1. The malicious user in C_2 inputs to its tamper-resistant device the faked public value r'_{24}, the identity ID_4, the time t, and his secret key k_2.

2. The tamper-resistant device computes $r'_{24} \oplus h(X||ID_2||ID_4||k_2) = k_3$.

3. The tamper-resistant device computes $h^t(a) = h^{t-t_2}(h^{t_2}(a))$ and $h^{z-t}(b) = h^{t_5-t}(h^{z-t_5}(b))$.

4. The tamper-resistant device computes $K_{3,t} = h(k_3 \oplus h^t(a) \oplus h^{z-t}(b))$.

Therefore, users in C_3 are able to compute the key $K_{3,t}$ for any $t_4 < t \leq t_5$, by colluding with other unauthorized classes with distinct time intervals.

4 Countermeasures

The weakness of Chien's scheme we have exploited relies in the fact that the public information may be faked to misuse tamper-resistant devices held by malicious users. In the following we propose some countermeasures to withstand the attack described in the previous section. In order to improve the security of the scheme, the ideas behind such countermeasures can be combined together.

4.1 A First Countermeasure

The first countermeasure we propose requires the TA to perform one more hashing operation for each public value, but does not change the kind of operations performed by the tamper-resistant devices held by the users. For each pair of distinct classes C_i and C_j such that $C_j \preceq C_i$ and there is no class C_ℓ such that $C_j \preceq C_\ell \preceq C_i$, the TA could compute the public value r_{ij} as $r_{ij} = h(X||ID_i||ID_j||k_i) \oplus h(X||k_j)$. The encryption key for class C_j at time t could be defined as $K_{j,t} = h(h(X||k_j) \oplus h^t(a) \oplus h^{z-t}(b))$.

It is easy to see that the public information cannot be used by malicious users to misuse their tamper-resistant devices. Indeed, even knowing their secret key k_j and the public information $r_{ij} = h(X||ID_i||ID_j||k_i) \oplus h(X||k_j)$, the users in C_j cannot compute the hash value $h(X||ID_i||ID_j||k_i)$ needed to compute the faked public value r'_{ij}, since they do not know the TA's secret key X, which is stored on the tamper-resistant device.

On the other hand, during the decrypting key derivation, given two distinct classes $C_j \preceq C_i$, the tamper-resistant device held by users in C_i, on inputs r_{ij}, ID_j t, and k_i, can compute the hash value $h(X||k_j) = r_{ij} \oplus h(X||ID_i||ID_j||k_i)$, needed to compute $K_{j,t} = h(h(X||k_j) \oplus h^t(a) \oplus h^{z-t}(b))$.

4.2 A Second Countermeasure

The second countermeasure we propose requires the TA and the tamper-resistant devices held by the users to perform encryptions and decryptions, respectively, by using a secure symmetric key cryptosystem [10]. Let E and D be the encryption and decryption functions of a symmetric cryptosystem. We denote by $E_K(x)$ and $D_K(y)$ the encryption of the plaintext x and the decryption of the ciphertext y under the symmetric key K, respectively. For each pair of distinct classes C_i and C_j such that $C_j \preceq C_i$ and there is no class C_ℓ such that $C_j \preceq C_\ell \preceq C_i$, the TA could compute the public value r_{ij} as $r_{ij} = E_{f(X,ID_i,ID_j,k_i)}(k_j)$, i.e., as the encryption of k_j under a symmetric key $f(X, ID_i, ID_j, k_i)$, where f denotes a suitable function, for example, $f(X, ID_i, ID_j, k_i) = h(X||ID_i||ID_j||k_i)$. Notice that the size of the symmetric key depends on the selected function f. Thus, by appropriately selecting f, several symmetric encryption schemes can be used.

It is easy to see that the public information cannot be used by malicious users to misuse their tamper-resistant devices. Indeed, if the symmetric cryptosystem is secure, users in C_j cannot derive the symmetric key $f(X, ID_i, ID_j, k_i)$ from the knowledge of the plaintext k_j and the ciphertext r_{ij}. Therefore, they cannot compute the faked public value r'_{ij} needed in our attack.

On the other hand, during the decrypting key derivation, given two distinct classes $C_j \preceq C_i$, the tamper-resistant device held by users in C_i, on inputs r_{ij}, ID_j, t, and k_i, can compute the symmetric key $f(X, ID_i, ID_j, k_i)$ and then compute $k_j = D_{f(X,ID_i,ID_j,k_i)}(r_{ij})$ by decrypting r_{ij}, in order to obtain $K_{j,t} = h(k_j \oplus h^t(a) \oplus h^{z-t}(b))$.

Clearly, in addition to computation of hash values, the tamper-resistant device should also be able to perform decryptions. Several symmetric encryption schemes have been designed to run efficiently in a broad range of environments, from smart-cards to programmable gate arrays (for example, the Advanced Encryption Standard (AES) [1]).

4.3 A Third Countermeasure

The third countermeasure we propose consists in authenticating the public information by using a Message Authentication Code (MAC) [10], in order to forbid malicious users to provide forged values to their tamper-resistant devices. More precisely, for each pair of distinct classes C_i and C_j such that $C_j \preceq C_i$ and there is no class C_ℓ such that $C_j \preceq C_\ell \preceq C_i$, the TA could compute a MAC for the message $< r_{ij}||ID_i||ID_j >$, by using a symmetric key $f(X, ID_i, ID_j, k_i)$, where f denotes a suitable function, for example, $f(X, ID_i, ID_j, k_i) = h(X||ID_i||ID_j||k_i)$. Hence, the TA could publish the pair $(<r_{ij}||ID_i||ID_j>, \quad MAC_{f(X,ID_i,ID_j,k_i)}\ (<r_{ij}||ID_i||ID_j>))$.

During the decryption key derivation, users in C_i are also required to input their device with the value $MAC_{f(X,ID_i,ID_j,k_i)}(< r_{ij}||ID_i||ID_j >)$ in addition to the values r_{ij}, ID_j, t, and k_i. Since the TA's secret key X and the identity ID_i are stored on the device, users in C_i can compute $MAC_{f(X,ID_i,ID_j,k_i)}(< r_{ij}||ID_i||ID_j >)$ and check if this coincides with the last input value. In this case the device will continue its computation. On the other hand, if the two values are different, the device will abort its computation. Clearly, in addition to computation of hash values, the tamper-resistant device should also be able to compute a MAC. Several constructions of MACs based on hash functions have been designed (for example, the HMAC [10]).

5 Conclusions

Chien [3] has recently proposed a novel time-bound cryptographic key assignment scheme in a partially-ordered hierarchy. His scheme is based on tamper-resistant devices and requires low computational load and implementation cost. In this paper we have shown how three malicious users may handle public and private information to misuse tamper-resistant devices in order to compute non-authorized encryption keys. We have also shown three countermeasures to withstand the weakness we have exploited.

Acknowledgements

This work was done while the second author was visiting the School of Computer Science at the University of Waterloo, Ontario, Canada. She would like to thank the Department for its hospitality.

References

[1] Advanced Encryption Standard, `http://www.nist.gov/aes/`

[2] S. G. Akl and P. D. Taylor, *Cryptographic Solution to a Problem of Access Control in a Hierarchy*, ACM Trans. Comput. Syst., 1(3), 239–248, 1983.

[3] H. Y. Chien, *Efficient Time-Bound Hierarchical Key Assignment Scheme*, IEEE Trans. on Knowledge and Data Engineering, 16(10), 1301–1034, 2004.

[4] A. De Santis, A. L. Ferrara, and B. Masucci, *Cryptographic Key Assignment Schemes for Any Access Control Policy*, Inf. Process. Lett., 92(4), 199–205, November 2004.

[5] L. Harn and H. Y. Lin, *A Cryptographic Key Generation Scheme for Multilevel Data Security*, Computers and Security, 9(6), 539–546, 1990.

[6] M. S. Hwang, *A Cryptographic Key Assignment Scheme in a Hierarchy for Access Control*, Math. and Comput. Modeling, 26(1), 27–31, 1997.

[7] H. T. Liaw, S. J. Wang, and C. L. Lei, *A Dynamic Cryptographic Key Assignment Scheme in a Tree Structure*, Computers and Math. with Applications, 25(6), 109–114, 1993.

[8] C. H. Lin, *Dynamic Key Management Schemes for Access Control in a Hierarchy*, Computer Communications, 20, 1381–1385, 1997.

[9] S. J. MacKinnon, P. D. Taylor, H. Meijer, and S. G. Akl, *An Optimal Algorithm for Assigning Cryptographic Keys to Control Access in a Hierarchy*, IEEE Trans. Comput., C-34(9), 797–802, 1985.

[10] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied cryptography*, CRC Press, 1996.

[11] R. S. Sandhu, *Cryptographic Implementation of a Tree Hierarchy for Access Control*, Inf. Process. Lett., 27, 95–98, 1988.

[12] X. Yi and Y. Ye, *Security of Tzeng's Time-Bound Key Assignment Scheme for Access Control in a Hierarchy*, IEEE Trans on Knowledge and Data Engineering, 15(4), 1054 –1055, 2003.

[13] W.-G. Tzeng, *A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy*, IEEE Trans. on Knowledge and Data Engineering, 14(1), 182–188, 2002.