Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 569397, 8 pages
http://dx.doi.org/10.1155/2014/569397

*Research Article*

# An Efficient Solution for Hierarchical Access Control Problem in Cloud Environment

**Bing-Zhe He,[1] Chien-Ming Chen,[2,3] Tsu-Yang Wu,[2,3] and Hung-Min Sun[1]**

[1] *Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan*
[2] *Innovative Information Industry Research Center, School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China*
[3] *Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen 518055, China*

Correspondence should be addressed to Hung-Min Sun; hmsun@cs.nthu.edu.tw

Received 19 June 2014; Accepted 16 October 2014; Published 28 October 2014

Academic Editor: Mohamed A. Seddeek



The time-bound hierarchical key assignment scheme provides a cryptographic solution for the access control problem in distributed systems (e.g., Pay-TV and cloud computing applications). Most time-bound hierarchical key assignment schemes can be divided into two types: adopting tamper-resistant devices and utilizing public values. Despite the fact that adopting tamper-resistant devices can easily resist to collusion attacks, utilizing public values is much cheaper and more suitable for cloud environment. In this paper, we proposed a new time-bound hierarchical key assignment scheme, which can effectively defeat the collusion attack. Besides, the proposed scheme utilizes public values instead of tamper-resistant devices, which will restrict user's convenience. Compared with the previous works, our scheme requires fewer public values and has better performance.

## 1. Introduction

The access control problem refers to control who can access the resources in a system. Members in an organization are divided into several classes and each class has different limitations on these resources. For example, in a computer system, an administrator has the right to access all files, including the sensitive files, but a normal user just can access some common files. Nowadays the access control problem is a widespread problem in our world, especially in the distributed environment.

A hierarchical key assignment scheme can provide a cryptographic solution ([1–5]) for the access control problem. In a hierarchical key assignment scheme, resources are encrypted by encrypting keys. Only the user who holds the corresponding encrypting key can access the resources. In addition, the classes will form hierarchical relations between themselves. If two classes have a relation, the user in the higher class can also access the resources in the lower class, but not vice versa. This relation is called partial-order hierarchy. In the previous example, suppose that the administrator belongs to the manager class and normal users belong to the user class. Obviously, these two classes form a hierarchical relation and the manager class has higher right than the user class. The members in the manager class can access the resources in the user class, but normal user cannot access the resources of manager class. However, in some applications such as Pay-TV systems, a user may subscribe to news and sport channels for a week or a month. When the time expires, the user cannot access the channels anymore. Hence, the key assignment scheme needs to consider not only the partial-order hierarchy but also the key update problem when a user leaves the class.

The time-bound hierarchical key assignment protocol is proposed for the above problem. In the time-bound hierarchical key assignment system, the encrypting key for a class is changed as time goes by. According to the user's subscription, the vendor generates his key information that can be used to compute the encrypting key and assigns it to the user. The key information only works in the duration of user's subscription. On the other hand, the user cannot derive the encrypting key except in the duration of his subscription. Since the encrypting key has the time-bound property, we do not need to consider the key update problem when a

user leaves the class. As the previous example, suppose that a user subscribes to the news channels in first time slot and then changes his subscription to the sport channels in fifth time slot. The key information can be used to compute the encrypting key of news channels only between first and fifth time slots. Afterwards, this key information can only be used to compute the encrypting key of sport channels.

The time-bound key assignment protocols can be divided into two types: one is based on tamper-resistant devices ([6, 7]) and the other is based on public values. Tamper-resistant devices can protect the secret information and prevent the secrets from revealing. If the encrypting key is stored in the device, the user is hard to reveal the encrypting key to other users. Despite the fact that the tamper-resistant devices can defeat collusion attacks, applying the tamper-resistant devices requires higher costs and is not suitable for the cloud networks. For this reason, some researches ([8–11]) apply public values instead of tamper-resistant devices. Users can download the public values and derive the encrypting key by his key information and these values.

In this paper, we propose a time-bound hierarchical key assignment scheme which is based on a bilinear pairing function. Due to the time-bound property, the user can subscribe to some classes in a certain period of time. Besides, our scheme utilizes public values instead of tamper-resistant devices. Utilizing public values is more suitable for cloud computing since cloud computing emphasizes that users can access resources anywhere through the Internet. If a cloud service requires tamper-resistant devices, this will restrict users' convenience. On the other hand, the public values can be downloaded anywhere from the cloud. A user can download and use these public values to derive the encrypting key anywhere. Moreover, the number of public values in our scheme is independent of the length of system life time or the number of classes. Compared to the previous works, the proposed scheme has few numbers of the public values and does not need the special requirement for constructing partial-order hierarchy.

The rest of the paper is organized as follows. In Sections 2 and 3, we introduce previous works and present the necessary preliminaries. The proposed scheme is described in Section 4. Then we provide the performance and security analysis of the proposed scheme in Section 5. Finally, we summarize our results.

## 2. Related Work

With the rapid growth of network technology, security issues have been a matter of concern in various network environments ([12–17]) such as wireless sensor networks, social networks, and Internet of things. In this paper, we put emphasis on access control problems in cloud environment.

In 1983, Akl and Taylor [1] first studied the access control problem in a hierarchy and proposed a cryptographic solution for this problem. Then, many researchers also studied this problem and proposed their solutions ([2–5]). However, these schemes do not consider that a user may belong to some classes only in a certain period of time. To solve this problem, Tzeng [8] proposed a time-bound hierarchical key assignment scheme based on Lucas function and RSA problem in 2002. Afterwards, many researchers have concentrated on proposing the time-bound key assignment schemes that either have better performance or can resist collusion attacks. These schemes can be divided into tamper-resistant devices based and public values based schemes. As the tamper-resistant device based schemes, Chien [6] presented his time-bound protocol in 2004 and his scheme is insecure against collusion attacks [18, 19]. Then Bertino et al. [7] also proposed an efficient time-bound hierarchy key management scheme which is based on elliptic curve. However, Sun et al. [20] also show that Bertino et al.'s scheme [7] is insecure against collusion attack and provided the improved scheme. As the public value based scheme, Yeh [11] proposed their public value based protocol in 2005. In 2006, Ateniese et al. [9] not only showed that Yeh's scheme is vulnerable to collusion attacks, but also introduced two different constructions of time-bound key assignment scheme in a hierarchy. Additionally, they also proved that these schemes are practical and provable-secure. Furthermore, de Santis et al. [10] showed how to construct a provable-secure time-bound hierarchy key assignment protocol and compared their protocol with other pervious works.

## 3. Preliminaries

In this section, we introduce some preliminaries about the proposed scheme, before describing our protocol.

*3.1. Bilinear Mapping.* Suppose that $G_1$ and $G_2$ are two cyclic groups with a prime order $q$, where $G_1$ is an additive cyclic group and $G_2$ is a multiplicative cyclic group. A bilinear mapping is a mapping $\hat{e} : G_1 \times G_1 \rightarrow G_2$ with the following properties:

(1) bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, for all $P, Q \in G_1$ and all $a, b \in F_q^*$;

(2) nondegenerative: $\hat{e}(P, Q) = 1, \forall Q \in G_1 \Leftrightarrow P = \infty$;

(3) computable: there exists an efficient algorithm to compute $\hat{e}(P, Q), \forall P, Q \in G_1$.

*3.2. Partial-Order Hierarchy.* In a partial-order hierarchy, a class $C_i$ represents a collection of some resources. Besides, there exists a binary relation "$\prec$" which partially orders these classes. For any two classes $C_i$ and $C_j$, "$C_j \prec C_i$" means that $C_i$ dominates $C_j$ and the security level of $C_i$ is higher than of $C_j$. In other words, the users of $C_i$ can access the resources in $C_j$ but not vice versa. For example, assume that $C_1 \prec C_3 \prec C_4$ and $C_2$ is an independent class. If a user belongs to $C_4$, this means that the user can access all of the resources in $C_1$, $C_3$, and $C_4$, but he does not have the access right of $C_2$. In other words, the user only holds the encrypting keys for $C_1$, $C_3$, and $C_4$. Moreover, we usually use a directed acyclic graph $G = (V, E)$ to represent a partial-order hierarchy, where $V$ denotes the set of classes and $E$ denotes the set of partial-order relations.

*3.3. Time-Bound Property.* A key assignment system with the time-bound property means that the encrypting keys in a class are different as time goes by. The user can only derive the encrypting keys that are within the duration of his subscription. Assume that a user belongs to $C_i$ from $t_1$ to $t_2$ and $C_j \prec C_i$. The user can just derive the encrypting key $K_i$ and $K_j$ at $t$ if and only if $t_1 < t < t_2$. Nevertheless, he cannot derive the encrypting key at $t_3$ when $t_3 \notin [t_1, t_2]$.

## 4. The Proposed Scheme

The detail of the proposed scheme is introduced in this section. The proposed scheme consists of three phases: initialization, user subscription, and encrypting key derivation. Each phase is described in the following. Then, we use a concrete example to explain our scheme. Finally, the notation is shown in the notation section.

*4.1. Initialization.* In this phase, we suppose that the vendor has already constructed a partial-order hierarchy. The system parameters are initialized as following steps.

(1) The vendor chooses an elliptic curve $E$ over a finite field $F_q$ and then selects a generating point $P \in E(F_q)$, where the order of $P$ is $n$.

(2) Afterwards, the vendor constructs a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$.

(3) Suppose that the maximum duration of each subscription for a user is $l$ and the system lifetime is $z$, where $l < z$. The system selects two random values $a$ and $b$ from $Z_q^*$ and generates $B_{x,y} = a^x b^y P$, where $x \in [1, l]$ and $y \in [0, x]$.

(4) For every class $i$, the vendor randomly generates $g_i$ and $h_i$, where $g_i h_i \equiv 1 \bmod n$. Then, the vendor computes the public values $r_{i,j} = h_i g_j \bmod n$ if and only if $C_j \prec C_i$.

(5) Finally, the public system parameters are $\{E, G_1, G_2, P, \hat{e}, H\}$ and the vendor keeps $\{a, b, g_i, h_i\}$ in secret. Besides, the encrypting key is computed by $K_{i,t} = \hat{e}(P, P)^{g_i a^t b^{z-t}}$.

When completing these steps, the vendor publishes $r_{i,j}$ and $B_{x,y}$ on an authenticated board. These public values can be downloaded through the Internet and used to compute the encrypting key.

*4.2. User Subscription.* In this phase, the system generates the key information for a user according to his subscription request. Then this key information is issued to the user through a secure channel. If a user subscribes class $C_i$ from $t_1$ to $t_2$, the key information $K_{i,t_1,t_2}$ can be computed by $K_{i,t_1,t_2} = g_1 a^{t_1} b^{z-t_2} P$.

Afterwards, the user uses $K_{i,t_1,t_2}$ and $B_{x,y}$ to compute the encrypting key $K_{i,t}$ at time slot $t$ if $t \in [t_1, t_2]$. Moreover, if $C_j$ and $C_k$ are dominated by $C_i$ ($C_j, C_k \prec C_i$), the user can use $K_{i,t_1,t_2}$, $B_{x,y}$, and $r_{i,j}$ (or $r_{i,k}$) to compute the encrypting key of class $C_j$ (or $C_k$) from $t_1$ to $t_2$. Otherwise, the system has to issue $K_{j,t_1,t_2}$ and $K_{k,t_1,t_2}$ to the user.

*4.3. Encrypting Key Derivation.* In this phase, we show how a user derives an encrypting key. This phase can be divided into two cases: the class which is not dominated by any other classes and the class which is dominated by some other classes.

*Case 1.* Suppose that a user subscribes to class $C_i$ which is not dominated by other classes. The user can use $K_{i,t_1,t_2}$ and $B_{x,y}$ to compute $K_{i,t}$, where $t \in [t_1, t_2]$ and $t_1 + x = t = t_2 - y$. The encrypting key for the class $C_i$ at $t$ can be computed as follows:

$$K_{i,t} = e\left(K_{i,t_1,t_2}, B_{x,y}\right). \tag{1}$$

*Case 2.* Suppose that the user subscribes to $C_i$ and $C_j$ is dominated by $C_i$ ($C_j \prec C_i$). In order to derive the encrypting key for $C_j$, the user first computes $Q = K_{i,t_1,t_2} \times r_{i,j}$. Then, the encrypting key for $C_j$ can be computed as follows:

$$K_{j,t} = \hat{e}\left(Q, B_{x,y}\right). \tag{2}$$

Now, we show the correctness of equations in both cases. The temporal encrypting key for class $C_i$ in time slot $t$ is $K_{i,t} = g_i a^t b^{z-t}$, where $t_1 + x = t = t_2 - y$.

The following is the correctness of Case 1:

$$\begin{aligned} K_{i,t} &= \hat{e}\left(K_{i,t_1,t_2}, B_{x,y}\right) \\ &= \hat{e}\left(g_i a^{t_1} b^{z-t_2} P, a^x b^y P\right) \\ &= \hat{e}(P, P)^{g_i a^{t_1+x} b^{z-t_2+y}} \\ &= \hat{e}(P, P)^{g_i a^t b^{z-t}}. \end{aligned} \tag{3}$$

The correctness of Case 2 is shown as follows:

$$\begin{aligned} K_{j,t} &= \hat{e}\left(K_{i,t_1,t_2} \times r_{i,j}, B_{x,y}\right) \\ &= \hat{e}\left(g_i a^{t_1} b^{z-t_2} P \times h_i g_j, a^x b^y P\right) \\ &= \hat{e}\left(g_j a^{t_1} b^{z-t_2} P, a^x b^y P\right) \\ &= \hat{e}(P, P)^{g_j a^{t_1+x} b^{z-t_2+y}} \\ &= \hat{e}(P, P)^{g_j a^t b^{z-t}}. \end{aligned} \tag{4}$$

*4.4. Example.* Here, we use an example to describe the proposed scheme. Suppose that system lifetime is 70 and the maximum duration of each subscription is 5. Therefore, we set $z$ to be 70 and $l$ to be 5 here. As shown in Figure 1, the partial-order hierarchy has five classes $C_1$ to $C_5$. Besides, $C_1$ dominates $C_2$, $C_3$, and $C_4$. The vendor generates random values $g_i$ for each class and computes $r_{i,j} = h_i g_j \bmod n$ if $C_i$ dominates $C_j$. In this example, the public values $r_{i,j}$ are $r_{1,2}$, $r_{1,3}$, $r_{1,4}$, $r_{2,4}$, and $r_{3,4}$. Then, the vendor publishes $\{E, G_1, G_2, P, \hat{e}, H\}$ and keeps $\{a, b, h_i, g_i\}$ in secret, where $E$

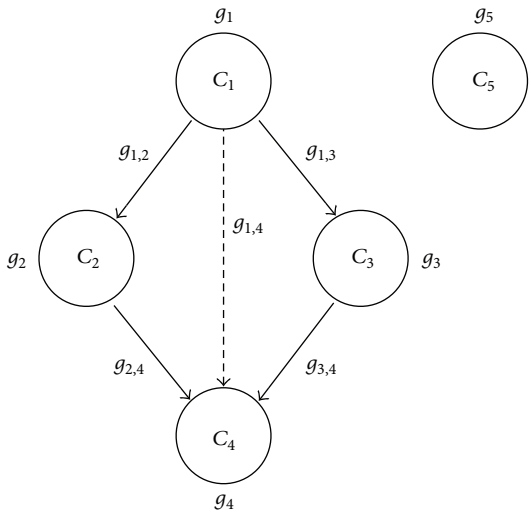


Figure 1: An example for the proposed scheme.

is over $F_q$, $a, b \in Z_p^*$, $\hat{e} : G_1 \times G_1 \rightarrow G_2$ and $P$ is a generating point with order $n$. Moreover, the public value $B_{x,y}$ is shown as follows:

$$\begin{aligned} B_{x,y} = \{&B_{1,0}, B_{1,1}, \\ &B_{2,0}, B_{2,1}, B_{2,2}, \\ &B_{3,0}, B_{3,1}, B_{3,2}, B_{3,3}, \\ &B_{4,0}, B_{4,1}, B_{4,2}, B_{4,3}, B_{4,4}, \\ &B_{5,0}, B_{5,1}, B_{5,2}, B_{5,3}, B_{5,4}, B_{5,5}\}, \end{aligned} \tag{5}$$

where $B_{x,y} = a^x b^y P$, $x \in [1, 5]$ and $y \in [0, x]$.

If a user subscribes to $C_2$ and $C_5$ from $t_1$ to $t_2$, then he will hold the public parameters $\{E, P, G_1, G_2, \hat{e}, H\}$, $K_{2,t_1,t_2}$, and $K_{5,t_1,t_2}$. Now suppose the user desires to access the resources in $C_2$ at time slot $t$ and $t_1 + 2 = t = t_2 - 3$. The encrypting key $K_{2,t}$ can be computed as follows:

$$\begin{aligned} K_{2,t} &= \hat{e}\left(g_2 a^{t_1} b^{z-t_2} P, B_{2,3}\right) \\ &= \hat{e}\left(g_2 a^{t_1} b^{z-t_2} P, a^2 b^3 P\right) \\ &= \hat{e}(P, P)^{g_2 a^{t_1+2} b^{z-t_2+3}} \\ &= \hat{e}(P, P)^{g_2 a^t b^{z-t}} \\ &= K_{2,t}. \end{aligned} \tag{6}$$

In a similar way, the temporal key of $C_5$ at $t$ can be computed by $K_{5,t} = \hat{e}(g_5 a^{t_1} b^{z-t_2} P, B_{2,3})$.

Now assume that the user desires to compute $K_{4,t}$ and $t_1 + 4 = t = t_2 - 1$. The encrypting key $K_{4,t}$ can be computed by the following:

$$\begin{aligned} K_{4,t} &= \hat{e}\left(g_2 a^{t_1} b^{z-t_2} P \times h_2 g_4, B_{4,1}\right) \\ &= \hat{e}\left(g_4 a^{t_1} b^{z-t_2} P, a^4 b^1 P\right) \\ &= \hat{e}(P, P)^{g_4 a^{t_1+4} b^{z-t_2+1}} \\ &= \hat{e}(P, P)^{g_4 a^t b^{z-t}} \\ &= K_{4,t}. \end{aligned} \tag{7}$$

## 5. Analysis

In this section, we not only analyze the security and performance of the proposed scheme but also discuss the applications for the time-bound key assignment scheme in a hierarchy. Compared with previous works, users do not need large storage capacity and perform many times of decryption, and our scheme has fewer broadcasting messages over the network.

*5.1. Security against Possible Attacks.* In the following, the security analysis will be divided into two parts: the security of key information and the security of encrypting key. For convenience, we first define two mathematical assumptions as follows.

(i) Discrete logarithm assumption: given $P, aP \in G_1$ for $a \in F_q^*$, no probabilistic polynomial-time algorithm can compute the value $a$.

(ii) Computational Diffie-Hellman (CDH) assumption: given $P, aP, bP \in G_1$ for $a, b \in F_q^*$, no probabilistic polynomial-time algorithm can compute $abP \in G_1$.

*5.1.1. Security of Key Information.* Now, we consider the two types of attackers: outside and inside attackers.

**Lemma 1.** *Under the computational Diffie-Hellman (CDH) assumption, any outside attacker $A$ cannot compute the key information of some classes in the proposed scheme even if $A$ has obtained all public information $r_{i,j} = h_i g_j$ and $B_{x,y} = a^x b^y P$.*

*Proof.* Without loss of generality, we assume that $A$ wants to compute the key information $K_{i,t_1,t_2} = g_i a^{t_1} b^{z-t_2} P$ of class $C_i$ from $t_1$ to $t_2$. Since $g_i$ and $h_j$ are secret values held by vendor, $A$ cannot obtain the individual values $g_i$ and $h_j$ from $r_{i,j}$. Meanwhile, attacker $A$ cannot obtain the values $a^x$ and $b^y$ under the CDH assumption. Hence, any outside attacker is infeasible to compute key information of some classes in the proposed scheme. □

**Lemma 2.** *In the proposed scheme, any inside attackers (malicious subscribers) cannot compute unauthorized key information of some class.*

*Proof.* Without loss of generality, we consider the following two cases to prove this lemma.

*Case I.* We assume that a subscriber $A$ in class $C_j$ from $t_1$ to $t_2$ tries to compute unauthorized key information $K_{i,t_1,t_2}$ of class $C_i$ for some index $i$ and $C_j \not\prec C_i$. If $A$ wants to compute $K_{i,t_1,t_2}$, he must find a value $r' = g_i h_j$ such that $r' \cdot K_{j,t_1,t_2} = (g_i \cdot h_j) \cdot g_j a^{t_1} b^{z-t_2} P = K_{i,t_1,t_2}$. Since $g_i$ and $h_j$ are secret values held by vendor, $A$ cannot compute the value $r' = g_i h_j$. Thus, it is infeasible to compute unauthorized key information $K_{i,t_1,t_2}$ from $K_{j,t_1,t_2}$ for the case $C_j \not\prec C_i$.

*Case II.* We assume that a subscriber $A$ in class $C_j$ from $t_1$ to $t_2$ tries to compute unauthorized key information $K_{i,t_1,t_2}$ of class $C_i$, where $C_j \prec C_i$. By similar way to Case I, $A$ must find a value $r'$ such that $r' \cdot K_{j,t_1,t_2} = K_{i,t_1,t_2}$. Although $A$ owns the value $r_{i,j} = h_i g_j$, he still cannot compute $r' = g_i h_j$ by the same reason in Case I. Thus, it is infeasible to compute $K_{i,t_1,t_2}$ from $K_{j,t_1,t_2}$ for the case $C_j \prec C_i$. □

**Lemma 3.** *In the proposed scheme, any inside attackers (malicious subscribers) cannot collude to compute unauthorized key information in some class.*

*Proof.* Without loss of generality, we consider the following two cases to prove this lemma.

*Case I.* We assume that two subscribers $A$ and $B$ collude to compute the unauthorized key information $K_{i,t_1,t_2}$ in class $C_i$ from $t_1$ to $t_2$, where $A$ subscribes to $C_j$ from $t_1$ to $t_2$ and $B$ subscribes to $C_k$ from $t_1$ to $t_2$ with $C_j \prec C_i$, $C_k \prec C_i$, and $C_j \not\prec C_k$. As mentioned in Lemma 2, two malicious subscribers must find a value $r' = g_i h_j$ such that $r' \cdot K_{j,t_1,t_2} = K_{i,t_1,t_2}$ or a value $r'' = g_i h_k$ such that $r'' \cdot K_{k,t_1,t_2} = K_{i,t_1,t_2}$. Even if they have $r_{i,j} = h_i g_j$ and $r_{i,k} = h_i g_k$, they still cannot compute $r'$ and $r''$. Since $g_i$ and $h_j$ are secret values held by vendor, they cannot obtain the individual values $g_i$ and $h_j$. Thus, it is infeasible to compute $K_{i,t_1,t_2}$ from $t_1$ to $t_2$ with $C_j \prec C_i$, $C_k \prec C_i$, and $C_j \not\prec C_k$.

*Case II.* We assume that three subscribers $A$, $B$, and $C$ collude to compute the unauthorized key information $K_{j,t_2,t_3}$ in class $C_j$ from $t_2$ to $t_3$, where $A$ subscribes to $C_i$ from $t_1$ to $t_2$, $B$ subscribes to $C_j$ from $t_3$ to $t_4$, and $C$ subscribes to $C_k$ from $t_5$ to $t_6$ with $C_k \prec C_j \prec C_i$ and $t_5 \le t_2 < t_3 \le t_6$. If they want to compute $K_{j,t_2,t_3} = g_j a^{t_2} b^{z-t_3} P$, they must find the two values $a^{t_2-t_3}$ and $b^{t_3-t_4}$ such that $a^{t_2-t_3} \cdot b^{t_3-t_4} \cdot K_{j,t_3,t_4} = g_j a^{t_2} b^{z-t_3} P$. However, it is infeasible to find $a^{t_2-t_3}$ and $b^{t_3-t_4}$ from $K_{i,t_1,t_2}$, $K_{j,t_3,t_4}$, and $K_{i,t_5,t_6}$ under the discrete logarithm and the computational Diffie-Hellman assumptions. □

By Lemmas 1–3, we can obtain Theorem 4.

**Theorem 4.** *Under the discrete logarithm and the computational Diffie-Hellman assumptions, any attackers (including inside and outside) cannot compute the unauthorized key information of some class $C_i$ in the proposed scheme.*

Table 1: The space complexity.

| Scheme | Public value | Private parameters | Key information |
|---|---|---|---|
| Ateniese et al. [9] (symmetric) | $O(\lvert V\rvert^2 \lvert T\rvert^3)$ | $O(\lvert V\rvert \lvert T\rvert)$ | $O(C)$ |
| Ateniese et al. [9] (pairing) | $O(\lvert V\rvert^2)$ | $O(\lvert V\rvert \lvert T\rvert)$ | $O(\lvert\widehat{T}\rvert)$ |
| Tzeng [8] | $O(\lvert V\rvert)$ | $O(\lvert V\rvert + \lvert T\rvert)$ | $O(C)$ |
| Bertino et al. [7] | $O(\lvert E\rvert)$ | $O(\lvert V\rvert)$ | $O(C)$ |
| Our scheme | $O(\lvert\widehat{T}\rvert^2 + \lvert E\rvert)$ | $O(\lvert E\rvert + \lvert V\rvert)$ | $O(C)$ |

*5.1.2. Security of Temporal Encryption Key.* The security of temporal encryption key is relying on the security of key information. Hence, we have the following result.

**Theorem 5.** *Under the discrete logarithm and the computational Diffie-Hellman assumptions, any attacker (including inside and outside) cannot compute the unauthorized temporal encryption key to access some class $C_i$ in the proposed scheme.*

*5.2. Performance Evaluation.* The performance of our scheme is evaluated in terms of storage requirements and computation costs. The storage requirements consist of three parts: private parameters, public values, and key information. These public values, including $B_{x,y}$, $r_{i,j}$, and $\{E, P, G_1, G_2, \hat{e}, H\}$, are published on an authentic board. Since $\{E, P, G_1, G_2, \hat{e}, H\}$ does not affect the storage complexity, we only discuss $B_{x,y}$ and $r_{i,j}$. All users can download and store these public values. Then, a user can use his key information and these public values to compute the encrypting key for accessing the resources. The private parameters are security parameters in the proposed scheme and are kept secret in the server. The vendor uses these private parameters to generate all key information and encrypting keys. The private parameters include $g_i$, $a$, and $b$. Finally, the key information is generated according to a user's subscription.

Table 1 shows the comparison of storage requirements between our scheme and other previous works. In the table, $|T|$ means the system lifetime and the maximum duration of each subscription for a user is $|\widehat{T}|$. We also denote the number of edges and classes in $G$ by $|E|$ and $|V|$.

In the proposed scheme, the vendor randomly selects secret values $g_i$ for each class in the partial-order hierarchy, where $g_i$ can be used to generate $K_i$ and $K_{i,t_1,t_2}$. Therefore, the space complexity of the private parameters in the server side is $O(|V|)$. Now, we consider the space complexity of the public values in the client side. The public values in our scheme are $B_{x,y}$ and $r_{i,j}$. Since the maximum duration of each subscription for a user is $|\widehat{T}|$, the number of $B_{x,y}$ is equal to $(|\widehat{T}|(|\widehat{T}| + 3))/2$ and the number of $r_{i,j}$ is $|E|$. Consequently, the space complexity of public values is $O(|\widehat{T}|^2 + |E|)$. In fact, the storage requirements in our protocol are irrelevant to $|T|$. Otherwise, if the storage requirements are related to $|T|$, the space complexity will rise dramatically when $|T|$ is very large. Compared to other schemes (as shown in Table 1),

our scheme has better performance on the space complexity and the space complexity is irrelevant to $|T|$. Although the number of public values in Bertino et al.'s scheme [7] is fewer than in our scheme, Bertino et al.'s protocol requires tamper-resistant devices, which require extra costs in the deployment phase and are not suitable for cloud computing.

The number of key information in our scheme depends on partial-order hierarchy and the number of classes which a user subscribes to. After a user registers to the system, the server will issue some key information to the user. Unlike some works ([9, 21]) which require that a user only belongs to one class, our scheme allows that a user can subscribe to many classes. The worst case is that every class is irrelevant to the other classes. In this case, the space complexity of the key information for a user is $O(|V|)$. However, generally, the number of key information for a user is equal to a constant number.

We use a concrete example to show the space requirements for the proposed scheme. First, we suppose that each time slot is one day. Then, we set the system lifetime to be 10 years ($z = 1 \times 365 \times 10 = 3650$) and the maximum duration of each subscription for a user is one month ($l = 1 \times 30 = 30$). Finally, we assume that there are 250 channels in the system. To put it simply, we set $|V|$ and $|E|$ to be both equal to 300 in the partial-order hierarchy. According to the previous analysis, the space requirements for the public values are related to the number of $B_{x,y}$ and $r_{i,j}$. Since the number of $B_{x,y}$ is equal to 495 and the number of $r_{i,j}$ is 300, we can compute that the space requirements for the public values are $(495 + 300) \times 160$ bits $\cong$ 16 KB. The space requirement for the private parameters depends on the number of $g_i$. Hence the space complexity for the private parameters is equal to $300 \times 160$ bits $\cong$ 6 KB. Finally, the size of each key information is 160 bits.

Now, we consider the computation cost of encrypting key derivation. In our scheme, a user only computes two pairing operations at most when he derives the encrypting key. In [22], the results show that the computation cost of pairing operation for a smartphone (HTC Desire HD A9191, Android 2.2) is affordable. Therefore, it is feasible that our scheme can be executed on low-power devices such as phone and set-top box.

*5.3. Application.* Cloud computing means that applications migrate from local PCs to Internet and sometimes is referred to as Software as a Service (SaaS) [23, 24]. Users can obtain the computing and storage capacities through Internet. Users pay for the network traffic or CPU utilization time instead of paying for software. When a cloud is only made available to some specific members in an organization, this cloud is called private cloud and only authorized users can access the cloud. For example, enterprises usually construct private data storage service for the employees.

Since security threats can influence the development of cloud computing, many security issues are discussed [25], especially the access control problem. In this section, we introduce two examples, data storage service and video-on-demand (VoD) service, and explain the access control problem in both examples. First application, cloud storage service, allows users to store their files on the remote servers and share their files with other users. Obviously, enterprises can gain great benefits from cloud storage services. However, enterprises always construct their own private cloud services instead of public cloud services because of the privacy and security consideration. In an enterprise or a government, data are always classified into several classes, for example, $C_1$ to $C_4$, and members are also categorized into these classes. Assume that $C_1 \prec C_2 \prec C_3 \prec C_4$, where $C_1 \prec C_2$ means that a user which belongs to $C_2$ can access the data in $C_1$ and $C_2$. In this case, a user which belongs to $C_1$ cannot access the data in the $C_2$, $C_3$, and $C_4$. Therefore, we can apply a key assignment scheme to solve the access control problem. A key assignment scheme can distribute encrypting keys to each member according to their access rights in the organization. Unfortunately, a user may only subscribe to some classes for a certain period of time. Traditional key assignment protocol cannot satisfy the time-bound requirement, but the proposed scheme can solve this problem easily. Besides, utilizing public values is more suitable for cloud computing than adopting tamper-resistant devices. Cloud computing claims that users can access the resources anywhere without any limitations, but adopting tamper-resistant devices in cloud services will restrict users' convenience. We use an example to explain how to apply the proposed scheme to enhance the cloud service. For example, an employee may become an agent of his manager when his manager takes a vacation. Suppose that the manager belongs to $C_3$ and the employee belongs to $C_1$. In addition, we also assume that the manger is on vacation from $t_5$ to $t_8$. Hence, the manager only needs to escalate the privilege of his agent into $C_3$ when he is on a vacation ($t_5$ to $t_8$) and our time-bound protocol can easily achieve this requirement. The manager just gives the agent the key information, $K_{3,t_5,t_8}$. This means that the agent can only access the data which belong to $C_3$ from $t_5$ to $t_8$. On the other hand, the agent cannot derivate the encrypting key of $C_3$ except for the specific periods of time ($t_5$ to $t_8$).

The second application is the video-on-demand (VoD) system. Recently, many researchers discuss how to utilize cloud services to support large-scale Internet-based applications such as video-on-demand (VoD) [26, 27]. In VoD systems, users can watch video content on demand. Video content can be either streamed or downloaded through a set-top box, a computer, or a mobile device. Users can subscribe to the content that they like and access the content in the duration of subscription. Video content is always encrypted in VoD systems and users must use his encrypting key to access the content. In addition, a user usually subscribes to a program only for a limited period such as a week or a month. Beyond the duration of subscription, users are not allowed to access the content. The proposed time-bound protocol can be easily deployed into VoD systems to manage the users' subscription. In the initial phase, the vender organizes all contents into several classes and constructs a partial-order hierarchy. For every class, the encrypting key will be used to protect the content and be changed as time goes by. A user in the system will obtain some key information according to his subscription. Finally, the user can compute the correct

encrypting key through his key information and public values to access the encrypted contents.

## 6. Conclusion

We have presented a time-bound key assignment scheme for a partial-order hierarchy. Since our scheme has the time-bound property, the vendor can offer great flexibility in the user subscription. Each user can only use his key information to compute the corresponding encrypting key in the duration of his subscription. Therefore our scheme can easily solve the key update problem without higher costs. In addition, our scheme applies public values instead of tamper-resistant devices, which is more suitable for cloud computing. Unlike previous schemes, the number of public values or key information does not depend on the length of system time. As a result, our scheme has lower space complexity and acceptable performance compared with previous works. Moreover, we also present that our scheme can defeat the collusion attacks.

## Notation Used in Our Scheme

$E$: The elliptic curve that the vendor selects
$\hat{e}$: A bilinear mapping function, where $\hat{e} : G_1 \times G_1 \rightarrow G_2$
$H$: An one-way hash function
$G_1$: Additive cyclic group
$G_2$: Multiplicative cyclic group
$P$: Generating point of $E$ with large order $n$
$a, b$: Secret random values
$C_i$: The secure class $i$
$g_i$: Secret value of $C_i$
$K_{i,t_1,t_2}$: User's key information which can be used to compute encrypting key of $C_i$ from $t_1$ to $t_2$
$K_{i,t}$: Temporal encrypting key of $C_i$ at time $t$
$r_{i,j}$: Public values and only used in the improved scheme, where $r_{i,j} = g_i h_i \bmod n$ if $C_j \prec C_i$
$B_{x,y}$: Public values, where $B_{x,y} = a^x b^y P$, $x \in [1, l]$ and $y \in [0, x]$.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The work of Chien-Ming Chen was supported in part by the Project HIT.NSRIF.2014098 supported by Natural Scientific Research Innovation Foundation in Harbin Institute of Technology and in part by Shenzhen Strategic Emerging Industries Program under Grant ZDSY20120613125016389. The work of Hung-Min Sun was supported in part by the National Science Council, Taiwan, under Grant NSC 101–2221-E-007-026-MY3.

## 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, 1983.

[2] F. Kuo, V. Shen, T. Chen, and F. Lai, "Cryptographic key assignment scheme for dynamic access control in a user hierarchy," *IEE Proceedings—Computers and Digital Techniques*, vol. 146, no. 5, pp. 235–240.

[3] 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.

[4] A. de Santis, A. L. Ferrara, and B. Masucci, "Cryptographic key assignment schemes for any access control policy," *Information Processing Letters*, vol. 92, no. 4, pp. 199–205, 2004.

[5] I.-C. Lin, M.-S. Hwang, and C.-C. 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.

[6] H.-Y. Chen, "Efficient time-bound hierarchical key assignment scheme," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 10, pp. 1301–1304, 2004.

[7] E. Bertino, N. Shang, and S. S. Wagstaff Jr., "An efficient time-bound hierarchical key management scheme for secure broadcasting," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 2, pp. 65–70, 2008.

[8] 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, 2002.

[9] G. Ateniese, A. de Santis, A. L. Ferrara, and B. Masucci, "Provably-secure time-bound hierarchical key assignment schemes," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 288–297, November 2006.

[10] A. de Santis, A. L. Ferrara, and B. Masucci, "New constructions for provably-secure time-bound hierarchical key assignment schemes," *Theoretical Computer Science*, vol. 407, no. 1–3, pp. 213–230, 2008.

[11] J.-H. Yeh, "An RSA-based time-bound hierarchical key assignment scheme for electronic article subscription," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM '05)*, pp. 285–286, November 2005.

[12] C.-M. Chen, K.-H. Wang, T.-Y. Wu, J.-S. Pan, and H.-M. Sun, "A scalable transitive human-verifiable authentication protocol for mobile devices," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1318–1330, 2013.

[13] Y. Liu, C.-C. Chang, and S.-C. Chang, "A secure and efficient $t$-out-of-$n$ oblivious transfer based on the generalized aryabhata remainder theorem," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 2, pp. 223–233, 2014.

[14] T.-Y. Wu and Y.-M. Tseng, "Publicly verifiable multi-secret sharing scheme from bilinear pairings," *IET Information Security*, vol. 7, no. 3, pp. 239–246, 2013.

[15] E. K. Wang, Y. Ye, and X. Xu, "Location-based distributed group key agreement scheme for vehicular AD hoc network," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 759601, 8 pages, 2014.

[16] C.-M. Chen, Y.-H. Lin, Y.-C. Lin, and H.-M. Sun, "RCDA: recoverable concealed data aggregation for data integrity in

wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, pp. 727–734, 2012.

[17] D. Taghaddos and A. Latif, "Visual cryptography for gray-scale images using bit-level," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 1, pp. 90–97, 2014.

[18] 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, 2003.

[19] X. Yi, "Security of Chien's efficient time-bound hierarchical key assignment scheme," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 9, pp. 1298–1299, 2005.

[20] H.-M. Sun, K.-H. Wang, and C.-M. Chen, "On the security of an efficient time-bound hierarchical key management scheme," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, no. 2, pp. 159–160, 2009.

[21] S.-Y. Wang and C.-S. Laih, "Merging: an efficient solution for a time-bound hierarchical key assignment scheme," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 1, pp. 91–100, 2006.

[22] A. de Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proceedings of the 16th IEEE Symposium on Computers and Communications (ISCC '11)*, pp. 850–855, July 2011.

[23] B. Hayes, "Cloud computing," *Communications of the ACM*, vol. 51, no. 7, pp. 9–11, 2008.

[24] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[25] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.

[26] B.-J. Sun and K.-J. Wu, "Research on cloud computing application in the peer-to-peer based video-on-demand systems," in *Proceedings of the 3rd International Workshop on Intelligent Systems and Applications (ISA '11)*, pp. 1–4, Wuhan, China, May 2011.

[27] Y. Wu, C. Wu, B. Li, X. Qiu, and F. C. M. Lau, "CloudMedia: when cloud on demand meets video on demand," in *Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS '11)*, pp. 268–277, Minneapolis, Minn, USA, June 2011.