CERIAS Tech Report 2008-15
IEEE Transactions on Dependable and Secure Computing
by Elisa Bertino, Ning Shang, Samuel S. Wagstaff, Jr.
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

An Efficient Time-Bound Hierarchical Key Management Scheme for Secure Broadcasting

Elisa Bertino, *Fellow, IEEE, ACM,* Ning Shang, *Member, AMS,* Samuel S. Wagstaff, Jr., *Member, AMS*

***Abstract*— In electronic subscription and pay TV systems, data can be organized and encrypted using symmetric key algorithms according to predefined time periods and user privileges, then broadcast to users. This requires an efficient way to manage the encryption keys. In this scenario, time-bound key management schemes for a hierarchy were proposed by Tzeng and Chien in 2002 and 2005, respectively. Both schemes are insecure against collusion attacks. In this paper, we propose a new key assignment scheme for access control which is both efficient and secure. Elliptic curve cryptography is deployed in this scheme. We also provide analysis of the scheme with respect to security and efficiency issues.**

***Index Terms*— Secure broadcasting, time-bound hierarchical key management, elliptic curves, elliptic curve discrete logarithm problem(ECDLP).**

I. INTRODUCTION

In a web-based environment, the data to be securely broadcast, e.g., electronic newspapers or other types of content, can be organized as a hierarchical tree and encrypted by distinct cryptographic keys according to access control policies. We need a key management scheme so that a higher class can retrieve data content that a lower class is authorized to access but not vice versa. In many applications (e.g., electronic newspaper/journal subscription, pay TV broadcasting), there is a time bound associated with each access control policy, so that a user is assigned to a certain class for just a period of time. The users' keys need to be updated periodically to ensure that the delivery of the information follows the access control policies of the data source. An ideal time-bound hierarchical key management scheme should be able to perform the above task in an efficient fashion and minimize the storage and communication of keys. In 2002, W.G. Tzeng attempted to solve this problem in [11]. Tzeng's scheme is efficient in terms of its space requirement, but is computationally inefficient, since a Lucas function operation is used to construct the scheme, and this incurs heavy computational load. Moreover, it is insecure against collusion attacks as shown by Yi and Ye [13].

Another time-bound hierarchical key assignment scheme, based on a tamper-resistant device and a secure hash function, was proposed by H.Y. Chien [5] in 2004. This scheme greatly reduces computational load and implementation cost. However, it has a security hole against X. Yi 's three-party collusion attack [12]. Inspired by Chien's idea, we propose in this paper a new method for access control using elliptic curve cryptography. This scheme is efficient and secure against X. Yi 's three-party collusion attack.

E. Bertino is with the CERIAS and the Department of Computer Sciences, Purdue University. Email: bertino@cs.purdue.edu

N. Shang is with the CERIAS, the Department of Mathematics and the Department of Electrical and Computer Engineering, Purdue University. Email: nshang@math.purdue.edu

S. S. Wagstaff, Jr. is with the CERIAS and the Department of Computer Sciences, Purdue University. Email: ssw@cerias.purdue.edu

Although there have been attacks on smart cards [2] and some other tamper-resistant devices, such attacks require special equipment which would cost more than a subscription. The only really valuable data on the smart cards our scheme uses is the master key. It must be kept secret because an attacker who obtained it could derive all the keys for the data that one could get with this smart card. Assuming the master key can be protected, there is good reason to believe that our scheme that uses tamper-resistant devices can have practical important applications, in areas such as digital right management.

Our original motivation for this paper was to provide a better key management scheme for [4], in which data are encoded in XML and need to be securely broadcast, but a solution to the key management scheme fails in terms of efficiency and security.

The rest of this paper is organized as follows: Section II presents the notation and definitions needed to give a hierarchical structure to the data source. Section III proposes the new time-bound key management scheme applied to a hierarchy. Section IV contains further discussion of the key management scheme. Section V summarizes our results.

II. DEFINITIONS AND NOTATION

Let $\mathcal{S}$ be the data source to broadcast. We assume $\mathcal{S}$ is partitioned into blocks of data called *nodes*.

The policy base $\mathcal{PB}$ is the set of access control policies defined for $\mathcal{S}$. In our setting, each access control policy $\mathsf{acp} \in \mathcal{PB}$ contains a temporal interval I among its components, which specifies the time period in which the access control policy is valid. A sample access control policy for XML documents might look like

$$\mathsf{acp} = (\mathsf{I}, \mathsf{P}, \mathsf{sbj\text{-}spec}, \mathsf{prot\text{-}obj\text{-}spec}, \mathsf{priv}, \mathsf{prop\text{-}opt}),$$

where I, P, $\mathsf{sbj\text{-}spec}$, $\mathsf{prot\text{-}obj\text{-}spec}$, priv and $\mathsf{prop\text{-}opt}$ are the temporal interval, the periodic expression, the credential specification, the protection object specification, the privilege and the propagation option of acp, respectively. Interested readers may refer to [3] and [4] for details.

It is important to notice that several policies may apply to each node in $\mathcal{S}$. In what follows we refer to the set of policies applying to a node in $\mathcal{S}$ as the **policy configuration** associated with the node. Also, in what follows $\mathcal{PC}_{\mathcal{PB}}$ denotes the set of all possible policy configurations which can be generated by policies in $\mathcal{PB}$.

We now introduce the notion of a class of nodes, a relevant notion in our approach. Intuitively, a class of nodes corresponds to a given policy configuration and identifies all nodes to which such configuration applies. Intuitively, a class of nodes includes the set of nodes to which the same set of access control policies apply.

Definition 1 (Class of nodes): Let Pc_i be a policy configuration belonging to $\mathcal{PC}_{\mathcal{PB}}$. The **class of nodes marked with** Pc_i,

denoted by $\mathcal{C}_i$, is the set of nodes belonging to the data source $\mathcal{S}$ marked by all and only the policies in Pc_i. Note that the empty set could be a class of nodes marked with a certain policy configuration. We denote by $\mathcal{C}$ the set of all classes of nodes defined over $\mathcal{S}$ marked with the policy configurations in $\mathcal{PC}_{\mathcal{PB}}$, and we have the following requirement: we distinguish and include in $\mathcal{C}$ the empty sets, if marked by policy configurations consisting of only one access control policy, and exclude from $\mathcal{C}$ the empty sets marked by any other policy configurations. Note that $\mathcal{C}$ corresponds to a subset of $\mathcal{PC}_{\mathcal{PB}}$.

We distinguish and include the empty sets corresponding to different singleton policy configurations so that keys can be assigned to these classes, which enable users belonging to these classes to derive required decryption keys of lower classes. This key derivation process will be described in Section III.

The idea for the secure broadcasting mode of the data source is this: the portions of the source marked by different classes of nodes are encrypted by different secret keys, and are broadcast periodically to the subscribers. Subscribers receive only the keys for the document sources that they can access according to the policies.

The following definition introduces a partial order relation defined over $\mathcal{C}$.

Definition 2 (Partial order relation on $\mathcal{C}$): Let $\mathcal{C}_i$ and $\mathcal{C}_j$ be two classes of nodes marked by Pc_i and Pc_j, respectively, where Pc_i and Pc_j are policy configurations in $\mathcal{PC}_{\mathcal{PB}}$. We say that $\mathcal{C}_i$ dominates $\mathcal{C}_j$, written $\mathcal{C}_j \preceq \mathcal{C}_i$, if and only if $Pc_i \subseteq Pc_j$. We also write $\mathcal{C}_j \prec \mathcal{C}_i$ if $\mathcal{C}_j \preceq \mathcal{C}_i$ but $\mathcal{C}_j \neq \mathcal{C}_i$. We also say that $\mathcal{C}_i$ directly dominates $\mathcal{C}_j$, written $\mathcal{C}_j \prec_d \mathcal{C}_i$, if and only if $\mathcal{C}_i \neq \mathcal{C}_j$ and $\mathcal{C}_j \preceq \mathcal{C}_* \preceq \mathcal{C}_i$ implies $\mathcal{C}_* = \mathcal{C}_i$ or $\mathcal{C}_* = \mathcal{C}_j$. We call "$\mathcal{C}_j \prec_d \mathcal{C}_i$" a directed edge. We say $\mathcal{C}_i$ dominates $\mathcal{C}_j$ via n directed edges if there exists $\{\mathcal{C}_{i_k}\}_{1\leq k\leq n-1} \subseteq \mathcal{C}$ such that $\mathcal{C}_j \prec_d \mathcal{C}_{i_1}$, $\mathcal{C}_{i_{n-1}} \prec_d \mathcal{C}_j$ and $\mathcal{C}_{i_{k-1}} \prec_d \mathcal{C}_{i_k}$ for $2 \leq k \leq n-1$.

III. KEY MANAGEMENT SCHEME

A. Initialization

Suppose we have already generated the set $\mathcal{C}$ of classes of nodes of the data source $\mathcal{S}$ marked with the policy configurations Pc_i in $\mathcal{PB}$. Such a set is partially ordered with respect to $\preceq$. Let n be the cardinality of $\mathcal{C}$.

In this step, the system parameters are initialized and the system's class keys K_i are generated.

1) The vendor chooses an elliptic curve E over a finite field $\mathbb{F}_q$ so that the discrete logarithm problem is hard on $E(\mathbb{F}_q)$.[1] The vendor also chooses a point $Q \in E(\mathbb{F}_q)$ with a large prime order, say, p. The vendor then chooses $2n$ integers n_i, g_i such that $n_i g_i$ are all different modulo p for $1 \leq i \leq n$. The vendor computes $P_i = n_i Q$ on $E(\mathbb{F}_q)$ and h_i such that $g_i h_i \equiv 1 \pmod p$. The class key $K_i = g_i P_i$ is computed for class $\mathcal{C}_i$. The points $R_{i,j} = g_i K_j + (-K_i)$ are also computed whenever $\mathcal{C}_j \prec \mathcal{C}_i$ (not just when $\mathcal{C}_j \prec_d \mathcal{C}_i$).
2) The vendor chooses two random integers a, b and a keyed-hash message authentication code (HMAC) [6] $H_K(-)$ built with a hash function $H(-)$ and a fixed secret key K. K serves as the system's master key and is only known to the vendor.
3) The vendor publishes $R_{i,j}$ on an authenticated board, whereas the integers g_i, h_i, a and b are kept secret. Parties can verify the validity of the $R_{i,j}$ obtained from the board. This can be realized by using digital signatures.

The public values $R_{i,j}$ are constructed in such a way that the owner of the key K_j of the lower class $\mathcal{C}_j$ cannot obtain any information about the class key K_i of the higher class $\mathcal{C}_i$ without knowing the secret value g_i, and the owner of the higher class key K_i cannot compute K_j on its own, due to the the difficulty of solving the discrete logarithm problem. It turns out that such construction is secure against the attack [12] which breaks Chien's earlier scheme [5]. We will discuss this in section IV-C.3.

B. Encrypting Key Generation

In this step we generate the temporal encryption class keys $K_{i,t}$ at time granule t by using the system's class keys K_i.

The class of nodes $\mathcal{C}_i \in \mathcal{C}$ is encrypted by a symmetric encryption algorithm, e.g., AES [1]. We denote by $K_{i,t}$ the secret key for $\mathcal{C}_i$ at time granule $t \in [T_b, T_e] = [1, Z]$. The generation process for $K_{i,t}$ is given by the formula below:

$$K_{i,t} = H_K\left((K_i)_Y \oplus H^t(a) \oplus H^{Z-t}(b) \oplus ID_i\right),$$

where $(K_i)_Y$ is the y-coordinate of K_i, $H^m(x)$ is the m-fold iteration of $H(-)$ applied to x, ID_i is the identity of $\mathcal{C}_i$ and $\oplus$ is the bitwise XOR. Note that we can choose $H(-)$ properly in the initialization process so that the output of H_K is the right length for a key for the symmetric encryption algorithm we use.

The one-way property of the hash function H ensures that $H^t(a)$ and $H^{Z-t}(b)$ can be calculated only when the values $H^{t_1}(a)$ and $H^{Z-t_2}(b)$ are available for some t_1, t_2 with $t_1 \leq t \leq t_2$. This is the idea for the construction of the "time-bound" of the key management scheme.

C. User Subscription

This is the user subscription phase, in which a tamper-resistant device storing important information is issued to the subscriber.

Upon receiving a subscription request, an appropriate access control policy acp_i is searched until there is a match, then the policy configuration in $\mathcal{PB}$ which contains **only** acp_i is found, and thus the corresponding class of nodes marked with it, say $\mathcal{C}_i$, is identified. Note that $\mathcal{C}_i$, which could be an empty set, is always in $\mathcal{C}$ by the construction in Definition 1. We define the **encryption information**, $EncInf_i$, as follows:

$$EncInf_i = \{\left(H^{t_1}(a), H^{Z-t_2}(b)\right)\},$$

where the set on the right side is defined for all acceptable time intervals $[t_1, t_2]$ for acp_i.

The vendor distributes the class key K_i to the subscriber through a secure channel. The vendor also issues the subscriber a tamper-resistant device storing H_K (thus H, K), E, $\mathbb{F}_q$, ID_i, h_i and $EncInf_i$. There is also a secure clock embedded in the device which keeps track of current time. The device is tamper-resistant in the sense that no one can recover K, h_i, $EncInf_i$, change the values of ID_i, or change the time of the clock.

[1] For more background on elliptic curve cryptography, see [14].

D. Decrypting Key Derivation

In this step the temporal keys for a class and the classes below it are reconstructed by the tamper-resistant device.

Assume that the subscription process mentioned above is completed for a subscriber U associated with class $\mathcal{C}_i$. U can then use the information received from the vendor to decrypt the data in class $\mathcal{C}_j$, with $\mathcal{C}_j \preceq \mathcal{C}_i$, as follows:

1) If $\mathcal{C}_j = \mathcal{C}_i$, U inputs only K_i into the tamper-resistant device; otherwise if $\mathcal{C}_j \prec \mathcal{C}_i$, U first retrieves $R_{i,j}$ from the authenticated public board, then inputs it together with the class identity ID_j of $\mathcal{C}_j$ and its secret class key K_i.
2) If K_j is the only input, the next step is executed directly. Otherwise, the tamper-resistant device computes the secret class key of $\mathcal{C}_j$:
$$K_j = h_i \cdot (R_{i,j} + K_i).$$
3) If $t \in [t_1, t_2]$ for some acceptable time interval $[t_1, t_2]$ of acp_i, the tamper-resistant device computes
$$H^t(a) = H^{t-t_1}(H^{t_1}(a)), H^{Z-t}(b) = H^{t_2-t}(H^{Z-t_2}(b)),$$
and $K_{j,t} = H_K((K_j)_Y \oplus H^t(a) \oplus H^{Z-t}(b) \oplus ID_j)$. Note that the values $H^{t_1}(a)$ and $H^{Z-t_2}(b)$ are pre-computed and stored in the tamper-resistant device.
4) At time granule t, the protected data belonging to class $\mathcal{C}_j$ can be decrypted by applying the key $K_{j,t}$.

E. An Example

We now provide an example to illustrate the above process.

Consider an electronic newspaper system. Let **one day** be a tick of time in this system and $Z = 70$ be the life time of the system, i.e., the system exists in the temporal interval $[1, 70]$. Let U be a user wishing to subscribe the sports portion of the newspaper for one week, say, the period $I = [8, 14]$. We could match U with an access control policy acp_1 = ([8,14], All days, Subscriber/type="full", Sports_supplement, view, CASCADE). Then we can find the class of nodes $\mathcal{C}_1$ marked with policy configuration acp_1 from a pre-generated table. These nodes are encrypted and broadcast periodically. U can derive the decryption key for the subscription period using the issued class key K_1 and the tamper-resistant device storing H_K, E, $\mathbb{F}_q$, ID_1, h_1 and $H^8(a)$, $H^{56}(b) = H^{70-14}(b)$. For example, U inputs K_1 into the device. To obtain the decryption key $K_{1,10}$ at time granule $t = 10$, the device computes

$$H^{10}(a) = H^2(H^8(a)), H^{60}(b) = H^4(H^{56}(b))$$

then $K_{1,10} = H_K((K_1)_Y \oplus H^{10}(a) \oplus H^{60} \oplus ID_1)$, the very thing needed. To obtain the decryption key at $t = 13$ for a class $\mathcal{C}_2 \preceq \mathcal{C}_1$, U inputs K_1, ID_2 and $R_{1,2}$ into the device. The device first computes the class key of $\mathcal{C}_2$

$$K_2 = h_1 \cdot (R_{1,2} + K_1).$$

Then it computes

$$H^{13}(a) = H^5(H^8(a)), H^{57}(b) = H(H^{56}(b))$$

and $K_{2,13} = H_K((K_2)_Y \oplus H^{13}(a) \oplus H^{57}(b) \oplus ID_2)$, the decryption key needed.

Note that all computations are executed by the tamper-resistant device. The device can prevent the results of the computations from being revealed, so that even the user U does not know the class key K_2 of the class of nodes $\mathcal{C}_2 \prec \mathcal{C}_1$. This makes the system secure.

IV. FURTHER DISCUSSION

We have proposed a key assignment scheme for secure broadcasting based on a tamper-resistant device. A secure hash function and the intractability of the discrete logarithm problem on elliptic curves over the finite field $\mathbb{F}_q$ are also assumed.

A. Tamper-resistant Devices

The tamper-resistant device plays an important role in our scheme. The system's master key, K, must be protected by the device. Leak of $EncInf_i$ will not help the attackers much, because they are not able to compute the HMAC, thus the temporal class keys, without knowing K. A leak of h_i will enable the user of class $\mathcal{C}_i$ to obtain the class key K_j of $\mathcal{C}_j$, where $\mathcal{C}_j \preceq \mathcal{C}_i$, by computing

$$K_j = h_i \cdot (R_{i,j} + K_i),$$

as is done by the device. However, this does not help the user decrypt any information belonging to a class no lower than $\mathcal{C}_i$. Unless K is discovered, the attacks to retrieve $EncInf_i$ and h_i on individual devices are not effective. With the use of a tamper-resistant device, the security of the scheme is strong enough. Attacks on tamper-resistant devices need special equipment. It is cheaper to buy a subscription than the special equipment. As such the attacker does not have economic incentives to mount such an attack unless he could capture the master key K. An attacker who could find all the information on several tamper-resistant devices could execute a collusion attack to compute extra temporal decryption keys.

As pointed out above, the only information which needs to be kept secret by the tamper resistant device is the system's master key K. The **Trusted Platform Module** (TPM) technology [10], which is good for storing and using secret keys, can well suit our need. We are aware that there are attacks on TPMs [9]. There are countermeasures against those attacks [9]. Moreover, none of these attacks is capable of extracting the exact secret information being protected (in our case, the system key K). Hence the attackers are not able to perform the HMAC operations. Therefore an attack relying on the knowledge of K is not feasible in practice. We believe the use of the tamper-resistant hardware is practical and secure in reality.

One might argue that if we need such a strong tamper-resistant device, then we might as well store the needed temporal decryption keys on it directly and discard the key management scheme. However, that approach is not practical, because the number of needed keys can be large, considering the temporal intervals and hierarchy. And in that case, the system's class keys can not be easily updated. Our proposed scheme is elegant and more efficient in terms of storage on the tamper-resistant devices.

B. Hash Functions and ECDLP

Some of the most widely used hash functions, e.g. SHA-0, MD4, Haval-128, RipeMD-128, MD5, were broken years ago; SHA-1 was announced broken early in the year 2005. Essentially, these hash functions have been proven not to be collision-free; but it is still hard to find a pre-image to a given digest in a reasonable time. In view of this, these attacks on hash functions

will not affect the security of our scheme, as long as the discrete logarithm problem on the elliptic curves is still hard. So far there is no foreseeable breakthrough in solving DLP on elliptic curves.

Without having to keep $Q \in E(\mathbb{F}_q)$ secret, no one, including the user U_i, can recover the secret values g_i, h_i of the system due to the difficulty of the elliptic curve discrete logarithm problem. Therefore, the system is secure.

C. Security Against Possible Attacks

Note that the tamper-resistant device in our scheme is an oracle that does calculation in the Decrypting Key Derivation process. This raises the question of whether such a device can be attacked by an adversary to gain secret information to subvert this process. This concern is necessary since Chien's scheme has been successfully attacked (see X. Yi [12]) due to the weakness of the oracle. We face a similar situation here.

1) Attack from outside: First, any attack against our scheme with only one input to the device will not work. Any attempt to gain the temporal decrypting key with only one input K_* to the device with identity ID_i will not succeed, unless the input is the right class key K_i bound to the same device. This can easily be seen since in this case the device will compute $H_K\left((K_*)_Y \oplus H^t(a) \oplus H^{Z-t}(b) \oplus ID_i\right)$ at time granule t (we may assume t is valid, i.e. it is in the subscription period). This value is meaningless unless $K_* = K_i$.

2) Collusion attack: Second, any collusion attack with more than one input to the device does not work either. Since the encryption information $EncInf_i$ for a device with identity ID_i is not likely to be modified because of the tamper-resistance of the device, any attempt to derive temporal decrypting keys for a class $\mathcal{C}_m$ which is no lower than $\mathcal{C}_i$ inevitably involves the computation of the class key K_m. According to Step 2 of the Decrypting Key Derivation process, $g_i K_m$ must be computable by the device with a suitable choice of the input parameters. However, we do not see any way to accomplish this computation without solving the discrete logarithm problem on $E(\mathbb{F}_q)$.

3) X. Yi's attack: As a particular case of the collusion attack just described, X. Yi's attack [12] against Chien's scheme [5] cannot be replayed here to break our scheme. We will demonstrate this case to give an impression of how the asymmetry introduced by elliptic curve cryptography helps to strengthen the scheme.

X. Yi's attack can not apply directly to our scheme due to our different construction. An analogue of it would work like this: two users collude to derive certain information *Inf* and pass it to a third user, U, so that U can input *Inf* together with her/his secret key to the tamper-resistant device to derive the decryption keys of a class no lower than U's. Suppose U belongs to class $\mathcal{C}_j$ and U wants to derive decryption keys $K_{i,t}$ of $\mathcal{C}_i$, which is no lower than $\mathcal{C}_j$. Then K_i needs to be computed by the device. Thus the information to be passed to U should be $Inf = g_j K_i + (-K_j)$, so that when U inputs Inf, ID_i and K_j, the tamper-resistant device will compute

$$h_j \cdot (Inf + K_j) = h_j \cdot (g_j K_i + K_j - K_j) = K_i.$$

In order to obtain Inf, someone must be able to compute $g_j K_i$. Given that class $\mathcal{C}_i$ is no lower than $\mathcal{C}_j$, $g_j K_i$ is not a summand of any of the published values on the authenticated board, and thus it cannot be produced via collusion, considering the fact that the elliptic curve discrete logarithm problem is hard.

Therefore, X. Yi's attack cannot be modified to attack our scheme.

D. Yet Another Good Feature

An important advantage of our scheme is that the vendor can change the class keys of the system at anytime without having to re-issue new devices to the users, while only the user's class keys and the public information $R_{i,j}$ need to be updated. However, when an individual user wants to change the subscription, a new device needs to be issued. This also needs to be done when a different class is desired.

E. Space and Time Complexity

Our scheme publishes one value $R_{i,j}$ for each partial order relation $\mathcal{C}_j \prec \mathcal{C}_i$. The total number of public values is at most $\frac{n(n-1)}{2}$, when n is the number of classes in $\mathcal{C}$. On the user's side, the tamper-resistant device stores only H_K, E, $\mathbb{F}_q$, ID_i, h_i and $EncInf_i$.

At any time granule t, the tamper-resistant device needs to perform $(t-t_1)+(t_2-t)+2 = t_2-t_1+2 \leq Z$ hash iterations. Note that there are two hash iterations per HMAC operation [6]. In a system of life period 5 years which updates user keys every hour, Z is approximately 43800. We did an experiment using SHA-1 as the hash function on a Gateway MX3215 laptop computer which has a 1.40GHz Intel(R) Celeron(R) M processor, 256 MB of memory and runs Ubuntu 6.10 Edgy Eft. The code is written in C and built with GNU C compiler version 4.1.2. The result showed that 43800 hash iterations took .0800 second of processing time. In practice, $t_2 - t_1$ is usually much smaller than Z and the hash computation is really fast.

The bulk of the computation performed by the tamper-resistant device is the calculation of $K_j = h_i(R_{i,j} + K_i)$ in Step 2 of the Decrypting Key Derivation phase. A rough estimate [7] shows that a 160-bit prime p (the order of Q on $E(\mathbb{F}_q)$) should give us enough security (against the best ECDLP attack) in this situation. In this case, to derive the class key K_j of class $\mathcal{C}_i \prec \mathcal{C}_i$ from K_i, the device needs to perform at most 160 elliptic curve doublings and 81 elliptic curve additions, when the method based on repeated doubling and adding is used. This amounts to 241 elliptic additions. Ignoring the negligible field addition in $\mathbb{F}_q$, each elliptic curve addition requires 1 field inversion and 2 field multiplications. If we choose q to be a 160-bit number and regard the time to perform a field inversion as that of 3 field multiplications, the class key derivation process needs roughly $241 \times 5 \times 160^2 \approx 2^{25}$ bit operations. Even a smart card can do this in a few seconds [8]. Our scheme is in fact slower than Chien's scheme, in which only hash computations are widely used. However it is still very efficient from the point of view of application and provides enhanced security.

We include in Appendix I a table comparing the three time-bound hierarchical key management schemes.

V. CONCLUSIONS

In this paper, we have proposed an efficient time-bound hierarchical key management scheme, based on the use of elliptic curve cryptography, for secure broadcasting of data. The number of encryption keys to be managed depends only on the number of access control policies. A tamper-resistant device plays an important role in our scheme.

The obvious solution of storing all needed decryption keys in a tamper-resistant device is not practical because the number of keys needed can be large. Also, with such a solution, when the system's class keys need to be updated, all devices containing these keys must be discarded and new devices need to be issued. Our approach to key management avoids these disadvantages.

In the future, we hope to analyze our system from the point of view of provable security. This would require a more formal description of our system than we have given here. We also plan to implement our scheme and do experiments on smart cards.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful comments.

The authors would also like to thank Abhilasha Bhargav-Spantzel for the suggestions about TPM technology.

The authors thank the Center for Education and Research in Information Assurance and Security at Purdue University for support.

The work reported in this paper has been partially supported by the National Science Foundation under Grant No. 0430274.

REFERENCES

[1] *Advanced Encryption Standard*. http://csrc.nist.gov/CryptoToolkit/aes/.
[2] R. Anderson and M. Kuhn. "Low Cost Attacks on Tamper Resistant Devices", *5th International Workshop on Security Protocols (LNCS 1361)*, Springer, pp. 125-136, 1997.
[3] E. Bertino, C. Bettini, E. Ferrari and P. Samarati. "An Access Control Model Supporting Periodicity Constraints and Temporal Reasoning", *ACM Transactions on Database Systems*, Vol. 23, No. 3, pp. 231-285, Sept. 1998.
[4] E. Bertino, B. Carminati and E. Ferrari. "A Temporal Key Management Scheme for Secure Broadcasting of XML Documents", *CCS'02*, pp. 31-40, Nov. 2002.
[5] H.-Y. Chien. "Efficient Time-Bound Hierarchical Key Assignment Scheme", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 16, No. 10, pp. 1302-1304, Oct. 2004.
[6] FIPS Pub 198, *The Keyed-Hash Message Authentication Code (HMAC)*. http://csrc.nist.gov/publications/fips/fips198/fips-198a.pdf.
[7] A. Jurisic and A. J. Menezes. "Elliptic Curves and Cryptography", *Dr. Dobb's Journal*, April 1997, 23-36.
[8] *Web article*. http://www.raaktechnologies.com/download/raak-c7-standard.pdf.
[9] E. R. Sparks, *A Security Assessment of Trusted Platform Modules*. Computer science technical report. http://www.ists.dartmouth.edu/library/341.pdf.
[10] *Trusted Platform Module.* https://www.trustedcomputinggroup.org/groups/tpm/.
[11] 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. *6th ACM Symposium on Access Control Models and Technologies (SACMAT 2001)*, May 3-4, 2001, Litton-TASC, Chantilly, Virginia, USA. ACM, 2001.
[12] 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, Sep. 2005.
[13] 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, Jul./Aug. 2003.
[14] L. C. Washington. *Elliptic Curves, Number Theory and Cryptography*. Chapman & Hall/CRC, 2003.

APPENDIX I
COMPARISON OF THREE SCHEMES

We compare the three time-bound hierarchical key management schemes in the following Table I.

Elisa Bertino is professor of Computer Science at Purdue University and serves as Research Director of the Center for Education and Research in Information Assurance and Security (CERIAS). Previously she was a faculty member at Department of Computer Science and Communication of the University of Milan where she directed the DB&SEC laboratory. She has been a visiting researcher at the IBM Research Laboratory (now Almaden) in San Jose, at the Microelectronics and Computer Technology Corporation, at Rutgers University, at Telcordia Technologies. Her main research interests include security, privacy, digital identity management systems, database systems, distributed systems, multimedia systems. In those areas, Prof. Bertino has published more than 250 papers in all major refereed journals, and in proceedings of international conferences and symposia. She is a co-author of the books "Object-Oriented Database Systems - Concepts and Architectures" 1993 (Addison-Wesley International Publ.), "Indexing Techniques for Advanced Database Systems" 1997 (Kluwer Academic Publishers), "Intelligent Database Systems" 2001 (Addison-Wesley International Publ.), and "Security for Web Services and Service Oriented Architectures" Springer (to appear in Fall 2007). She has been a co-editor in chief of the Very Large Database Systems (VLDB) Journal from 2001 to 2007. She serves also on the editorial boards of several scientific journals, incuding IEEE Internet Computing, IEEE Security&Privacy, ACM Transactions on Information and System Security, ACM Transactions on Web. Elisa Bertino is a Fellow member of IEEE and a Fellow member of ACM and has been been named a Golden Core Member for her service to the IEEE Computer Society. She received the 2002 IEEE Computer Society Technical Achievement Award for "For outstanding contributions to database systems and database security and advanced data management systems" and the 2005 IEEE Computer Society Tsutomu Kanai Award "For pioneering and innovative research contributions to secure distributed systems".

Ning Shang is a graduate student in the Department of Mathematics, the Department of Electrical and Computer Engineering and the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University. He is a member of the AMS and SIAM. His research focuses on computational number theory, elliptic and hyperelliptic cryptography, and implementation of cryptographic schemes.

Samuel S. Wagstaff, Jr. received a B.S. from the Massachusetts Institute of Technology, Cambridge, Massachusetts, USA and a Ph.D. from Cornell University, Ithaca, New York, USA, both in Mathematics. He is a professor of Computer Science at Purdue University, West Lafayette, Indiana, USA. Before coming to Purdue, he taught at the Universities of Rochester (New York), Illinois (Urbana) and Georgia (Athens). He worked at the Institute for Advanced Study, Princeton, New Jersey, in 1971–1972. He is a member of the AMS, MAA and UPE. His research interests include primality testing, integer factorization, cryptography, secure patch distribution and watermarking. He is the leader of the Cunningham Project, which factors numbers of the form $b^n \pm 1$. He has supervised five Ph.D. theses and published five books and more than 60 research papers. An algorithm that he and R. Baillie invented and published in 1980 was selected as ANSI Standard X9-80 for choosing industrial-grade primes for use in cryptography. It is used worldwide as part of the Secure Socket Layer.

TABLE I

A COMPARISION OF THREE SCHEMES

Comparison of three schemes			
	Tzeng	Chien	Ours
implementation requirements	Lucas function	tamper-resistant device	tamper-resistant device, ECC
# of public values	$n+6$	$n-1$	$n(n-1)/2$
# of operations to derive temporal secret key of own class	$(t_2 - t_1)T_e, (t_2 - t_1)T_L, T_h$	$(t_2 - t_1 + 1)T_h$	$(t_2 - t_1 + 2)T_h$
# of operations to derive temporal secret key of direct child class	$(t_2 - t_1 + r)T_e, (t_2 - t_1)T_L, T_h$	$(t_2 - t_1 + 2)T_h$	$(t_2 - t_1 + 2)T_h, T_E$
# of operations to derive temporal secret key of l-edge-distance child class	$(t_2 - t_1 + r)T_e, (t_2 - t_1)T_L, T_h$	$(t_2 - t_1 + 1 + l)T_h$	$(t_2 - t_1 + 2)T_h, T_E$
security against Yi and Ye's attack	insecure	secure	secure
security against X. Yi's attack	N/A	insecure	secure

Suppose $\mathcal{C}_j \preceq \mathcal{C}_i$, $t \in [t_1, t_2]$.
Notation:

n: number of classes $|\mathcal{C}|$
r: number of child classes $\mathcal{C}_i$ on path from $\mathcal{C}_i$ to $\mathcal{C}_j$
T_h: hashing operation
T_e: modular exponentiation
T_L: Lucas function operation
T_E: elliptic curve scalar multiplication