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

Elisa Bertino* Ning Shang† Samuel S. Wagstaff, Jr.‡

Abstract

A time-bound key management scheme for secure broadcasting of XML documents was proposed by E. Bertino, et al., in 2002, in which a hierarchical structure is defined for the XML source according to access control policies of the document source. A scheme due to Tzeng was suggested in that paper. Tzeng's scheme is efficient in terms of space requirement. However, heavy public-key computations are inevitably involved, which can make the implementation costly. What's more, X. Yi and Y. Ye showed it insecure against collusion attacks. In October 2004, H.Y. Chien suggested an efficient time-bound key assignment scheme built on tamper-resistant devices and hash functions. Chien's scheme seemed to suit our need well until a security flaw was found by X. Yi in a paper published in September 2005. In this paper, we shall propose a new key assignment scheme for access control which is both efficient and secure.

Keywords: XML, secure broadcasting, time-bound hierarchical key management, elliptic curves, elliptic curve discrete logarithm problem(ECDLP).

1 INTRODUCTION

In a web-based environment, the data to be securely broadcast, e.g., electronic newspapers or other typcs of content, can be encoded according to XML [9] and broadcast periodically to the community of subscribers. By secure broadcasting we mean the delivery of the information should follow the access control policies of the data source. There is a natural tree structure for XML, and this can be handled conveniently by an XML-related standard called XPath. We may define the *class of nodes* of an XML source (S) according to the *policy configurations* and organize the set of such classes into a partially ordered set by defining the

*CERIAS and Department of Computer Sciences, Purdue University. Email: bertino@cerias.purdue.edu

†CERIAS and Department of Mathematics, Purdue University. Email: nshang@math.purdue.edu

‡CERIAS and Department of Computer Sciences, Purdue University. Email: ssw@cerias.purdue.

relation $\preceq$. This was elaborated in [4] and a time-bound scheme suggested by W.G. Tzeng [8] was used for key assignment. Tzeng's scheme is efficient in terms of space requirement, but it is insecure against collusion attacks as was shown by Yi and Ye [12].

An efficient 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 computation load and implementation cost. However, it has a security hole against X. Yi 's three-party collusion attack [11]. Inspired by Chien's idea, we propose in this paper a new method for access control with the use of elliptic curve cryptography. This scheme is secure against X. Yi 's three-party collusion attack and efficient. 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 the subscription they would be used to steal. Hence there is a good reason to believe that our scheme that uses tamper-resistant devices can have practical important applications, in areas such as digital right management.

The rest of this paper is organized as follows: Section 2 presents the notation and definitions needed to give a hierarchical structure to the XML source. Section 3 proposes a new key management scheme applied to the hierarchy. Section 4 contains further discussions of the key management scheme. Section 5 concludes our results.

2 DEFINITIONS AND NOTATION

We will follow the notation of [4] in this article and recall some important definitions below.

2.1 Time-related definitions

In our scheme, we assume time to be quantized, i.e., there is a smallest unit of time, namely a *tick*, e.g., a day. We call a *temporal interval* a continuous period of time which can be decomposed into a finite number of ticks.

Definition 1 (Calendar) *A* **calendar** *is a countable set of contiguous temporal intervals, numbered by integers, called* **indices** *of the intervals. Given two calendars C_1 and C_2, we say that C_2 is a* **subcalendar** *of C_1 if each interval of C_1 can be exactly covered by a finite number of intervals of C_2, and we write $C_2 \sqsubseteq C_1$ in this case. A* **basic calendar** *is a calendar such that each interval is a tick. Note that a basic calendar is always a subcalendar of any calendar.*

Throughout this article, we assume the existence of a totally ordered finite set $\mathcal{F}$ of calendars with respect to $\sqsubseteq$, with the basic calendar the minimum element in $\mathcal{F}$. The term *time granules* is used to indicate the indices of the basic calendar and t_1, t_2, ..., t_n are notation for a set of time granules.

Definition 2 (Periodic Expression) *Given calendars C_d, C_1, ..., C_n in $\mathcal{F}$, a* **periodic expression** *is defined as $P = \sum_{i=1}^{n} O_i.C_i \rhd r.C_d$, where O_1 is set to be "all", $O_i \in 2^{\mathbb{N}}$ and $C_i \sqsubseteq C_{i-1}$ for $i = 2, \ldots, n$, $C_d \sqsubseteq C_n$, and $r \in \mathbb{N}$. A periodic expression is separated by the symbol $\rhd$. The first part gives the set of starting points of the temporal intervals, and the second part specifies the duration of each temporal interval in terms of calendar C_d. We have the convention that O_i is omitted when its value is "all", and that $r.C_d$ is omitted when it is equal to $1.C_d$.*

For details on periodic expressions, we refer the reader to [3].

Definition 3 (Function $\prod(-)$) *Given a periodic expression $P = \sum_{i=1}^{n} O_i.C_i \rhd r.C_d$, $\prod(P)$ is the infinite set of temporal intervals corresponding to P, whose common duration is $r.C_d$. The set St of starting points of temporal intervals in $\prod(P)$ is inductively defined as follows:*

1. *If $n = 1$, then $St = \{$all starting points of the intervals of calendar $C_1\}$;*
2. *If $n > 1$, then $St = \{$starting point of the k^{th} interval of calendar C_n included in each interval of $\prod(\sum_{i=1}^{n-1} O_i.C_i \rhd 1.C_{n-1}) : k \in O_n\}$.*

We assume the existence of a life time $[T_b, T_e]$ of the system, where T_b and T_e are time granules. Note that by a translation along the time line, we can assume $T_b = 1$ and $T_e = Z$, with Z being the number of time granules of the life time of the system. Throughout this paper, we shall denote by $[T_b, T_e] = [1, Z]$ the life time of the system, starting at time granule 1.

Given a temporal interval $[T_1, T_2]$, we only consider the intersection of it and $[T_b, T_e]$, i.e., $I = [T_1, T_2] \cap [T_b, T_e]$ and define the restriction of $\prod(-)$ to I.

Definition 4 (Restriction of $\prod(-)$) *Let P be a periodic expression and I be an interval contained in $[T_b, T_e]$. The* **restriction of $\prod(P)$ to** *I is defined as*

$$\prod(P)|_I = \{[t_1, t_2] \cap I : [t_1, t_2] \in \prod(P)\}.$$

Note that $\prod(P)|_I$ is always a finite set.

2.2 Access Control Policies for XML Documents

Access control policies for XML documents are expressed in terms of four basic components.

Subject credentials A *credential* is a set of attributes representing subject properties that are relevant for access control purpose. Credentials with the same structure are grouped into *credential types*. Both credentials and credential types are encoded using χ-Sec, an XML-based language [10].

An access control policy applies to a subject if and only if its credential satisfies the constraints in the policy specification. The set of credentials to which an access control policy applies are denoted through XPath-compliant expressions.

Protection objects A *protection object* is the document portion to which a policy applies. The set of protection objects to which an access control policy applies are denoted through XPath-compliant expressions.

Privileges We only concentrate on browsing privileges, i.e., privileges for reading the information in a protection object. We assume the supported browsing privileges are: view, navigate and browse_all.

Propagation options A *propagation option* states whether and how a policy specified on a given protection object *obj* propagates to protection objects that are related to *obj* by some semantic relationship. Two types of propagation, *implicit* and *explicit propagation*, are supported. There are three explicit propagation options supported: NO_PROP, FIRST_LEVEL and CASCADE.

We refer the reader to [4] for more details about these components.

Now we are ready to give the definition of an access control policy.

Definition 5 (Access Control Policy) *An* **access control policy** *of an XML source is a tuple (I, P, sbj-spec, prot-obj-spec, priv, prop-opt), where: I is a temporal interval; P is a periodic expression; sbj-spec is an XPath-compliant expression on credentials or credential types; prot-obj-spec is an XPath-compliant expression denoting the protection objects to which the policy applies; priv*$\in$ *{view, navigate, browse_all} is a browsing privilege; and prop-opt*$\in$*{NO_PROP, FIRST_LEVEL, CASCADE} is a propagation option. Given an access control policy acp=(I, P, sbj-spec, prot-obj-spec, priv, prop-opt), we denote with I(acp), P(acp), sbj-spec(acp), prot-obj-spec(acp), priv(acp) and prop-opt(acp) the temporal interval, the periodic expression, the credential specification, the protection object specification, the privilege and the propagation option of acp, respectively.*

The policy base $\mathcal{PB}$ is the set of access control policies defined for an XML source $\mathcal{S}$.

It is important to notice that several policies may apply to each node in a document. In what follows we refer to the set of policies applying to a node in a document as **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 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.

Definition 6 (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 as $\mathcal{C}_i$, is the set of nodes belonging to documents in the XML 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 $\mathcal{C}$ is a subset of $\mathcal{PC}_{\mathcal{PB}}$.*

The placement of the empty sets in the construction of $\mathcal{C}$ may seem artificial here at this moment, but the reason we do this will become clear when we explain the user subscription phase in the key management scheme below.

The idea for the secure broadcasting mode of the documents in the XML source is this: the portions of the XML document 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 documentodes that they can access according to the policies.

If there are N_p access control policies in $\mathcal{PB}$, the set $\mathcal{PC}_{\mathcal{PB}}$ consists of 2^{N_p} policy configurations, and $\mathcal{C}$ consists of 2^{N_p} elements in the worst case. Therefore in the worst case the system should manage 2^{N_p} different secret keys. This could be a large number. So we need an efficient key management scheme for our system to work reasonably well with respect to performance.

3 KEY MANAGEMENT SCHEME

The first step is to make $\mathcal{C}$ a partially ordered set by introducing the partial order relation "$\preceq$" on $\mathcal{C}$.

Definition 7 (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 $\mathcal{C}_j \neq \mathcal{C}_i$.*

3.1 Initialization

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

1. The vendor chooses an elliptic curve E over a finite field $\mathbb{F}_q$ so that the discrete logarithm problem is hard on $E(\mathbb{F}_q)$. The vendor also chooses a point $Q \in E(\mathbb{F}_q)$ with a large prime order, say, p. $2n$ integers n_i, g_i are then randomly chosen by the vendor 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 $\mathcal{C}_j \prec \mathcal{C}_i$.

2. The vendor chooses a one-way hash function $H(-)$ and two random integers a and b to use.

3. The vendor publishes $R_{i,j}$ on an authenticated board, whereas the integers g_i, h_i, a and b are kept secret. People can verify the validity of the $R_{i,j}$ obtained from the board. This can be realized by using digitial signatures.

3.2 Encrypting Key Generation

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:

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

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 is suitable for the symmetric encryption algorithm we use.

3.3 User Subscription

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 6. We define the **encryption information**, $EncInf_i$, as follows:

$$EncInf_i = \{(H^{t_1}(a), H^{Z-t_2}(b)) : [t_1, t_2] \in \prod (P(acp_i))|_{I(acp_i)}\}$$

where $|_{I(acp_i)}$ is the restriction as is defined in Definition 4.

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, 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 $EncInf_i$, H, h_i, change the values of ID_i, or change the time of the clock.

3.4 Decrypting Key Derivation

Now assume the subscription process mentioned above is completed for a subscriber U associated to 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 her/his 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 $\mathcal{C}_j$:

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

3. If $t \in [t_1, t_2]$ for some $[t_1, t_2] \in \prod(P(acp_i))|_I(acp_i)$, 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 $K_{j,t} = H((K_j)_Y \oplus H^t(a) \oplus H^{Z-t}(b) \oplus ID_j)$. 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 $K_{j,t}$.

3.5 An Example

We give a real-world 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 in the subscription period using the issued class key K_1 and the tamper-resistant device storing H, 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

$$H^{10}(a) = H^2(H^8(a)), H^{70}(b) = H^4(H^{56}(b))$$

then $K_{1,10} = H((K_1)_Y \oplus H^{10}(a) \oplus H^{60} \oplus ID_1)$, the very thing we need. 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 $\mathcal{C}_2$

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

Then it computes

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

and $K_{2,13} = H((K_2)_Y \oplus H^{13}(a) \oplus H^{57}(b) \oplus ID_2)$, the encryption 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 $\mathcal{C}_2 \preceq \mathcal{C}_1$. This makes the system secure.

4 FURTHER DISCUSSIONS

We have proposed a key assignment scheme for secure broadcasting of XML documents which is based on a tamper-resistant device. A secure hash function and the hardness of solving the discrete logarithm problem on elliptic curves over finite fields are also assumed.

4.1 Tamper-resistant Devices

The tamper-resistant device plays an important role in our scheme. Obviously, without a tamper-resistant device, our scheme will not work. Leak of $EncInf_i$ makes the scheme vulnerable to a trivial two-user collusion attack. However, 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 carry on such an attack.

4.2 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; although it is still hard to find a pre-image to a given digest in a reasonable time. In this sense, these attacks to 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 security of the system is enforced.

4.3 Security Against Known 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 that whether such device can be attacked by an adversary to give unwanted information to subvert this process. This concern is necessary since Chien's scheme has been successfully attacked (see X. Yi [11]) due to the weakness of the oracle. We are facing a similar situation here.

First, any collusion 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 binded to the same device. This can be easily seen since in this case the device will compute $H(K_Y \oplus H^t(a) \oplus H^{Z-t}(b) \oplus ID_i)$ at time granule t (we may assume t is valid, i.e. it is in the subscription period). This value is meaningless unless $K = K_i$.

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 have not seen any way to accomplish this computation without solving the discrete logarithm problem on $E(\mathbb{F}_q)$. And it is clear that X. Yi's attack [11] against Chien's scheme [5] cannot be replayed here to break our scheme.

4.4 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 re-issuing new devices to the users, while only the user's class keys and the public information $R_{i,j}$ need to be updated.

4.5 Space Requirement And Efficiency

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 $\mathcal{C}$. This can be proven easily by induction. On the user's side, the tamper-resistant device stores only H, E, $\mathbb{F}_q$, ID_i, h_i and $EncInf_i$.

A 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 [6] 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 240 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 [7]. Our scheme is in fact slower than Chien's scheme, in which only hash computations are widely used. However it

is still very efficient towards a point of view of application and provides us more security.

5 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 XML documents. 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. We believe our scheme can be generalized for use in other applications, such as in Digital Rights Languages and in RFID systems.

References

[1] *Advanced Encryption Standard*. http://csrc.nist.gov/CryptoToolkit/aes/.

[2] Ross Anderson and Markus 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] Hung-Yu 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] Aleksandar Jurisic and Alfred J. Menezes. "Elliptic Curves and Cryptography", *Dr. Dobb's Journal*, April 1997, 23-36.

[7] *Web article*. http://www.semiconductors.philips.com/news/backgrounders/bg0026/.

[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, Jan./Feb. 2002.

[9] World Wide Web Consortium. Available at http://www.w3.org/XML/

[10] E. Bertino, S. Castano, E. Ferrari. "On specifying security policies for web documents with an XML-based language", *6th ACM Symposium on Access Control Models and Technologies (SACMAT 2001)*, May 3-4, 2001, Litton-TASC, Chantilly, Virginia, USA. ACM, 2001.

[11] 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.

[12] 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 Date Engineering*, Vol. 15, No. 4, pp.1054-1055, Jul./Aug. 2003.

[13] L. C. Washington. *Elliptic Curves, Number Theory and Cryptography.* Chapman & Hall/CRC, 2003.