The Journal of Systems and Software 84 (2011) 679–685

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Secure key management scheme for dynamic hierarchical access control based on ECC

Yu-Li Lin [a], Chien-Lung Hsu [b,c,*]

[a] Ministry of Justice, Investigation Bureau (MJIB), Taipei 231, Taiwan, ROC
[b] Department of Information Management, Chang Gung University, Tao-Yuan 333, Taiwan, ROC
[c] Taiwan Information Security Center at NTUST (TWISC@NTUST), Taiwan

ARTICLE INFO

Article history:
Received 23 September 2009
Received in revised form 29 October 2010
Accepted 30 November 2010
Available online 23 December 2010

Keywords:
Key management
Key assignment
Elliptic curve
Hierarchical access control

ABSTRACT

An access control mechanism in a user hierarchy is used to provide the management of sensitive information for authorized users. The users and their own information can be organized into a number of disjoint sets of security classes according to their responsibilities. Each security class in a user hierarchy is assigned an encryption key and can derive the encryption keys of all lower security classes according to predefined partially ordered relation. In 2006, Jeng and Wang proposed an efficient key management scheme based on elliptic curve cryptosystems. This paper, however, pointed out that Jeng–Wang scheme is vulnerable to the so-called compromising attack that the secret keys of some security classes can be compromised by any adversary if some public information modified. We further proposed a secure key management scheme based on elliptic curve cryptosystems to eliminate the pointed out the security leak and provide better security requirements. As compared with Jeng and Wang's scheme (Jeng and Wang, 2006), the proposed scheme has the following properties. (i) It is simple to execute the key generation and key derivation phases. (ii) It is easily to address dynamic access control when a security class is added into or deleted from the hierarchy. (iii) It is secure against some potential attacks. (iv) The required storage of the public/secret parameters is constant.

1. Introduction

The access control problem in a user hierarchy is used to many applications such as schools, governments, military, corporations, computer network systems, and database management systems. All users in such a system form a user hierarchy and can be assigned into a number of disjoint sets of security classes, say $SC=\{SC_1, SC_2, \ldots, SC_n\}$, which are partially ordered by a binary relation "$\leq$". In $(SC, \leq)$, $SC_j \leq SC_i$ means that the security level of class SC_i is higher than or equal to the security class SC_j. In other words, users in SC_j can access the encrypted information held by users in SC_j, but the opposite is disallowed. The secret key K_i is used by each security class SC_i to encrypt/decrypt its sensitive information. When a user in SC_i would like to retrieve data encrypted by SC_j, he should get the right key K_j.

Akl and Taylor (1983) first proposed a solution to solve the hierarchical access control problem. In their scheme, each security class is assigned a secret key and a public parameter. The security class SC_i can successfully use its secret key and some public parameters to derive the secret key of the security class SC_j such that $SC_j \leq SC_i$. Main drawback of Akl and Taylor's scheme is that the size of the public parameter grows linearly with the number of security classes. Latter, Mackinnon et al. (1985) presented an optimal algorithm, called the canonical assignment, to reduce the value of public parameters. However, it is difficult to find an optimal canonical algorithm. Above two schemes adopted the top-down approach to generate all secret keys. All secret keys must be re-generated when a security class is added into or deleted from the user hierarchy. The dynamic access control problems in access control cannot be efficiently solved. Harn and Lin (1990) proposed a bottom-up key generating scheme to improve the computational and storage complexities. Since then, several schemes (Chang et al., 1992, 2004; Chung et al., 2008; Das et al., 2005; Hsu et al., 2008; Hwang and Yang, 2003; Jeng and Wang, 2006; Kuo et al., 1999; Shen and Chen, 2002; Wu and Wei, 2006; Wu et al., 1995; Wu and Chang, 2001; Yang and Li, 2004) have been proposed to efficiently deal with the dynamic access control problems.

Chang et al. (1992) proposed a key assignment scheme based on Newton's interpolations method and one-way function. In their scheme, a user with higher security class must iteratively perform the key derivation process for deriving the secret keys of its lowest security class(es). It is inefficient in the key derivation

* Corresponding author at: Department of Information Management, Chang Gung University 259, Wen-Hwa 1st Road, Kwei-Shan, Tao-Yuan 333, Taiwan, ROC. Tel.: +886 3 211 8800x5827; fax: +886 3 211 8020.
E-mail address: clhsu@mail.cgu.edu.tw (C.-L. Hsu).

0164-1212/$ – see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2010.11.926

process. Wu and Chang (2001) and Shen and Chen (2002) proposed cryptographic key assignment schemes to solve the access policy using polynomial interpolations. In their schemes, the system does not need to maintain the security classes' and the users' secret keys. That is, any user can freely change his/her secret key for some security reasons. However, Hsu and Wu pointed out a security leak inherent in both schemes (Hsu and Wu, 2003). An attacker can violate the predefined access control policy to access to the unauthorized information. Latter, Yang and Li (2004) proposed a cryptographic key assignment scheme based on one-way hash function. The cryptographic key of Yang and Li's scheme is determined by one-way hash functions. Hsu et al. (2008) also pointed out some security flaws of Yang and Li's scheme to show that the claimed security requirement is violated. That is, the users can overstep his authority to access unauthorized information. Hsu et al. further proposed two improvements to eliminate the pointed out flaws.

Recently, Jeng and Wang (2006) proposed an efficient key management and derivation scheme based on the elliptic curve cryptosystems (it is denoted as the Jeng–Wang scheme for short). In Jeng–Wang scheme, the secret key of each security class can be determined by itself instead of a trusted central authority. Major advantage of Jeng–Wang scheme is to solve dynamic key management efficiently and flexibly. It is unnecessary to re-generate keys for all the security classes in the hierarchy when the security class is added into or deleted from the user hierarchy. This paper, however, pointed a compromising attack on Jeng–Wang scheme, which implies their scheme cannot achieve the claimed requirements. Finally, we proposed a secure key management scheme based on elliptic curve cryptosystem against the compromise attack. As compared with Jeng and Wang's scheme (Jeng and Wang, 2006), the proposed scheme has the following properties. (i) It is simple to execute the key generation and key derivation phases. (ii) It is easily to address dynamic access control when a security class is added into or deleted from the user hierarchy. (iii) It is secure against both interior and exterior attacks. (iv) The required storage of the public/secret parameters is constant.

The rest of this paper is sketched as follows. In Section 2, we reviewed Jeng–Wang's key management scheme and demonstrated the compromise attack on Jeng–Wang scheme. In Section 3, we proposed a secure key management scheme based on elliptic curve cryptosystem. In Section 4, we discussed the dynamic key management. We analyzed the security and performance of the proposed scheme in Sections 5 and 6, respectively. Finally, we give some conclusions.

2. The Jeng–Wang key management scheme and its security leak

In this section, we briefly reviewed Jeng and Wang's key management scheme (Jeng and Wang, 2006). We also demonstrated a compromising attack on their scheme to show that the claimed necessary security requirement is violated.

2.1. The Jeng–Wang scheme

In 2006, Jeng and Wang proposed an efficient key management and derivation scheme based on the elliptic curve cryptosystem to solve the hierarchical access control problems (Jeng and Wang, 2006). Their scheme consists of the initialization, the key generation, and the key derivation phases. In the initialization phase, a central authority (CA) determines all system parameters. In the key generation phase, each security class determines a secret point on an elliptic group over a finite field as its secret key. All secret points are sent to CA via a secure channel for constructing a key relationship derivation hierarchy. In the key derivation phase, the predecessor can use its own secret key and the public information related to the successor(s) to derive the encryption/decryption key(s) for accessing the authorized file(s). Detailed descriptions of these phases are given below.

Initialization phase – CA randomly chooses a large prime p and an elliptic curve $E_p(a, b): y^2 = x^3 + ax + b (\mathrm{mod}\, p)$ with a point O at infinity, where $a, b \in Z_p^*$ are two random integers satisfying that $4a^3 + 27b^2 \bmod p \neq 0$. Let $G \in E_p(a, b)$ be a base point of order q, where q is a large prime. CA also selects a transformation function $\tilde{A}: (x, y) \to v$ for transforming a point on $E_p(a, b)$ into a real number $v \in Z_p^*$. Finally, CA publishes $(p, q, \tilde{A}, E_p(a, b), G)$.

Key generation phase – Initially, CA determines its secret key $k_{ca} \in Z_q^*$ and publishes the corresponding public key $Y_{ca} = k_{ca}G$. Without loss of generality, let $SC = \{SC_1, SC_2, \ldots, SC_n\}$ be a user hierarchy with n disjoint sets of security classes which are partially ordered by a binary relation "$\leq$". Each security class owns two secret keys, a secret key and an encryption key. The secret key is used to derive the successor's encryption key. The encryption key is used to encrypt messages for confidentiality. Each security class SC_i (for $i = 1, 2, \ldots, n$) randomly chooses a secret key $k_{i,1} \in Z_q^*$ and an encryption key $k_{i,2} \in [1, p-1]$, and computes the corresponding public key $Y_i = k_{i,1}G$. Each security class SC_i further chooses a random integer $r_i \in Z_q^*$, computes

$$C_{i,1} = r_i G \tag{1}$$

$$C_{i,2} = (k_{i,2}, k_{i,1}) + r_i Y_{ca} \tag{2}$$

and transmits $(C_{i,1}, C_{i,2})$ to the central authority CA. The CA can derive $(k_{i,2}, k_{i,1})$ from $(C_{i,1}, C_{i,2})$ by the following equation:

$$(k_{i,2}, k_{i,1}) = C_{i,2} - k_{ca}C_{i,1} \tag{3}$$

For security class SC_i (for $i = n, n-1, \ldots, 1$), CA employs the bottom-up approach to compute

$$v_{i,j} = \tilde{A}(k_{j,1}Y_i) \tag{4}$$

and construct a polynomial $f_i(x)$ by interpolating the points $(v_{i,j}, k_{j,2})$'s for all $SC_i < SC_j$. CA finally publishes $f_i(x)$'s for $i = 1, 2, \ldots, n$.

Key derivation phase – When the security class SC_i wants to access the encrypted and held by SC_j where $SC_j < SC_i$, it can use its secret key $k_{i,1}$, the public key Y_j of SC_j, and the public information $f_i(x)$ to derive $k_{j,2} = f_j(\tilde{A}(k_{i,1}Y_j))$. With the knowledge of the encryption key $k_{i,2}$, the security class SC_i can decrypt and access the data encrypted by SC_j.

2.2. Compromising attack on Jeng–Wang scheme

First we proposed a compromising attack on Jeng–Wang scheme to show that any outsider is able to derive an unauthorized encryption key if the relationship between any two security classes is updated.

Recall Jeng–Wang scheme, each security class SC_i generates its key pair $(k_{i,1}, Y_i)$ and an encryption key $k_{i,2}$, which are contributed to construct the polynomial $f_j(x)$ for its successor SC_j where $SC_j < SC_i$. Considering the scenario of the dynamic access control management that CA can add or delete some predecessors into or from SC_j, CA will update the public polynomial as $f_j'(x)$. Let $\bar{G}_j$ be the set of the security classes SC_l's $(SC_j < SC_l)$, which still remain as the predecessors of SC_j. We can see that the secret keys belonged to the security class $SC_l \in \bar{G}_j$ are also contributed to the new polynomial $f_j'(x)$. It means that the point $(v_{l,j}, k_{j,2})$ associated with the security class $SC_l \in \bar{G}_j$ will satisfy the polynomial $\psi(v_{l,j}) = 0$, where $\psi(x) = f_j(x) - f_j'(x)$. With the knowledge of $f_j(x)$ and $f_j'(x)$, the adversary can try to derive all $v_{l,j}$'s such that $\psi(v_{l,j}) = 0$ by finding the roots