International Journal of Communication Networks and Security

Volume 1 | Issue 2 Article 9

July 2011

An Efficient and Secure Key Management Scheme for Hierarchical Access Control Based on ECC

Laxminath Tripathy
Department of Information technology, Eastern Academy of Science and Technology, Bhubaneswar, laxmi_iiit@eastodissa.ac.in

Nayan Ranjan Paul
Department of Computer Science, KMBB College of Engineering and Technology, Khurda, nayan.p@kmbb.in

Recommended Citation

Tripathy, Laxminath and Paul, Nayan Ranjan (2011) "An Efficient and Secure Key Management Scheme for Hierarchical Access Control Based on ECC," *International Journal of Communication Networks and Security*: Vol. 1 : Iss. 2 , Article 9.
DOI: 10.47893/IJCNS.2011.1020
Available at: https://www.interscience.in/ijcns/vol1/iss2/9

An Efficient and Secure Key Management Scheme for Hierarchical Access Control Based on ECC

Laxminath Tripathy[1] Nayan Ranjan Paul[2]
1Department of Information technology, Eastern Academy of Science and Technology, Bhubaneswar
2Department of Computer Science, KMBB College of Engineering and Technology, Khurda
1 laxmi_iiit@eastodissa.ac.in 2 nayan.p@kmbb.in

Abstract

In a key management scheme for hierarchy based access control, each security class having higher clearance can derive the cryptographic secret keys of its other security classes having lower clearances. In 2006 Jeng-Wang proposed an efficient scheme on access control in user hierarchy based on elliptic curve cryptosystem. Their scheme provides solution of key management efficiently for dynamic access problems. However, in this paper, we propose an attack on Jeng-Wang scheme to show that Jeng-Wang scheme is insecure against our proposed attack. We show that in our proposed attack, an attacker (adversary) who is not a user in any security class in a user hierarchy attempts to derive the secret key of a security class .

Key Words: ***Key management, Elliptic curve, Hierarchical Access control, Security, Dynamic Exterior attacks.***

1. Introduction

Hierarchical access control is a fundamental problem in computer and network systems. In a hierarchical access control, a user of higher security level class has the ability to access information items (such as message, data, files, etc.) of other users of lower security classes. A user hierarchy consists of a number n of disjoint security classes, say, $SC_1, SC_2, \ldots, SC_n$. Let this set be $SC = \{SC_1, SC_2, \ldots, SCn\}$. A binary partially ordered relation $\geq$ is defined in SC as $SC_i \geq SC_j$, which means that the security class SC_i has a security clearance higher than or equal to the security class SC_j . In addition the relation $\geq$, satisfies the following properties:

(a) [**Reflexive property**] $SC_i \geq SC_i$ □ $SC_i \in SC$.
(b) [**Anti-symmetric property**] If SC_i, $SC_j \in SC$ such that $SC_i \geq SC_j$ and $SC_j \geq SC_i$, then $SC_i = SC_j$.
(c) [**Transitive property**] If SC_i, SC_j , $SC_k \in SC$ such that $SC_i \geq SC_j$ and $SC_j \geq SC_k$, then $SC_i \geq SC_k$.

If $SC_i \geq SC_j$, we call SC_i as the predecessor of SC_j and SC_j as the successor of SC_i. If $SC_i \geq SC_k \geq SC_j$,then SC_k is an intermediate security class. In this case SC_k is the predecessor of SC_j and SC_i is the predecessor of SC_k. In a user hierarchy, the encrypted message by a successor security class is only decrypted by that successor class as well as its all predecessor security classes in that hierarchy. Akl and Taylor [1] first developed the cryptographic key assignment scheme in an arbitrary partial order set (poset) hierarchy. MacKinnon et al. [11] presented an optimal algorithm, called the canonical assignment,to reduce the value of public parameters. Harn and Lin [6] then proposed a bottom up key generating scheme,instead of using a top-down approach as in the Akl and Taylor scheme and MacKinnon et al.'s scheme. In order to solve dynamic access control problems, many schemes have been proposed in the literature[10], [7], [9], [16], [3], [13], [14], [4]. Chang et al. [3] proposed a key assignment scheme based on Newton's interpolation method and one-way hash function. In their scheme, a user with higher security clearance must iteratively perform the key derivation process for deriving the secret key of a user who is not an immediate successor. Other proposed schemes [16], [14] enhance Akl and Taylor's scheme [1], and explore other possible approaches that can enable a user in a hierarchy to modify the secret key as and when necessary. Thus,a predecessor can directly and efficiently derive the secret keys of its successor(s). Kuo et al. later developed a method [9] that employs the public key to encrypt the secret key. Their scheme has a straightforward key assignment algorithm, small storage space requirement, and uses a one-way hash function. In 2006, Jeng-Wang proposed an efficient key management and derivation scheme based on the elliptic curve cryptosystem. An attractive advantage of their scheme is that it solves dynamic key management efficiently and flexibly. However, we show that their scheme is vulnerable to dynamic exterior attack.

In this paper, we propose our dynamic exterior attack on Jeng-Wang scheme to show that their scheme is vulnerable under the proposed attack. The rest of this paper is sketched as follows. In Section 2, we review some mathematical backgrounds which are useful to review Jeng-Wang scheme. We then give briefly an overview of Jeng-Wang scheme [4] in Section 3. In Section 4, we describe our proposed dynamic exterior attack on Jeng-Wang scheme [4]. Finally, we conclude the paper in Section 5.

2. Mathematical backgrounds

In this section, we discuss the elliptic curve and its properties. We then discuss the rules for adding points on elliptic curve and the elliptic curve discrete logarithm problem. We, finally, discuss the properties of a one-way hash function.

2.1 Elliptic Curve over Finite Field

Let a and $b \in Z_p$, where $Z_p = \{0,1, \ldots, p-1\}$ and $p > 3$ be a prime, such that $4a^3 + 27b^2 \neq 0 \ (mod\ p)$. A non-singular elliptic curve $y^2 = x^3 + ax + b$ over the finite field $GF(p)$ is the set $E_p(a,b)$ of solutions $(x,y) \in Z_p \times Z_p$ to the congruence:

$$y^2 = x^3 + ax + b \ (mod\ p),$$

where a and $b \in Z_p$ are constants such that $4a^3 + 27b^2 \neq 0 \ (mod\ p)$, together with a special point $\mathcal{O}$ called the point at infinity or zero point.

The condition $4a^3 + 27b^2 \neq 0 \ (mod\ p)$ is the necessary and sufficient to ensure that the equation $x^3 + ax + b = 0$ has a non-singular solution [12]. If $4a^3 + 27b^2 \neq 0 \ (mod\ p)$, then the corresponding elliptic curve is called a singular elliptic curve. If $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be points in $E_p(a,b)$, then $P + Q = \mathcal{O}$ implies that $x_Q = x_P$ and $y_Q = -y_P$. Also , $P + \mathcal{O} = \mathcal{O} + P = P$, for all $P \in E_p(a,b)$. Moreover, an elliptic curve $E_p(a,b)$ over Z_p has roughly p points on it. More precisely, a well-known theorem due to Hasse asserts that the number of points on $E_p(a,b)$, which is denoted by $\#E$, satisfies the following inequality [15]:

$$\mathrm{p} + 1 - 2\sqrt{\mathrm{p}} \leq \#\mathrm{E} \leq \mathrm{p} + 1 + 2\sqrt{\mathrm{p}}.$$

In addition, $E_p(a,b)$ forms an abelian group or commutative group under addition modulo p operation.

2.1.1 Addition of Points on Elliptic Curve over Finite Field

We take an elliptic curve over a finite filed $GF(p)$ as $E_p(a,b) : y^2 = x^3 + ax + b \ (mod\ p)$, where a and $b \in GF(p)$. The field size p is considered as a large prime. We take G as the base point on $E_p(a,b)$ whose order is n, that is, $nG = G + G + \cdots + G(n\ times) = \mathcal{O} \ (mod\ p)$.

The elliptic curve addition differs from the general addition [8]. Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be two points on elliptic curve $y^2 = x^3 + ax + b \ (mod\ p)$, with $P \neq -Q$, then $R = (x_3, y_3) = P + Q$ is computed as follows:

$$x_3 = (\lambda^2 - x_1 - x_2)(mod\ p)$$
$$y_3 = (\lambda(x_1 - x_3) - y_1)(mod\ p),$$
$$where\ \lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} \ (mod\ p), if\ P \neq Q \\ \frac{3x_1^2 + a}{2y_1} \ (mod\ p), if P = Q \end{cases}$$

In elliptic curve cryptography, multiplication is defined as repeated additions. For example, if $P \in E_p(a,b)$, then $6P$ is computed as $6P = P + P + P + P + P + P \ (mod\ p)$.

2.2 Discrete Logarithm Problem

The discrete logarithm problem (DLP) is as follows: given an element g in a finite group G whose order is n, that is, $n = |G|$ and another element $h \in G$, find an integer x such that $g^x = h \ (mod\ n)$. It is relatively easy to calculate discrete exponentiation $g^x \ (mod\ n)$ given $g, x\ and\ n$, but it is computationally infeasible to determine x given $h, g\ and\ n$, when n is large.

2.3 Elliptic Curve Discrete Logarithm Problem

Let $E_p(a,b)$ be an elliptic curve modulo a prime p. Given two points $p \in E_p(a,b)$ and $Q = kP \in E_p(a,b)$, for some positive integer k . $Q = kP$ represents the point P on elliptic curve $E_p(a,b)$ is added to itself k times. The elliptic curve discrete logarithm problem (ECDLP) is to determine k given P and . It is relatively easy to calculate Q given k and , but it is computationally infeasible to determine k given Q and P , when the prime P is large.

2.4 One-way Hash Function

A one-way hash function $h: \{0,1\}^* \rightarrow \{0,1\}^l$ takes an arbitrary-length input $X \in \{0,1\}^*$, and produces a fixed-length (say, l-bits) output $h(X) \in \{0,1\}^l$, called the message digest. The hash function is the fingerprint of a file, a message, or other data blocks, and has the following attributes [15].

1. X can be applied to a data block of all sizes.

2. For any given variable $X, h(X)$ is easy to operate, enabling easy implementation in software and hardware.
3. The output length of $h(X)$ is fixed.
4. Deriving X from the given value $Y = h(X)$ and the given hash function $h(.)$ is computationally infeasible.
5. For any given variable , finding any $Y \neq X$ so that $h(Y) = h(X)$ is computationally infeasible.
6. Finding a pair of inputs $(X, Y), X \neq Y)$, , so that $h(X) = h(Y)$ is computationally infeasible.

3 Overview

3.1 Key generation algorithm

Step 1. Suppose there are m, m Є N, security classes in a user hierarchy over the partial-order relation(≤). CA determines an elliptic group Ep(a,b) as $y^2 = x^3 + ax + b$ (mod p), where p is a large prime number, and the coefficients satisfy $4a^3 + 27b^2 \neq 0$ (mod p). Then CA selects a base point G = (x,y) from $E_p(a,b)$ whose order is a very large value n such that nG = O. CA makes Ep(a,b), G and the value n public.

Step 2. CA selects an algorithm Ã :(x, y) → v, for representing a point on $E_p(a,b)$ as a real number v.CA makes Ã public. CA chooses a secret parameter n_{ca} and makes P_{ca} public, where $P_{ca} = n_{ca}G$.

Step 3. For security class SC_i,1≤ i≤ m, it chooses its own secret key K_i, $1 \leq K_i \leq p - 1$, and a secret parameter ni, ni < n, firstly. It makes Pi public, where Pi = niG. Then it encrypts the point (K_i,n_i) by adding kP_{ca} to it and sends the pair of points $\{kG, (K_i,n_i) + kP_{ca}\}$ to CA, where k is a positive integer selected randomly.

Step 4. For each pair of points $\{kG, (K_i,n_i) + kP_{ca}\}$,$1 \leq i \leq m$, CA multiples the first point by his secret parameter nca and subtracts the result from the second point to derive (K_i,n_i).

$(K_i; n_i) + kPc_a - nc_a(kG) = (K_i, n_i) + k(nc_aG) - nc_a(kG) = (K_i, n_i)$

Step 5. For security class SC_i, $1 \leq i \leq m$, CA constructs a polynomial $H_i(x)$ for him.

$H_i(x) = \prod_t (x - \tilde{A}(n_i P_t)) + K_i$ for all $C_i < C_t$

Example-

Figure-1 shown SC_1 determines its own secret key K_1, and its secret parameter n_1, and then generates its public parameter $P_1 = n_1$ G. Then it sends its pair of points kG, $(K_1,n_1) + kP_{ca}\}$ to CA. The other classes in the hierarchy do the same job. While CA derives all the secret keys and secret parameters, he constructs the corresponding polynomials for each class and then makes them public. The polynomials are generated as follows:

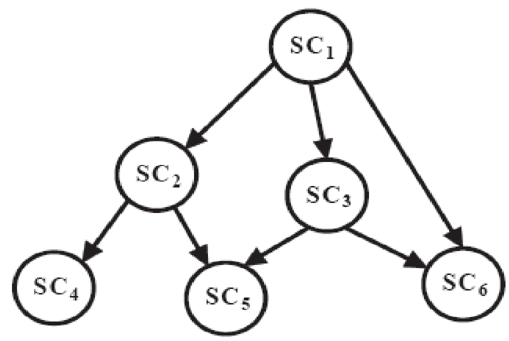

Figure-1

$H_1(x)$=nil, which means no other class has access to SC_1.
$H_2(x)=(x- \tilde{A}(n_2P_1)+k_2$.
$H_3(x)=(x- \tilde{A}(n_3P_1)+k_3$.
$H_4(x)=(x- \tilde{A}(n_4 P_2))(x- \tilde{A}(n_4 P_1))+k_4$
$H_5(x)=(x- \tilde{A}(n_5 P_2))(x- \tilde{A}(n_5 P_3))(x- \tilde{A}(n_5P_1))+k_5$.
$H_6(x)=(x- \tilde{A}(n_6P_3))(x- \tilde{A}(n_6P_1))+k_6$.

3.2 Key Derivation Algorithm-

Assume user u_i in the security class SC_i wants to access the encrypted data held by user u_j in one of his successor classes SC_j, u_i can derive the secret key K_j of u_j by the following steps:

Step 1. Get the public parameters $H_j(x)$ and P_j of u_j.

Step 2. Compute $H_j(\tilde{A}(n_iP_j)$ and then K_j can be obtained. Now suppose a user in SC_1 wants to derive the secret key K_4. Using his own secret parameter n_1 along with the public parameters $H_4(x)$ and P_4, he can derive the secret key K_4 by computing $H_4(\tilde{A}(n_1P_4)$shown below.
$H_4(\tilde{A}(n_1P_4))=(\tilde{A}(n_1P_4- \tilde{A}(n_4P_2))(\tilde{A}(n_1P_4- \tilde{A}(n_4P_1))+k_4$.
$=(\tilde{A}(n_1P_4- \tilde{A}(n_4P_2))(\tilde{A}(n_1n_4G)- \tilde{A}(n_4n_1G))+k_4$
$=k_4$

3.3 Inserting new security class-

If a new security class SC_a is inserted in to hierarchy such that $SC_i \leq SC_a \leq SC_j$..CA will do following process to update the partial relationship to manage the accessing priority when SC_a joins the hierarchy.

Step-1

For security class SC_a, it chooses its own secrete key $K_a, 1 \leq K_a \leq p-1$,and a secrete parameter n_a, $n<n$.
It makes Pa public, where $P_a=n_aG$. Then it encrypts the point (K_a,n_a) by adding KP_{ca} to it and sends the pair of points $\{KG,(K_a, n_a)+KP_{ca})\}$ to CA , where K is a positive integer selected randomly.

Step-2

For pair of points $\{KG,(K_a,n_a)+KP_{ca})\}$, CA multiplies the first point by his secrete parameter n_{ca} and subtracts the result from the second point to derive (K_a,n_a)

$$(K_a,n_a)+KP_{ca}-n_{ca}KG=(k_a,n_a) + K(n_{ca}G)-n_{ca}(KG)=(K_a,n_a)$$

Step-3

For security class SCa, ,CA constructs a polynomial Ha(x) for him
$Ha(x)= \prod_t (x - \tilde{A}(n_{ca}P_j)) + ka$ for all j satisfying SCa<SCj and j≠a.

Step-4

Determine the public polynomial Hi'(x) by following equation
$Hi'(x)= \prod_t (x - \tilde{A}(n_iP_j))\ (x - \tilde{A}(n_iP_a))$
Where $\prod t$ is performed identical to eq-1 and for each SCj such that SCi ≤SCa.

Example-

It assumes that a new security class SC_7 is inserted into the user hierarchy such
that $SC_6 \leq SC_7 \leq SC_1$ in Fig.2. Afterward the information $K_7,n_7,P_7,H_7(x),H'_6(x)$ will generate the information by using following steps.

Step-1. For security class SC_7, it chooses its own secrete key $K_7, 1 \leq K_7 \leq p-1$,and a secrete parameter n_7, $n_7<n$.
It makes P_7 public, where $P_7=n_7G$. Then it encrypts the point (K_7,n_7) by adding KP_{ca} to it and sends the pair of points $\{KG,(K_7, n_7)+KP_{ca})\}$ to CA , where K is a positive integer selected randomly.

Step-2

For pair of points $\{KG,(K_7,n_7)+KP_{ca})\}$, CA multiplies the first point by his secrete parameter n_{ca} and subtracts the result from the second point to derive (K_7,n_7)
$(K_7,n_7)+KP_{ca}-n_{ca}KG=(k_7,n_7) + K(n_{ca}G)-n_{ca}(KG)=(K_7,n_7)$

Step-3

For security class SC_7, ,CA constructs a polynomial $H_7(x)$ for him
$H_7(x)= (x- \tilde{A}(n_7 P_1))+k_7$.

Step-4

Determine the public polynomial $H_6'(x)$ by following equation
$H_6'(x)=(x- \tilde{A}(n_6P_3))(x- \tilde{A}(n_6P_1))(x- \tilde{A}(n_6P_7))+k_6$

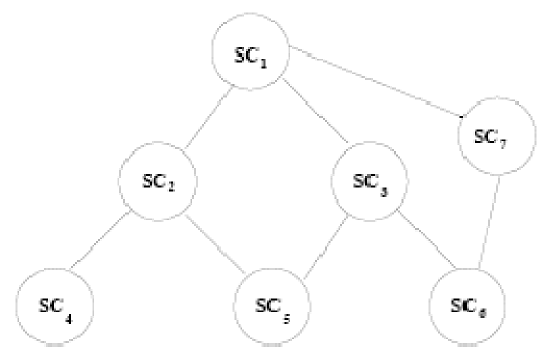

Figure-2

3.4 Deleting existing security class phases-

Suppose that a security class SC_a is to be removed from a user hierarchy such that the relationship $SC_i \leq SC_a \leq SC_j$ breaks up.

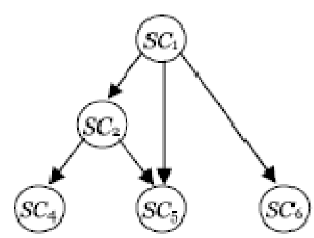

Figure-3

Suppose there is an existing security class SC_a and let it is to be removed from a user hierarchy, such that the relationship $SC_i \leq SCa \leq SC_j$ breaks up then the corresponding key Ka and parameter na,Pa are deleted.

The public polynomial Hi(x) for SCi will also be changed as follows-

H_i'(x)=$\prod_t (x - \tilde{A}(n_i P_t)) + K_i$ $.for\ all\ SC_i < SC_t$

Example-
It assumes that the existing security class SC_3 is removed from hierarchy as shown in figure-3. So the corresponding key K_3 and the parameter P_3,n_3 are deleted. The public polynomial $H_5(x)$ will be changed to H_5'(x) and $H_6(x)$ will be H_6'(6)

H_5'(x)=(x- Ã(n_5P_1))(x- Ã(n_5P_2))+K_5.

H_6'(x)=(x- Ã(n_6P_1))+K_6

4. On the security of Jeng-Wang Scheme-

However, the WWC-scheme still cannot resist another case of the exterior attack which is not discussed in [14]. Before we introducing this novel exterior attack, we must review the result of product $(X - r_1)(X - r_2) \dots (X - r_n)$ by the following theorem:

Theorem 1[2]: The product$(X - r_1)(X - r_2) \dots (X - r_n)$ can be expanded as follows.
$(X - r_1)(X - r_2) \dots (X - r_n) = \sum_{0 \le k \le n} (-1)^k\, s_k X^{n-k}$,
Where

$s_k = s_k(r_1, r_2, \dots r_n) = \sum_{1 \le i_1 < i_2 < \cdots < i_k \le n} r_{i_1}\, r_{i_2} \dots r_{i_k}$
For instance,$s_0 = 1, s_1 = r_1 + r_2 + \cdots + r_n, s_2 = \sum_{1 \le i < j \le n} r_i r_j\ and\ s_n = r_1 r_2 \dots r_n$

4.1 Dynamic Exterior Attack

When an illegal user w wishes to access the security key k_i of SC_i through the related public information when a new class joins the hierarchy.

Consider the example as shown in the figure-4.the public polynomial of SC_6 is formed $H_6(x) = \left(x - \tilde{A}(n_6 P_3)\right)\left(x - \tilde{A}(n_6 P_1)\right) + k_6$ before SC_7 joins the hierarchy. After SC_7 joins the hierarchy the public polynomial $H_6'(x) = \left(x - \tilde{A}(n_6 P_3)\right)\left(x - \tilde{A}(n_6 P_1)\right)\left(x - \tilde{A}(n_6 P_7)\right) + k_6$ and$H_7(x) = \left(x - \tilde{A}(n_7 P_1)\right) + k_7$ are formed.

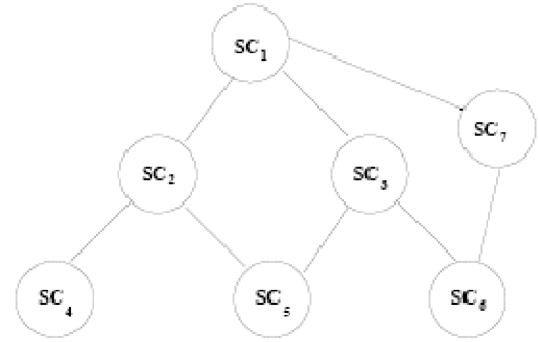

Figure-4

As $H_6(x)$ and $H_6'(x)$ are public information, so any one can obtain that information. Therefore anyone can discover the secrete key k_6 from public information by the following equations.

$H_6(x) = \left(x - \tilde{A}(n_6 P_3)\right)\left(x - \tilde{A}(n_6 P_1)\right) + k_6$
(1)

$H_6'(x) = \left(x - \tilde{A}(n_6 P_3)\right)\left(x - \tilde{A}(n_6 P_1)\right)\left(x - \tilde{A}(n_6 P_7)\right) + k_6$ (2)

Therefore from equation-1 and equation-2 we can find out the coefficient of x in $H_6(x)$ is - $\tilde{A}(n_6 P_3) - \tilde{A}(n_6 P_1)$ and the coefficient of x^2 in $H_6'(x)$ is $-\tilde{A}(n_6 P_3) - \tilde{A}(n_6 P_1) - \tilde{A}(n_6 P_7)$ respectively. Therefore we can recover the information $\tilde{A}(n_6 P_7)$ by subtracting coefficient of x^2 from the coefficient of . Then by putting $\tilde{A}(n_6 P_7)$ in equation-2, we will find out the secrete key k_6 . Hence this proposed scheme is insecure when a new security class joins the hierarchy.

4.2 On the security of removing existing security class-

Consider the example as shown in the figure-1, when the existing security class SC_3 is removed from user hierarchy, the public polynomial $H_5(x)$ of SC_5 becomes $H_5'(x)$ and $H_6(x)$ of SC_6 becomes $H_6'(x)$, whose equations are given below.

$H_5(x) = \left(x - \tilde{A}(n_5 P_2)\right)\left(x - \tilde{A}(n_5 P_3)\right)\left(x - \tilde{A}(n_5 P_1)\right) + k_5$ (3)
$H_5'(x) = \left(x - \tilde{A}(n_5 P_2)\right)\left(x - \tilde{A}(n_5 P_1)\right) + k_5$ (4)

$$H_6(x) = \left(x - \tilde{A}(n_6P_3)\right)\left(x - \tilde{A}(n_6P_1)\right) + k_6 \quad (5)$$

$$H_6'(x) = \left(x - \tilde{A}(n_6P_1)\right) + k_6 \quad (6)$$

As all above information are public so anyone can get these information.By dynamic exterior attack, the attacker can easily obtain$\tilde{A}(n_5P_3) = a_1 - b_1$ .where $a_1 = -\tilde{A}(n_5P_1) - \tilde{A}(n_5P_2)$ is the coefficient of x in equation-4 and $b_1 = -\tilde{A}(n_5P_1) - \tilde{A}(n_5P_3)$ is the coefficient of x^2 in equation-3 respectively. Similarly the attacker can also obtain $\tilde{A}(n_6P_3) = a_2 - b_2$ where $a_2 = -\tilde{A}(n_6P_1)$ and $b_2 = -\tilde{A}(n_6P_1) - \tilde{A}(n_6P_3)$ is the coefficient of x in equation-5.thetefore it is feasible for the attacker to obtain the secrete keyk_5 and k_6 with knowing the value $\tilde{A}(n_5P_3)$and $\tilde{A}(n_6P_3)$ respectively. Hence the proposed scheme is insecure when an existing security class is removed from the hierarchy.

5 Conclusion

In this paper we have shown that an illegal user can find out the secrete key when a new class joins or an existing security class is removed from the hierarchy in jeng-wang scheme.

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 (TOCS)*, 1(3):239–248, 1983.

[2]Bruce Schneier "*Applied Cryptography*" Second edition Wiley publication

[3] C. C. Chang, I. C. Lin, H. M. Tsai, and H. H.Wang. A Key Assignment Scheme for Controlling Access in Partially Ordered User Hierarchies. In *Proceedings of 18th International Conference on Advanced Information Networking and Applications (AINA'04)*, volume 2, pages 376 – 379, 2004.

[4] Y. F. Chung, H. H. Lee, F. Lai, and T. S. Chen. Access control in user hierarchy based on elliptic curve cryptosystem. *Information Sciences*, 178(1):230–243, 2008.

[5] H. Cohen. *A course in computational algebraic number theory*. Springer-Verlag, 1991.

[6] L. Harn and H. Y. Lin. A cryptographic key generation scheme for multilevel data security. *Computers & Security*, 9(6):539–546, 1990.

[7] F. G. Jeng and C. M. Wang. An efficient key-management scheme for hierarchical access control based on elliptic curve cryptosystem. *Journal of Systems and Software*, 79(8):1161–1167, 2006.

[8] N. Koblitz. Elliptic Curves Cryptosystems. *Mathematics of computation*, 48:203–209, 1987.

[9] F. H. Kuo, V. R. L. Shen, T. S. Chen, and F. Lai. Cryptographic key assignment scheme for dynamic access control in a user hierarchy. *Computers and Digital Techniques, IEE Proceedings*, 146(5):235–240, 1999.

[10] C. H. Lin. Dynamic key management schemes for access control in a hierarchy. *Computer Communications*, 20(15):1381–1385, 1997.

[11] S.J. Mackinnon, P.D. Taylor, H. Meijer, and S.G. Akl. An Optimal Algorithm for Assigning Cryptographic Keys to Control Access in a Hierarchy. *IEEE Transactions on Computers*, 34(9):797–802, 1985.

[12] R.W. D. Nickalls. A new approach to solving the cubic: Cardan's solution revealed. *The Mathematical Gazette*, 77(480):354–359, 1993.

[13] V. L. R. Shen and T. S. Chen. A novel key management scheme based on discrete logarithms and polynomial interpolations. *Computers & Security*, 21(2):164–171, 2002.

[14] V. L. R. Shen, T. S. Chen, and F. Lai. Novel cryptographic key assignment scheme for dynamic access control in a hierarchy. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E80-A(10):2035–2037, 1997.

[15] W. Stallings. *Cryptography and Network Security: Principles and Practices*. Prentice Hall, 3rd edition, 2003.

[16] H. M. Tsai and C. C. Chang. A cryptographic implementation for dynamic access control in a user hierarchy. *Computers & Security*, 14(2):159–166, 1995.