초 타원 곡선 암호시스템에서 동시 역원 알고리즘을 가진 안전한 스칼라 곱셈

박 택 진*

Secure Scalar Multiplication with Simultaneous Inversion Algorithm in Hyperelliptic Curve Cryptosystem

Taek-Jin Park*

요 약

유비쿼터스 환경에서 계산의 복잡성,메모리,전력소비등의 제약성으로 인하여 공개키 암호시스템을 적용하기는 매우 어렵다. 초타원 곡선 암호시스템은 RSA나 ECC보다 짧은 비트 길이를 가지고 동일한 안전성을 제공한다. 초타원 곡선 암호시스템에서 스칼라 곱셈은 핵심적인 연산이다. T.Lange는 다수의 좌표를 사용하여 초타원 곡선 암호시스템에서 역원 연산이 없는 스칼라 곱셈 알고리즘을 개발 하였다.그러나 다수의 좌표를 사용하는 것은 SCA에 노출되고 더 많은 메모리가 요구 된다. 본 논문에서는 초 타원곡선 암호시스템에서 동시 역원알고리즘을 가진 안전한 스칼라 곱셈 알고리즘을 개발하였다. 안전성 과 성능을 위하여 동시역원 알고리즘을 적용하였다 개발한 알고리즘은 SPA와 DPA 에 안전하다.

ABSTRACT

Public key cryptosystem applications are very difficult in Ubiquitos environments due to computational complexity, memory and power constrains. HECC offers the same of levels of security with much shorter bit-lengths than RSA or ECC. Scalar multiplication is the core operation in HECC. T.Lange proposed inverse free scalar multiplication on genus 2 HECC. However, further coordinate must be access to SCA and need more storage space. This paper developed secure scalar multiplication algorithm with simultaneous inversion algorithm in HECC. To improve the over all performance and security, the proposed algorithm adopt the comparable technique of the simultaneous inversion algorithm. The proposed algorithm is resistant to DPA and SPA.

Keywords : HECC, Jacobian, Scalar multiplication, Simultaneously inversion algorithm, SCA

* 강릉영동대학교 의료전자과 (tjpark@gyc.ac.kr)
접수일자 : 2011년 09월 17일, 수정일자 : 2011년 11월 18일, 심사완료일자 : 2011년 12월 10일

Ⅰ. Introduction

Elliptic Curve Cryptosystem (ECC) have now widely been studied and applied in e-commerce, e-government and other secure communications. The practical advantages of ECC is that it can be realized with much smaller parameter compared to the conventional Discrete Logarithm Problem(DLP) based cryptosystems or RSA, but with the same levels of security[1].

Hyperelliptic curves, a generalization of elliptic curves, require decreasing field size as genus increases. Hyperelliptic curves of genus g achieve equivalent security of ECC with field size $1/g$ times the size of field of ECC for $g < 4$[2][3]. A hyperelliptic curve has genus $g \geq 1$. A hyperelliptic curve of genus $g = 1$ is same as an elliptic curve. HyperEllipticCurve Cryptosystem (HECC) was proposed by N. Koblitz[4].

In the environments with limited prosessing power, memory and band width. HECC offers the same of levels of security with much shorter bit-lengths than RSA or ECC. For example, such as ECC, if an 160 bit security is desired, then the underlying field should have oder approximately $2^{160}, 2^{80}, 2^{54}$ for $g = 1,2,3$ respectively.

Thus hyperelliptic curves are allow for operand bit-lengths $50 \sim 80$ bit.

These cryptosystems are realized in imbedded processors with limited resource (moibile phone, smart phone, PDA,smart card ect.) and channels with limited bandwith under constrained environments. In constrained environments, inversions are extremely critical problems which need more space storage and running times. An example are smart cards, as usually multiplications are optimized there, whereas inversion sare very slow even with coprocessors. The proposed algirithm compute several inversions simultaneously in HECC. Only one previous work can found in inverse free arithmatic on genus 2 hyperelliptic curve, but further coordinate must be access to SCA(Side Chennel Attacks), and more stroage space. The proposed secure scalar multiplication resistant to SCA.

Ⅱ. Preliminaries

2.1 Jacobian of hyperelliptic curves

Let q be a power of some prime and F_q be the finite field of q elements.

A hyperelliptic curve C of genus g over F_q $(g \geq 1)$ is defined by the Weierstrass equation

$$C: y^2 + h(x)y = f(x) \quad , \qquad (1)$$

where $h(x) \in F$ is a polynomial of degree at most g, $f(x) \in F[x]$ is a monic polynomial of degree $2g+1$. There is no solutions $(a,b) \in \overline{F} \times \overline{F}$ which simultaneously satisfy the equation

$$b^2 + h(a)b = f(a) \qquad (2)$$

and the partial differential equations $2b + h(a) = 0$

and $h'(a)b - f'(a) = 0$.

F : a finite field.

$\overline{F}$: the algebraic closure of F.

A singular point on C is a pair $(a,b) \in \overline{F} \times \overline{F}$ which simultaneously satisfy the equation $b^2 + h(a)b = f(a)$

and partial differential equation $2b + h(a) = 0$

and $h'(a)b - f'(a) = 0$.

A Divisor is finite formal sum of

$F_q - points.$

$$D = \sum m_i p_i \qquad (3)$$

Divisor : D
$= m_1P_1 + m_2P_2 + m_3P_3 + \ldots, m_ip_i$ on C with its degree defined as the integer $\sum m_i$.

The Jacobian group$J_c(F_q)$ of the curve C over F_q is an Abelian group composed of reduced divisors on C. Every element or reduced divisor D in $J_c(F_q)$ can be uniquely expressed by a pair of polynomials

$< a(u), b(u) >$ with the properties

$$\begin{cases} \deg_u b(u) < \deg_u a(u) \le g \\ b(u)^2 + h(u)b(u) - f(u) = \mathrm{mod}\, a(u) \end{cases} ,(4)$$

where $a(u), b(u) \in F_q[u]$.

Generally,$a(u)$ monic polynomial of degree g and $b(u)$ is a polynomial of degree $g-1$. The element of $J_c(F_q)$ can be expressed as $< 1,0 >$.

2.2 DLP on the Jacobian

Computing the group order of the Jacobian is believed to be a computationally hard task because it involves counting the number of rational points of a given hyperelliptic curve over an extension field of a base field of degree up to genus.

DLP on the Jacobians of hyperelliptic curves is point multiplication by an integer k, namely computing kD for a point D on the Jacobian. This operation is called scalar multiplication.

The most time consuming operation in HECC is the scalar multiplication.

In practical HECC, the critical computation that dominates the whole running time is scalar multiplication, that is, the computation of the repeated divisor adding

$kD = \overbrace{D + D + D + \ldots}^{k\ time}$ (5) for a given divisor $D \in F_{q^n}$ and a positive integer $k \ge 1$, which is denoted as kD.

III. Algorithm HEC(A,D)

Simultaneous addition and doubling algorithm was first proposed by Izu and Takagi [5],[6] for elliptic curve scalar multiplication. The algorithm $HEC(A,D)$ computes the addition $D_3 = D_1 + D_2$ and the doubling $D_3 = 2D_1$ with simultaneous inversion. D_1, D_2 are divisors of Jacobian. The Jacobian of a hyperelliptic curve defined over a finite field. This algorithm is used to construct an algorithm for scalar multiplication. A major obstacle in the implementation of HECC is the problem of performing efficient arithemetic in the Jacobians of hyperelliptic curves. Recently to avoid these field inversions several point representations for HECC divisors have been proposed in Tanja Lange([7],[9]). In affine coordinates HECC operations require one field inversion each. Over prime fields, field inversion is very expensive operation. For genus 2 HECC curve, Tanja Lange gave explicit formulae which need $1[I] + 22[M] + 3[S]$ for a group addition and $1[I] + 22[M] + 5[S]$ for a group doubling in affine coordinate system(Where I : Inversion, M : multiplication and S : Squaring). On the other hand, in [7] the explicit formula is developed for inversion free arithmetic in the Jacobian, the authors introduced a further coordinate called Z to represent the elements of the divisor class group and obtain explicit formulae in projective coordinate system. In [7], $47[M] + 4[S]$ are required for a group addition and $40[M] + 6[S]$for a group doubling and inverse free arithmetic.

3.1 Simultaneous Inverses

Suppose we wish to invert to α and β. Instead of computing two inverses we can compute the product $\alpha\beta$ and compute its inverse.

Then we can compute inverse of α and β by two more multiplications : $\alpha^{-1} = (\alpha\beta)^{-1}\beta$ and $\beta^{-1} = (\alpha\beta)^{-1}\alpha$. Thus inverses of two field elements can be computed at the cost of one inversion and three multiplications. This trick and its elegant generalization to n elements is called Montgomery's trick[8].

It takes one inversion and $3(n-1)$ multiplications to compute the inverse of n elements using the Montgomery trick.

In Montgomery trick are computed as follow.

Let $x_1, x_2, \ldots, x_n$ be the elements to be inverted. Set $\alpha_1 = x_1$ and for $i = 2, \ldots, n$ compute $a_i = a_{i-1}x_i$. Then invert a_n and compute

$$x_n^{-1} = a_{n-1}a_n^{-1}. \qquad (6)$$

Now, for $i = n-1, n-2, \ldots, 2,$ compute $a_i^{-1} = x_{i+1}a_{i+1}^{-1}$ and $x_i^{-1} = a_{i-1}a_i^{-1}$. Finally compute $x_1^{-1} = a_1^{-1} = x_2a_2^{-1}$.

This procedure provides $x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}$ using a total of $3(n-1)$ multiplications and one inversion.

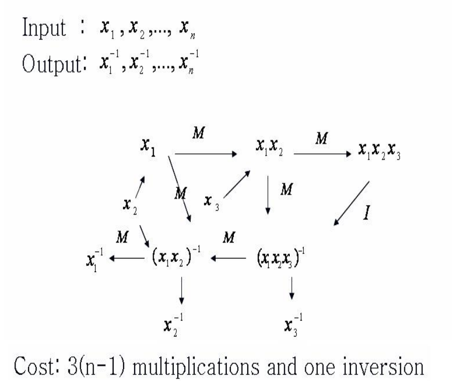
Input : $x_1, x_2, \ldots, x_n$
Output: $x_1^{-1}, x_2^{-1}, \ldots, x_n^{-1}$
x_1
x_2
x_3
x_1x_2
$x_1x_2x_3$
M
I
$(x_1x_2)^{-1}$
$(x_1x_2x_3)^{-1}$
x_1^{-1}
x_2^{-1}
x_3^{-1}
Cost: 3(n-1) multiplications and one inversion

Figure 1 : Simultaneous Inverses

3.2 *HECADD* and *HECDBL*

The explicit formulae to add and double divisors in HECC have a long process of evolution. Let us consider the addition and doubling algorithm in [9], in the most general and frequent case. These can be divided into three parts.

Part 1 : Some multiplications and squarings of the underlying field elements are carried out.

Part 2 : A field element generated in part 1 is inverted.

Part 3 : The inverse so obtained in part 2 is used in part 3 along with some more multiplications and squarings of field elements. The output of part three provides the required divisor.

ADD_1	DBL_1
ADD_2 One inversion	DBL_2 One inversion
ADD_3	DBL_3

Figure 2 : HECADD and HECDBL [9]

Let us name the modules of these algorithms as ADD_1, ADD_2, ADD_3 (part of addition algorithms) and DBL_1, DBL_2, DBL_3 (parts of doubling algorithms). Note that in ADD_1 and ADD_3 we need to compute a total of 22 multiplications and 3 squarings([10],[11]). In DBL_1 and DBL_3 we compute a total of 22 multiplications and 5 squarings . In each of ADD_2 and DBL_2 we compute only 1 inversions. We describe the following simultaneous add and double algorithm. We will use the following protocol. Suppose D_1, D_2 are divisors and we are required to compute $D_1 + D_2$ and $2D_1$. By $\alpha = ADD_1(D_1, D_2)$ we denote the field element α that is to be inverted in part ADD_2. By $ADD_{RLT} = ADD_3(D_1, D_2)$, we denote the result of the equation,

$$ADD_{RLT} = ADD_3 = D_1 + D_2. \tag{7}$$

Similarly, by$\beta = DBL_1(D_1)$ we denote the field element β which is to be inverted in part DBL_2 and $DBL_{RLT} = DBL_3(D_1)$ denotes the result of the equation

$$DBL_{RLT} = 2D_1. \tag{8}$$

```
Algorithm 1 $HEC(A, D)$
Input : $D_1$ and $D_2$.
Output : $D_1 + D_2$ and $2D_1$.
$[AD_1]$ : Let $\alpha = ADD_1(D_1, D_2)$ .
$[AD_2]$ : Let $\beta = DBL_1(D_1)$.
$[AD_3]$ : Compute $\alpha^{-1}$ and $\beta^{-1}$
using the method of the simultaneous inversion algorithm
$[AD_4]$ : Let $ADD_{RLT} = ADD_3(D_1, D_2)$.
$[AD_5]$ : Let $DBL_{RLT} = DBL_3(D_1)$.
$[AD_6]$ : Return $(ADD_{RLT}, DBL_{RLT})$.
```

Proposition 1 : Algorithm 1 $HEC(A, D)$ computes $D_1 + D_2$ and $2D_1$ with a cost of $1[I] + 47[M] + 8[S]$. Algorithm $HEC(A, D)$ outputs $D_1 + D_2$ and $2D_1$.

Proof : Computational cost of steps $[AD_1]$, $[AD_2]$,$[AD_4]$ and $[AD_5]$ is the sum of those of $ADD_1, ADD_3, DBL_1, DBL_3$. The cost of $[AD_1]$,$[AD_2]$,$[AD_4]$ and $[AD_5]$ is 44 multiplications and 8 squarings. The cost of step $[AD_3]$, by simultaneous inversion algorithm, is $1[I] + 3[M]$. Thus, the total cost of the algorithm is $1[I] + 47[M] + 8[S]$.

In simultaneous add and double algorithm, generally $1[I]$ is considered computationally equivalent to $30[M]$ and $1[S]$ is considered equivalent to $0.8[M]$([12],[13],[14]).

IV. Proposed Algorithm

4.1 Precomputation

In most HECC, the base divisor D is fixed and k varies from user to user. Therefore, the cryptographic system allows some pre-computations involving D and is capable of accumulating some precomputed values as a part of the system setup.

Let $(k_{l-1}, \ldots, k_1, k_0)$ be binary representation of an integer k, i.e.,$k = \sum_{i=0}^{i=l-1} k_i 2^i$, with $k_i \in 0, 1$, and let w be an integer such as $w \geq 2$;

we set $d = [\frac{l}{w}]$.

D be a divisor, for all $[b_{w-1},\dots,b_1,b_0]D \in Z_2^w$. we define

$$[b_{w-1},\dots,b_1,b_0] = b_0 + b_1 2^d + b_2 2^{2d} +,\dots, b_{w-1} 2^{(w-1)d} \quad . \tag{9}$$

We generate the sequence of points $[b_{w-1},\dots,b_1,b_0]D$, for all $[b_{w-1},\dots,b_1,b_0]D \in Z_2^w$, such as

$$[b_{w-1},\dots,b_1,b_0]D{=}b_{w-1}2^{(w-1)d}D+,\dots, + b_2 2^{2d}D + b_1 2^d + b_0 D$$

4.2 Precomputation Phase

Step 1 : Compute $2^d D, 2^{2d}D, \dots, 2^{(w-1)d}D$ which cost $(w-1)d$ doubling operations.

Step 2 : Consists in computing all possible combinations $\sum_{i=r}^{i=s} b_i 2^{id}$,

with $b_i \in 0,1$, and $0 \le r < s \le w-1$.

The number of these combinations is $2^w - w$. Thus, the cost of Step 2 is $2^w - w$ adding operations.

```
Algorithm 2  Proposed Algorithm
Input : $k = (k_{l-1},\dots,k_1,k_0)_2$
and $D$.
Output : $E = kD$.
1.(Precomputation) compute $[b_{w-1},\dots,b_1,b_0]D$, for all $[b_{w-1},\dots,b_1,b_0]D \in Z_2^w$.
2. If $k \bmod 2 = 0$ then $k' \leftarrow k+1$ else $k' \leftarrow k+2$ .
3. For $i$ from $l-2$ down to 0 do
    3.1 $E[0] = D$, $E[1] = 2D$
    3.2
        Return $E[0]$.
4. $D' \leftarrow 2D$.
    $(E[b_i \oplus 1], E[b_i])$
      $= HEC(A,D)(E[b_i], E[b_i \oplus 1])$
5. If $k \bmod 2 = 0$ then return $E - D$ else return $E - D'$.
```

V. Algorithm Security and Complexity

5.1 Security against SCA SCA(Side Channel Attack) is a persistent threat to the implementations of a cryptosystem. The perfect SCA-Resistance does not exist. However, by using appropriate countermeasures, it is possible to make the attacker's task harder.

5.2 Security against SPA

SPA(Simple Powr Analysis) uses only a single observed information, while the DPA(Differential Power Analysis) uses a lot of observed information together with the statistic tools. SPA can detect whether the secret scalar is even or not. To deal with this threat, The proposed algorithm convert the scalar k to $k' \leftarrow k+2$.

- Computation

If k is odd : $((k+2)D - 2D) \rightarrow kD$.

As shown before, all bit strings representing k, are different from zero. Thus, the proposed algorithm computes the scalar multiplication with a uniform behaviour.

The Proposed algorithm uses $HEC(A,D)$ algorithm(simultaneous addition and doubling with simultaneous inversion algorithm in HECC) and new bit-string representation of k against SPA. Hence, it is secure against SPA.

5.3 Security against DPA

Even if a scheme is SPA-Resistant, it is not always DPA- Resistant, because the DPA uses not only a simple power trace but also a statistic analysis, which has been captured by several executions of the SPA. A scalar multi-plication algorithm which is protected against SCA may still be vulnerable to DPA.

In Algorithm 2 (Step 3), both input and output for $HEC(A,D)$ are ordered pairs, which are, $(E[0],E[1])$ for $b_i=1$ and $(E[1],E[0])$ for $b_i=0$. $HEC(A,D)$ outputs the pair $(E[0]+E[1],2E(b_i))$. Suppose the value of bit b_i has been assigned in two bits, $b_i=0$ and $b_i=1$, XOR-ing of two bits yield a value of bit b_i.

Then the values are assigned as follows : $E[0]=E[2]$, $E[1]=E[1]$ if $b_i=0$, and $E[0]=E[1]$,$E[1]=E[2]$ if $b_i=1$. It is clear that the computation of $b_i=0$, and $b_i=1$, is not correlated. Thus the power con-sumption trace in each iteration is independent of the bit b_i.

Average power trace will not be correlated with the measured power traces. Clearly, b_i,'s value can not be predicted.

5.4 Complexity

The most expensive operation in a crypto-system based on the DLP for the Jocobian of a HECC is point multiplication by an integer k. The n bit binary representation of k denote $(k_{l-1},\dots,k_1,k_0)_2$.

A standard Double-and-Add(DBL_
AND _ADD) algorithm can computer the scalar multiplication, but it is not secure against the Time Attack(TA)[15].
The Double-and-always-Add(DBL

Arithmetic	Algorithm	DPA-Resistant	SPA-Resistant	Complexity	Remark
Inverse Free [7]	DBL_AND_ADD	×	×	$69.9n[M]$	$(44.8n+50.2(n/2))[M] \approx 69.9n[M]$
	DBL_AND_ALWS_ADD	×	○	$95.9n[M]$	$(44.8n+50.2n)[M] \approx 95n[M]$
Simultaneous Inverse	Proposed Algorithm	○	○	$(83.4n+56)[M]$	

Table 1 : Complexity and Security of the Other Algorithms (n bits)

-AND_ALWS_ADD)algorithm can resist the TA. TA can be regarded as a class of SPA[16].

In [7], Inverse Free algorithm,

$HECDBL$=$47M+4S$=$50.2M$,

$HECADD$=$40M+6S$=$44.8M$,

where $1[S]=0.8[M]$.

Total cost of DBL_AND__ADD algorithm is $(44.8n+50.2(n/2))M \approx 69.9n[M]$ and DBL_AND_ALWS_ADD is $(44.8n+50.2(2n))M \approx 95n[M]$.

In proposed algorithm, $HECADD = 1[I]+22[M]+3[S]$, $HECDBL = 1[I]+22[M]+5[S]$.

For each bit, cost of Algorithm $HEC(A,D)$ is $1[I]+47[M]+8[S] \approx 83.4[M]$,

where$1[I]=30[M]$

and $1[S]=0.8[M]$.

Besides there is a doubling at beginning, which costs $1[I]+22[M]+5[S] \approx 56[M]$.

So total cost of the computation is $(83.4n+56)[M]$.

Although average case complexity of [7] is better than proposed algorithm, which is not DPA- and SPA-Resistant.

VI. Conclusion

This paper developed an efficient and secure scalar multiplication algorithm in Genus 2 HECC. It is more secure to use formulae which involve inversion of field elements rather than inversion free arithmetic. An important contribution of the proposed scalar multiplication algorithm is a secure algorithm with simultaneous inversion algorithm in HECC.

References

[1] L. You and J. Zeng. Fast Scalar Multiplications on Hyperelliptic Curve Cryptosystems. Informatica 34, pp.227 - 236. 2010.

[2] B. Balamohan. Accelerating the scalar Multiplication on genus 2 HyperellipticCurve Cryptosystems. In Cryptology ePrint Archive Report no. 578. 2011.

[3] S. Erickson, M. J. Jacobson, N. Shang. and S. Shen A. Stein. Efficient formulas for real hyperelliptic curves of genus 2 in affine representation. In C. Carlet and B. Sunar, editors, Arithmetic of finite fields, volume 4547 of Lecture Notes in Computer Science, pages 202 - 218. Springer Berlin / Heidelberg, 2010.

[4] N.Koblitz. Hyperelliptic cryptosystems. In Ernest F.Brickell,editor, Journal of Cryptology, No,pp.139-150. 1989.

[5] T. Izu and T. Takagi. A Fast Parallel Elliptic Curve Multiplication Resistant against Side Channel Attacks.
Technical Report CORR 2002-03, University of waterloo, 2002.

[6] T. izsu, B. Moller and T. Takagi. Improved Elliptic Curve Multiplication Methods Resistant Against Side Channel Attacks. Proceedings of Indocrypt 2002, LNCS 2551,pp 296-313. 2002.

[7] T. Lange. Inversion-Free Arithmetic on Genus 2 Hyperelliptic Curves. ePrint Archive, Report 2002/147, 2002.

[8] Pradeep Kumar Mishra and Palash Sarkar. Application of Montgomery's Trick to Scalar Multiplication for Elliptic and Hyperelliptic Curves Using a Fixed Base Point. Cryptology Research Group, Applied Statistics Unit, Indian Statistical Institute.

Technical Report Number CRG/2003/16, September, 2003.

[9] T. Lange. Efficient Arithmetic on Hyperelliptic curves. PhD thesis, Universit ä t Gesamthochschule Essen, 2001.

[10] T. Lange and M. Stevens, " Efficient doubling on genus 2 curves over binary fields" Selected Areas in Cryptography-SAC 2004, Springer Verlag, 2004.

[11] T. Lange. Formulae for arithmetic on genus 2 hyperelliptic curves, J. AAECC-Applicable Algebra in Engineering, communication and Computing, vol.15, no.5, pp.295-328, 2005.

[12] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone Handbook of Applied Cryptography. CRC Press, 1997.

[13] K. Okeya and K. Sakurai. Efficient Elliptic Curve Cryptosystems from a Scalar Multiplication Algorithm with Recovery of the y-coordinate on a Montgomery from Elliptic Curve In CHES 2001, LNCS 2162,pp 126-141, Springer-Verlag 2001.

[14] M. Katagi, I. Kitamura, T. Akishita, and T. Takagi ''Novel efficient implementations of hyperelliptic curve cryptosystems using degenerate divisors'' WISA 2004, LNCS3325, pp. 347-361, Springer-Verlag, 2005.

[15] P.Kocher. Timing attacks on Implementations of Diffie-Hellman, RSA,DSS and Other System, CRYPTO'96,LNCS 1109, pp.104-113, Springer-Velag, 1996.

[16] Mustapha Hedabou , Pierre Pinel , Lucien Beneteau, A comb method to render ECC resistant against Side Channel Attacks In Cryptology ePrint Archive, Report no 342. 2002.

저자약력

박 택 진(Taek-Jin Park)

2005 KAIST/성균관대학교 전기전자 컴퓨터공학과 박사

1987.1~1993.2 한국통신 기술과장

1993.3~ 현재 강릉영동대학교 의료전자과 부교수

<관심분야> 정보보호 및 암호, 암호 알고리즘, 초타원곡선 암호, 해쉬 함수등