3-3 A Generalized Harley Algorithm for Genus Two Hyperelliptic Curves

SUGIZAKI Hiroki, MATSUO Kazuto, CHAO Jinhui, and TSUJII Shigeo

A fast addition algorithm for divisor classes of genus two hyperelliptic curves over finite fields of odd characteristics was proposed by Harley in 2000 and a lot of improvements of the algorithm has been proposed, besides extensions of the algorithm for the curves over finite fields of characteristic two have been proposed by the authors and Lange independently. However, any Harley algorithm over arbitrary characteristic fields have not been known until now. This paper shows a generalization of the Harley algorithm to genus two hyper elliptic curves over finite fields of arbitrary characteristics. The proposed algorithm takes I+26M, I+31M for an addition and a doubling respectively, where I and M denote the cost for an inversion and a multiplication over the definition field respectively.

Keywords

Hyperelliptic curve cryptosystems, Hyperelliptic curves, Addition algorithm, Cantor algorithm, Harley algorithm

1 Introduction

In order to realize secure electronic commerce within public digital communication infrastructures such as the Internet, information security techniques are indispensable, covering areas from secret communications to personal privacy protection to authentication. Among these techniques, public key cryptography is an essential basic technology. This core technology supports a range of societal structures, such as electronic payment systems, electronic government, and advanced healthcare.

The best public key cryptography currently available is the elliptic curve cryptosystem. However, the elliptic curves themselves used in this method form only a small class of all algebraic curves. Thus in order to construct a more efficient public key cryptosystem, many researches have recently been conducted on cryptosystems using the discrete logarithm problems on hyperelliptic curves, a more general class of algebraic curves.

A fast group operation (addition) algorithm is indispensable when constructing a cryptosystem based on the discrete logarithm problem with elliptic curves and hyperelliptic curves. For elliptic curves, many practical algorithms have long been used. However, no efficient algorithms were available for hyperelliptic curves. Recently, Cantor[1] proposed a fast addition algorithm (the Cantor algorithm) for divisor classes of hyperelliptic curves. This algorithm has enabled use of hyperelliptic curves in the construction of practical cryptosystems. However, the hyperelliptic curve cryptosystems based on the Cantor algorithm are at least several times slower than elliptic curve cryptosystems.

In 2000, Harley[2][3] proposed a fast addition algorithm on genus two hyperelliptic curves (the Harley algorithm) based on a method different from that used in the Cantor algorithm. Unlike the Cantor algorithm, the Harley algorithm limits itself to genus two

hyperelliptic curves over odd-characteristic fields. Similar to the Cantor algorithm, the Harley algorithm uses Mumford's representation to represent the input and output divisor classes, but it uses a method similar to the chord-tangent law for elliptic curves, instead of using the binary quadratic form computation of polynomials applied in the Cantor algorithm. The Harley algorithm also uses the Karatsuba multiplication, as well as Chinese reminder theorem and the Newton iteration, in the computation procedure, to obtain an algorithm that is faster than the Cantor algorithm. Consequently, it has been shown that the hyperelliptic curve cryptosystems based on the Harley algorithm can theoretically match the speed of elliptic curve cryptosystems[4]. Since then, the Harley algorithm has been the subject of extensive study *1. Specifically, References[8] and[9] present extended algorithms developed from the Harley algorithm for hyperelliptic curves over fields of characteristic 2. As such, the Harley algorithm has been subject to separate study for odd-characteristic definition fields and for even-characteristic definition fields. Nevertheless, it is important to construct a general algorithm for hyperelliptic curves over finite fields of arbitrary characteristics in order to improve the research and our understanding of addition algorithm. The general algorithm is also useful for practical implementation in symbolic computation software.

Therefore, this paper shows a Harley algorithm that does not depend on the characteristics of the definition field. The proposed algorithm enables addition and doubling for divisor classes of genus two hyperelliptic curves over the finite field $\mathbf{F}_q$ of an arbitrary characteristic at the computational cost of I+26M and I+31M, respectively. Here, I and M denote the cost for inversion and multiplication over $\mathbf{F}_q$, respectively.

*1 For the latest research relating to the Harley algorithm, see[5][6], and[7], for example. All the divisors D of C obviously form a commutative group.

2 Preliminaries

2.1 Hyperelliptic curves

Let $\mathbf{F}_q$ be a finite field with q elements. A genus two hyperelliptic curve C over $\mathbf{F}_q$ is defined as follows:

$$\begin{aligned} C \ : \ & Y^2 + H(X)Y = F(X), \\ & H(X) = X^2 + h_1X + h_0, \\ & F(X) = X^5 + f_4X^4 + \cdots + f_1X + f_0. \end{aligned} \tag{1}$$

Condition

$$(x, y) \in \overline{\mathbf{F}}_q{}^2 \text{ such that } y^2 + H(x)y + F(x) = 0$$

Let us define points on C as the pair (x, y) that satisfies the above condition together with the only point at infinity, P_∞. For point $P = (x, y) \neq P_\infty$ on C, $(x, -y - H(x))$ is another point on C. Here, this point is represented as $-P$. Also, $-P_\infty = P_\infty$ holds. A P such that $P = -P$ is called a ramification point. For a ramification point $P = (x, y)$ other than P_∞,

$$2y + H(x) = 0 \tag{2}$$

holds. A ramification point on C is either a point (x, y) on C that satisfies (2), or P_∞.

2.2 Divisors of hyperelliptic curves and the Jacobian variety

Let us define the divisor D on C as the following finite formal sum of points P_i on C.

$$D = \sum_{P_i \in C} n_i P_i, \ n_i \in \mathbf{Z}. \tag{3}$$

Let us define the degree of the divisor D given by (3) as

$$\deg D = \sum_i n_i .$$

The set $\mathbf{D}^0$ consisting of all divisors of degree 0 forms a group.

Let us define the divisor (f) of a rational function f on C as

$$(f) = \sum m_{P_i} P_i - \sum m_{Q_j} Q_j .$$

Here, P_i is a zero of multiplicity m_{P_i} of f on C, and Q_j is a pole of multiplicity m_{Q_j}. The divisor (f) is called a principal divisor. Let us denote the set of all principal divisors as $\mathbf{D}^l$. $\mathbf{D}^l$ is known to form a subgroup of $\mathbf{D}^0$.

The Jacobian variety $\mathbf{J}_c$ of C is defined as

follows:

$$\mathbf{J}_C = \mathbf{D}^0/\mathbf{D}^l.$$

Let us denote the set of divisor classes of $\mathbf{J}_c$ fixed by a q-th power Frobenius map as $\mathbf{J}_c(\mathbf{F}_q)$. $\mathbf{J}_c(\mathbf{F}_q)$ forms a finite commutative group. As the discrete logarithm problem can be defined for this group, a cryptosystem can be constructed on the group. This cryptosystem is called a hyperelliptic curve cryptosystem. To construct a fast hyperelliptic curve cryptosystem, a fast addition algorithm is needed for the divisor classes of $\mathbf{J}_c(\mathbf{F}_q)$. Thus, this paper investigates addition for the divisor classes of $\mathbf{J}_c(\mathbf{F}_q)$.

2.3 Divisor classes and their expressions

For D_1 and D_2 that satisfy $D_1, D_2 \in \mathbf{D}_0$, if $D_1 - D_2 \in \mathbf{D}^l$ is true, D_1 and D_2 are said to be linearly equivalent and represented by $D_1 \sim D_2$.

Any divisor classes of $\mathbf{J}_c$ can be represented by divisors of the following form.

$$D = \sum_i m_i P_i - \left(\sum_i m_i\right) P_\infty,\ m_i \geq 0.$$

Here, $P_i \neq -P_j$ is assumed to hold for $i \neq j$.

A divisor in the form of (4) is called a semi-reduced divisor. Specifically, a semi-reduced divisor that satisfies $\sum_i m_i \leq g$ called a reduced divisor. Reduced divisors can uniquely represent the divisor classes of $\mathbf{J}_c$.

Using polynomials $U, V \in \bar{\mathbf{F}}_q[X]$, a semi-reduced divisor D given by the form of (4) can be represented as

$$D = (U, V). \tag{5}$$

Here, denoting $P_i = (x_i, y_i)$,

$$U = \prod (X - x_i)^{m_i} \tag{6}$$

and V is the only polynomial that satisfies

$$\begin{aligned} F - HV - V^2 &\equiv 0 \bmod U, \\ \deg V &< \deg U, \\ y_i &= V(x_i). \end{aligned} \tag{7}$$

This representation of a semi-reduced divisor is called Mumford's representation.

For $D = (U, V)$, $U, V \in \mathbf{F}_q[X]$ is equivalent to $D \in \mathbf{J}_c(\mathbf{F}_q)$, so that in the following discussion we assume $U, V \in \mathbf{F}_q[X]$.

When a divisor class D of $\mathbf{J}_c(\mathbf{F}_q)$ is represented by Mumford's representation, its inverse $-D$ is easily determined. Specifically, for $D = (U, V)$ ($\deg U = 2$),

$$-D = (U, U - V - H) \tag{8}$$

holds. Specifically, when $D = P_r + P'_r - 2P_\infty$ for ramification points P_r and P'_r,

$$-D = D$$

also holds. For $D = (X + u_0, v_0)$ ($\deg U = 1$),

$$-D = (X + u_0, -v_0 - H(u_0))$$

holds.

3 Generalization of the Harley algorithm

This section proposes a Harley algorithm for genus two hyperelliptic curves on $\mathbf{F}_q$ given by (1). Similar to the Harley algorithm over odd-characteristic fields, the proposed algorithm uses different procedures for addition and for doubling. It also uses reduced divisors for input and output.

This section is organized as follows: Section **3.1** describes the classification of the input divisors. Section **3.2** discusses the details of the addition algorithm. Section **3.3** discusses the details of the doubling algorithm.

In the following, lower-case characters denote elements of $\mathbf{F}_q$ and upper-case characters denote polynomials of X over $\mathbf{F}_q$.

3.1 Classification of input divisors

This section shows the necessary conditions for the input divisors of the proposed algorithm and the method used to classify the input divisors.

In the addition $D_3 = (U_3, V_3) = D_1 + D_2$ of reduced divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$,

$$\begin{aligned} &\deg U_1 = \deg U_2 = \deg U_3 = 2, \\ &\gcd(U_1, U_2) = 1 \end{aligned}$$

almost always holds for $\mathbf{F}_q$ sufficiently

large (for example, 80 bits) for use in cryptosystems.

Thus, the input divisors are first classified according to whether or not this condition is satisfied. Specifically, the input divisors are first classified by the degrees of U_1 and U_2, and then the resultant of U_1 and U_2 is used to classify the input divisors with

$$\mathrm{res}(U_1, U_2) \neq 0 \Leftrightarrow \gcd(U_1, U_2) = 1\,. \tag{9}$$

Section **3.2** describes the computation procedure for the case in which the input and output divisors satisfy the above conditions.

In the doubling $D_2 = (U_2, V_2) = 2D_1$ of reduced divisors $D_1 = (U_1, V_1)$,

$$\deg U_1 = \deg U_2 = 2,$$
$$\gcd(U_1, 2V_1 + H) = 1$$

almost always holds for $\mathbf{F}_q$ sufficiently large for use in cryptosystems. The procedure is similar to the addition procedure: the input divisors are classified first by the degrees of U_1 and U_2, then the resultant of U_1 and $2V_1 + H$ is used to classify the input divisors with

$$\mathrm{res}(U_1, 2V_1 + H) \neq 0 \Leftrightarrow \gcd(U_1, 2V_1 + H) = 1. \tag{10}$$

Section **3.3** describes the computation procedure for the case in which the input and output divisors satisfy the above conditions.

When the input divisors for addition/doubling do not satisfy the above conditions, the procedures described in Sections **3.2** and **3.3** cannot be applied. In such a case, further detailed classification is performed and a different procedure is used for each class. In practical terms, the classification procedures for addition are the same as those presented in [3], and the classification procedures for doubling are the same as those presented in [8]. The procedures themselves can be obtained easily by modifying the procedures described in [3] for addition and [8] for doubling according to Sections **3.2** and **3.3**.

3.2 Addition algorithm

This section shows the computation procedure of the addition $D_3 = D_1 + D_2 = (U_3, V_3)$ for reduced divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$ with $\deg U_1 = \deg U_2 = 2$ and $\gcd(U_1, U_2) = 1$.

Similar to the Harley algorithm over odd-characteristic fields, the Harley algorithm for genus two hyperelliptic curves over $\mathbf{F}_q$ of arbitrary characteristics consists of a composition part and a reduction part.

First, the composition part establishes D_1 and D_2. Specifically, it computes the semi-reduced divisor, $D = (U, V)$ which is linearly equivalent to $-D_3$ and satisfies

$$U = U_1 U_2\,.$$

V is obtained from

$$\begin{aligned} V &\equiv V_1 \bmod U_1, \\ V &\equiv V_2 \bmod U_2 \end{aligned}$$

using Chinese reminder theorem as

$$\begin{aligned} V &= SU_1 + V_1, \\ S &\equiv (V_2 - V_1)U_1^{-1} \bmod U_2,\ \deg S \leq 1\,. \end{aligned} \tag{11}$$

Next, the reduction part computes the reduced divisor $D'_3 = (U'_3, V'_3)$ for which $D'_3 \sim D$ holds. U'_3 is a quadratic polynomial whose roots are the X coordinates of two among the six cross points (including multiplicity) of C and V, which are not the roots of U. Specifically, using the procedures shown in [4] and [10], we arrive at the following calculation:

$$U'_3 = s_1^{-2}\frac{F - HV - V^2}{U} \tag{12}$$

(Probability of $s_1 = 0$ is very small for $\mathbf{F}_q$ values sufficiently large for use in cryptosystems. However, if this does occur, another algorithm is required. This alternate algorithm is easily obtained by briefly modifying the procedures shown in [8], and thus is omitted here. The output divisor $D_3 = (U_3, V_3)$ in this case, satisfies $\deg U_3 = 1$.) V'_3 is the only polynomial that satisfies (7) for U'_3. Specifically, this polynomial is calculated from

$$V'_3 \equiv V \bmod U'_3$$

and (11) as

$$V'_3 = S(U'_3 - U_1) - s_1(u'_{31} - u_{11})U'_3 + V_1\,.$$

Finally, the output divisor is obtained from (8) as

Table 1 Addition algorithm

Input	A genus 2 hyperelliptic curve $C : Y^2 + H(X)Y = F(X)$, Reduced divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$ such that $\gcd(U_1, U_2) = 1$ and $\deg U_1 = \deg U_2 = 2$	
Output	The reduced divisor $D_3 = (U_3, V_3) = D_1 + D_2$ such that $\deg D_3 = 2$	
Step	Procedure	Cost
1	Compute the resultant r of U_1 and U_2. $w_4 \leftarrow u_{10} - u_{20}$; $i_1 \leftarrow u_{11} - u_{21}$; $i_0 \leftarrow u_{21} i_1 - w_4$; $r \leftarrow u_{20} i_1^2 - w_4 i_0$;	$4M$
2	If $r = 0$ then call the other procedure.	—
3	Compute $I = i_1 X + i_0 \equiv r(U_1)^{-1} \bmod U_2$.	—
4	Compute $T = t_1 X + t_0 \equiv I(V_2 - V_1) \bmod U_2$. $w_0 \leftarrow v_{20} - v_{10}$; $w_1 \leftarrow v_{21} - v_{11}$; $t_2 \leftarrow w_1 i_1$; $t_0 \leftarrow w_0 i_0$; $t_1 \leftarrow (w_0 + w_1)(i_1 + i_0) - t_0 - t_2(u_{21} + 1)$; $t_0 \leftarrow t_0 - t_2 u_{20}$;	$5M$
5	If $t_1 = 0$ then call the other procedure.	—
6	Compute $S = s_1 X + s_0$. $w_0 \leftarrow (r t_1)^{-1}$; $w_1 \leftarrow w_0 r$; $w_2 \leftarrow w_0 t_1$; $w_3 \leftarrow w_1 r$; $s_1 \leftarrow w_2 t_1$; $s_0 \leftarrow w_2 t_0$;	$I + 6M$
7	Compute $U_3 = X^2 + u_{31} X + u_{30} = s_1^{-2}((SU_1 + V_1)^2 + H(SU_1 + V_1) - F)/(U_1 U_2)$. $w_0 \leftarrow w_3^2$; $w_1 \leftarrow s_0 w_3$; $u_{31} \leftarrow i_1 - w_0 + 2w_1 + w_3$; $u_{30} \leftarrow (s_0^2 + s_0 + u_{11} + u_{21} - f_4) w_0 - (u_{21} - 2v_{11} - h_1) w_3$ $+(2w_1 - u_{21}) i_1 + w_4$	$6M$
8	Compute $V_3 = v_{31} X + v_{30}$. $w_1 \leftarrow u_{11} - u_{31}$; $w_0 \leftarrow u_{10} - u_{30}$; $w_2 \leftarrow s_1 w_1$; $w_3 \leftarrow s_0 w_0$; $w_4 \leftarrow (s_1 + s_0)(w_1 + w_0) - w_2 - w_3$; $v_{31} \leftarrow u_{31}(h_2 + w_2) - w_4 - v_{11} - h_1$; $v_{30} \leftarrow u_{30}(h_2 + w_2) - w_3 - v_{10} - h_0$;	$5M$
Total		$I + 26M$

$$D_3 = (U_3, V_3) = (U_3', U_3' - V_3' - H).$$

The detail follows [4] [8], and [10], and the operations on the definition field are optimized. Specifically, (11) is not calculated but substituted in (12) to obtain a polynomial whose coefficients are efficiently combined to reduce the number of operations. The Karatsuba multiplication is also used to reduce the number of multiplication operations over the definition field. Further, similar to [8] and [10], the Montgomery multiple inversion technique is used. This technique converts two inversions in the algorithm into four multiplications and one inversion to reduce the number of inversions.

Consequently, an algorithm is obtained that can be used to calculate $D_3 = D_1 + D_2$ with a computational cost of $I + 26M$.

Table 1 shows the details of the addition algorithm described here and the computational cost for each step.

3.3 Doubling algorithm

This section shows the computation procedure of the doubling $D_2 = (U_2, V_2) = 2D_1$ for reduced divisors $D_1 = (U_1, V_1)$ that satisfy $\deg U_1 = 2$ and $\gcd(U_1, 2V_1 + H) = 1$.

Similar to the addition algorithm, the doubling algorithm consists of a composition part and a reduction part.

First, the composition part computes the semi-reduced divisor $D = (U, V)$ which is equivalent to $-D_2$ and satisfies

$$U = U_1^2 .$$

V is obtained from

$$V \equiv V_1 \bmod U_1$$

using the Newton iteration as

$$V = SU_1 + V_1,$$
$$S \equiv \frac{F - HV_1 - V_1^2}{U_1}(2V_1 + H)^{-1} \bmod U_2, \ \deg S \leq 1 .$$

Next, the reduction part computes the output divisor $D_2 = (U_2, V_2)$ from D following the same procedure used in the addition, obtaining

$$U_2 = s_1^{-2}\frac{F - HV - V^2}{U},$$
$$V_2 = U_2(h_2 - s_1(u_{21} - u_{11})) - S(U_2 - U_1) - V_1 - H .$$

Similar to the addition process, the proce-

Table 2 Doubling algorithm

Input	A genus 2 hyperelliptic curve $C : Y^2 + H(X)Y = F(X)$, A reduced divisor $D_1 = (U_1, V_1)$ such that $\gcd(U_1, 2V_1 + H) = 1$ and $\deg U_1 = 2$	
Output	The reduced divisor $D_2 = (U_2, V_2) = 2D_1$ such that $\deg U_2 = 2$	
Step	Procedure	Cost
1	Compute the resultant r of U_1 and $2V_1 + H$. $w_0 \leftarrow 2v_{10} + h_0$; $w_1 \leftarrow 2v_{11} + h_1$; $i_1 \leftarrow u_{11} - w_1$; $i_0 \leftarrow u_{11}i_1 - u_{10} + w_0$; $r \leftarrow i_0 w_0 + u_{10}(u_{10} - w_0 - i_1 w_1)$	$4M$
2	If $r = 0$ then call the other procedure.	—
3	Compute $I = i_1 X + i_0 \equiv r(2V_1 + H)^{-1} \bmod U_1$.	—
4	Compute $T = t_1 X + t_0 \equiv I(F - HV_1 - V_1^2)/U_1 \bmod U_1$. $w_0 \leftarrow v_{11}(h_1 + v_{11})$; $w_1 \leftarrow 2u_{10}f_4$; $w_2 \leftarrow u_{11}(6u_{10} + 2v_{11} - 2f_3 + u_{11}(3f_4 - 4u_{11})) + f_2 - v_{10} - w_0 - w_1$; $w_3 \leftarrow u_{11}(3u_{11} - 2f_4) - v_{11} - 2u_{10} + f_3$; $t_1 \leftarrow i_1 w_2 + i_0 w_3$; $w_2 \leftarrow u_{10}w_3$; $w_3 \leftarrow f_2 - w_0 - w_1 - v_{10} + u_{11}(v_{11} - f_3 + 4u_{10} + u_{11}(f_4 - u_{11}))$; $t_0 \leftarrow i_0 w_3 - i_1 w_2$;	$12M$
5	If $t_1 = 0$ then call the other procedure.	—
6	Compute $S = s_1 X + s_0$. $w_0 \leftarrow (rt_1)^{-1}$; $w_1 \leftarrow w_0 r$; $w_2 \leftarrow w_0 t_1$; $w_3 \leftarrow w_1 r$; $s_1 \leftarrow w_2 t_1$; $s_0 \leftarrow w_2 t_0$;	$I + 6M$
7	Compute $U_2 = X^2 + u_{21}X + u_{20} = s_1^{-2}((SU_1 + V_1)^2 + H(SU_1 + V_1) - F)/U_1^2$. $u_{21} \leftarrow w_3(2s_0 + 1 - w_3)$; $u_{20} \leftarrow w_3(w_3(2u_{11} - f_4 + s_0(s_0 + 1)) - u_{11} + 2v_{11} + h_1)$;	$4M$
8	Compute $V_2 = v_{21}X + v_{20}$. $w_1 \leftarrow u_{11} - u_{21}$; $w_0 \leftarrow u_{10} - u_{20}$; $w_2 \leftarrow s_1 w_1$; $w_3 \leftarrow s_0 w_0$; $w_4 \leftarrow (s_1 + s_0)(w_1 + w_0) - w_2 - w_3$; $v_{21} \leftarrow u_{21}(1 + w_2) - w_4 - v_{11} - h_1$; $v_{20} \leftarrow u_{20}(1 + w_2) - w_3 - v_{10} - h_0$;	$5M$
Total		$I + 31M$

dures shown in [4] [8], and [10] are used in actual calculation to reduce the number of operations over the definition field.

Consequently, an algorithm is obtained enabling calculation of $D_2 = 2D_1$ with a computational cost of $I + 31M$.

Table 2 shows the details of the doubling algorithm described here and the computational cost for each step.

4 Conclusions

In order to improve the research and our understanding of addition algorithm for hyperelliptic curves, this paper extends the Harley algorithm for hyperelliptic curves over finite field $\mathbf{F}_q$ of arbitrary characteristic. The proposed algorithm can execute addition and doubling, for the divisor classes of genus two hyperelliptic curves over $\mathbf{F}_q$, at a computational cost of $I + 26M$ and $I + 31M$, respectively, offering sufficient efficiency for implementation within symbolic computation software.

Reference

1 D. G. Cantor, "Computing in the Jacobian of hyperelliptic curve", Math. Comp., Vol. 48, No. 177, pp. 95-101, 1987.

2 P. Gaudry and R. Harley, "Counting points on hyperelliptic curves over finite fields", In W. Bosma, editor, ANTS-IV, No. 1838 in Lecture Notes in Computer Science, pp. 313-332. Springer-Verlag, 2000.

3 R. Harley, adding.text, doubling.c. http://cristal.inria.fr/~harley/hyper/, 2000.

4 K. Matsuo, J. Chao, and S. Tsujii, "Fast genus two hyperelliptic curve cryptosystems", Technical Report ISEC2001-31, IEICE Japan, 2001.

5 K. Matsuo, S. Arita, and J. Chao, "A survey on algebraic curve cryptosystems", Transactions of the Japan Sciety for Industrial and Applied Mathematics, Vol. 13, No. 2, pp. 231-243, 2003. (in Japanese).

6 K. Matsuo, S. Arita, and J. Chao, "Public-key cryptosystems on algebraic curves", IPSJ Magazine, Vol. 45, No. 11, pp. 1114-1116, 2004. (in Japanese).

7 M. Gonda, K. Matsuo, K. Aoki, J. Chao, and S. Tsujii, "Improvements of addition algorithm on genus 3 hyperelliptic curves and their implementation", IEICE Trans., Vol. E88-A, No. 1, 2005. 89-96.

8 H. Sugizaki, K. Matsuo, J. Chao, and S. Tsujii, "An extension of Harley addition algorithm for hyperelliptic curves over finite fields of characteristic two", Technical Report ISEC2002-9, IEICE Japan, 2002.

9 T. Lange, "Efficient arithmetic on genus 2 hyperelliptic curves over finite fields via explicit Formulae", Cryptology ePrint Archive, Report 2002/121, 2002. http://eprint.iacr.org/.

10 Y. Miyamoto, H. Doi, K. Matsuo, J. Chao, and S. Tsujii, "A fast addition algorithm of genus two hyperelliptic curves", In Proc. of SCIS2002, pp. 497-502, 2002. (in Japanese).

11 H. Sugizaki, K. Matsuo, J. Chao, and S. Tsujii, "A generalized Harley algorithm for genus two hyperelliptic curves", In Proc. of SCIS2003, pp. 917-921, 2003.

SUGIZAKI Hiroki

Graduate School of Science and Engineering, Chuo University

Hyperelliptic Curve Cryptosystems

***MATSUO Kazuto**, Dr. Eng.*

Professor at Institute of Information Security, Professor of the Research and Development Initiative at Chuo University

Algebraic Curve Cryptography and Computer Cryptology

***CHAO Jinhun**, Ph.D.*

Professor of Department of Information and System Eng. Faculty of Science and Engineering, Chuo University

Elliptic and Hyperelliptic Cryptography

***TSUJII Shigeo**, Ph.D.*

President of Institute of Information Security, Professor of the Research and Development Initiative at Chuo University

Information Security and Cryptology