社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

超楕円曲線上のHarley加算アルゴリズムにおけるresusltant計算について

入海　　淳†　松尾　和人††,†††趙　　晋輝†　辻井　重男††,†††

† 中央大学理工学部情報工学科　〒 112–8551 東京都文京区春日 1–13–27
†† 情報セキュリティ大学院大学　〒 221–0835 横浜市神奈川区鶴屋町 2–14–1
††† 中央大学研究開発機構　〒 112–8551 東京都文京区春日 1–13–27

あらまし　本論文では，超楕円曲線上の高速加算アルゴリズムとして知られる Harley アルゴリズムに必要となる resultant 計算の演算量の少ない計算手順の構成方法を提案する．また，これまでに知られる Harley アルゴリズムに提案構成法を適用しその効果を確認する．本構成法により，種数 3 の超楕円曲線上のこれまでに知られる最少演算量アルゴリズムが得られた．更に，本構成法により得られた種数 3 の超楕円曲線上の Harley アルゴリズムを実装し，Alpha EV68 1.25GHz 上で種数 3 の超楕円曲線上の 160-bit 整数倍算を 163μs で実現した．

キーワード　超楕円曲線暗号，超楕円曲線，加算アルゴリズム， Harley アルゴリズム

On the Resultant Computation in the Harley Algorithms on Hyperelliptic Curves

Jun NYUKAI†, Kazuto MATSUO††,†††, Jinhui CHAO†, and Shigeo TUJII††,†††

† Dept. of Information & System Engineering, Chuo University　Kasuga 1–13–27, Bunkyo-ku, Tokyo, 112–8551 Japan
†† Instutute of Information Security　Tsurumicho 2–14–1, Kanagawa-ku,Yokohama, 221–0835 Japan
††† RDI, Chuo University　Kasuga 1–13–27, Bunkyo-ku, Tokyo, 112–8551 Japan

Abstract　This paper proposes a construction method of the resultant computation for the Harley algorithms, which are known as fast addition algorithms on hyperelliptic cureves, and shows the efficiency of the method to apply it to the computation in the known algorithms. It shows the fastest addition algorithms on genus 3 hyperelliptic curves as the result. Moreover this paper shows a implementation result of the obtained addition algorithms. The result shows that a 160-bit scalar multiplication can be done within 163μs on Alpha EV68 1.25GHz.

Key words　Hyperelliptic curve cryptosystems, Hyperelliptic curves, Addition algorithms, Harley algorithms

1. はじめに

超楕円曲線暗号系には，超楕円曲線上の因子類の高速加算アルゴリズムが必要となる．この因子類の加算アルゴリズムとして，近年 Harley [3], [4] により奇標数の有限体上の種数 2 の超楕円曲線上の高速加算アルゴリズムが提案された．

Harley アルゴリズムを用いた超楕円曲線暗号は楕円曲線暗号と理論上同程度の速度を実現可能なことが知られており [11]，以降 Harley アルゴリズムについて多くの研究がなされてきた [2], [5]〜[10], [12]〜[17] ．その結果，奇標数の有限体上の種数 2 の超楕円曲線のみならず，任意標数の有限体上の種数 4 以下の超楕円曲線上で Harley アルゴリズムが利用可能となった．また, [3], [4] に示された Mumford 表現，中国人剰余定理，Newton 反復，Karatsuba 乗算の利用以外にも，射影 Mumford 表現，Montgomery 同時逆元計算，Toom 乗算，仮想多項式乗算等の手法を利用したアルゴリズム効率化のための一般的な手法が知られるようになり，提案当初のアルゴリズムと比較し，より高速なアルゴリズムが得られている．

Harley アルゴリズムでは，最初に入力因子の詳細な分類を行い，各ケース毎に個別のアルゴリズムが用意される．この分類には resultant 計算が必要となるが，これまではアドホックな方法でこの演算量の削減が行われてきた．唯一 [13] は，これを Bezout 行列の行列式計算によって行うことで，種数 3 の超楕円曲線上のアルゴリズムの演算量が削減されることを示した．しかし，この方法に於いても演算量が削減される理由は示されていなかった．

Harley アルゴリズムでは，分類計算の直後に多項式逆元の計算が行われる．この多項式逆元計算には既に計算済みである

resultant 計算中に現れる計算式を再利用可能であることが知られており，これまでに提案されたアルゴリズムに於いてもこの再利用によって演算量の削減が行われていた．最近になって，Guyot, Kaveh, Patankar [2] はこの resultant 計算と多項式逆元計算の各々について詳細に議論し，これらの類似性をより積極的に利用することで種数 3 の超楕円曲線上のアルゴリズムの演算量を削減した．

本論文では，[2] に示された改良を基に上記 resultant 計算と多項式逆元計算の類似性について議論し，Harley アルゴリズムに於ける resultant 計算と多項式逆元計算の構成方法を提案する．また，提案構成方法により得られた resultant 計算と多項式逆元計算計算をこれまでに知られる Harley アルゴリズムに適用した結果を示す．更に，これにより得られた種数 3 の超楕円曲線上の Harley アルゴリズムの改良アルゴリズムの実装結果を示す．

2. 超楕円曲線とその因子類群

p を素数，n を自然数とし，$q = p^n$ とする．このとき，有限体 $\mathbb{F}_q$ 上の種数 g の超楕円曲線 C を下式で定義する：

$$\begin{aligned} C:\ & Y^2 + H(X)Y = F(X), \\ & F(X) = X^{2g+1} + f_{2g-1}X^{2g-1} + \cdots + f_0 \in \mathbb{F}_q[X], \\ & H(X) = X^g + h_{g-1}X^{g-1} + \cdots + h_0 \in \mathbb{F}_q[X] \end{aligned}$$

ただし，

$$\{(x,y) \in \overline{\mathbb{F}_q}^2 \mid y^2 + H(x)y - F(x) = 2y + H(x) = H'(x)y - F'(x) = 0\} = \emptyset$$

とする．特に $p \neq 2, 2g+1$ のとき，C を下式で定義可能である：

$$\begin{aligned} C:\ & Y^2 = F(X), \\ & F(X) = X^{2g+1} + f_{2g-1}X^{2g-1} + \cdots + f_0 \in \mathbb{F}_q[X] \end{aligned}$$

このとき $\mathrm{disc}\,(F) \neq 0$ である．

C の $\mathbb{F}_q$ 上の因子類群 $\mathcal{J}_C(\mathbb{F}_q)$ は有限 abel 群となり，$\mathcal{J}_C(\mathbb{F}_q)$ 上で離散対数問題に基づく暗号系を構成可能である．

$\mathcal{J}_C(\mathbb{F}_q)$ の任意の元 $\mathcal{D}$ を多項式の組 $(U,V) \in (\mathbb{F}_q[X])^2$ で表現可能である．ここで，U, V は $P_i = (x_i, y_i) \in C$, $i \in \mathbb{N}$ に対し $\mathcal{D}$ を

$$\mathcal{D} = \sum_{P_i \in C} m_i P_i, m_i \in \mathbb{Z}$$

と書いたときに，

$$\begin{aligned} U &= \prod (X - x_i)^{m_i}, \\ y_i &= V(x_i), \\ \deg V &< \deg U, \\ F - HV - V^2 &\equiv 0 \mod U \end{aligned}$$

を満足する多項式である．この $\mathcal{D}$ の表現を Mumford 表現と呼ぶ．

本論文では Mumford 表現を用いて $\mathcal{J}_C(\mathbb{F}_q)$ の元を $\mathcal{D} = (U,V)$ 等と書く．また，$\mathcal{D} = (U,V)$ に対して U の次数を因子の weight と呼ぶ．更に，weight が g 以下の因子を被約因子と呼ぶ．

$\mathcal{J}_C(\mathbb{F}_q)$ の任意の因子類は被約因子により一意表現可能である．

3. Harley アルゴリズム

本章では，本論文で扱う超楕円曲線上の Harley 加算アルゴリズムを概説する．

Harley アルゴリズムは入出力に Mumford 表現で表された被約因子を利用する．以降では，加算に於いては $\mathcal{D}_1 = (U_1, V_1)$, $\mathcal{D}_2 = (U_2, V_2)$ を入力因子，$\mathcal{D}_O = (U_O, V_O) = \mathcal{D}_1 + \mathcal{D}_2$ を出力因子とする．また，2 倍算に於いては $\mathcal{D}_1 = (U_1, V_1)$ を入力因子，$\mathcal{D}_O = (U_O, V_O) = 2\mathcal{D}_1$ を出力因子とする．

アルゴリズムでは，始めに入力因子の詳細な分類を行う．この分類において，加算では入力因子 $\mathcal{D}_1$, $\mathcal{D}_2$ に対し $\deg U_1 = \deg U_2 = g$ かつ $\gcd(U_1, U_2) = 1$ の場合が最も高い確率で起こり，2 倍算では入力因子 $\mathcal{D}_1 = (U_1, V_1)$ に対し $\deg U_1 = g$, $\deg V_1 = g-1$ かつ $\gcd(U_1, V_1) = 1$ の場合が最も高い確率で起こる．入力因子が上記に分類されない確率は $\mathcal{O}(1/q)$ であることが知られている．そこで，本論文では上記に分類される場合のみを扱うこととする．

実際の分類計算で，始めに入力因子の weight が検査され，その後最大公約因子の計算が行われる．この $\mathbb{F}_q$ 上の 2 多項式に対する最大公約因子の計算は，$F_1, F_2 \in \mathbb{F}_q[X]$ に対して

$$\gcd(F_1, F_2) = 1 \iff \mathrm{res}(F_1, F_2) \neq 0$$

が成立することを利用し，実際には resultant 計算によって行われる．

分類計算が終了した後には，まず weight が $2g$ で $-\mathcal{D}_O$ と同値な因子が計算され，次にこの $-\mathcal{D}_O$ から出力因子 $\mathcal{D}_O$ が計算される．

この $-\mathcal{D}_O$ と同値な因子と出力因子 $\mathcal{D}_O$ の計算には，これまでの研究により，中国人剰余定理，Newton 反復，Karatsuba/Toom 乗算，Montgomery 同時逆元計算，仮想多項式乗算等，演算量削減のための多くの一般的手法が知られるようになった．しかし，分類計算に必要な resultant 計算においては，上述のような一般的な演算量削減手法は知られていなかった．唯一種数 3 の場合において，[13] は Bezout 行列の行列式計算により resultant 計算を行い演算量の削減を行ったが，この手法に於いても演算量が削減される理由は明確ではなかった．

最近，種数 3 の場合において，この resultant 計算と $-\mathcal{D}_O$ と同値な因子の計算の最初に現れる多項式逆元計算の類似性を利用することにより，これらの合計演算量を削減可能なことが [2] に示された．しかし，その議論は依然としてアドホックなものであり，一般的な手法ではなかった．

4. Resultant 計算と多項式逆元計算の類似性

本章では，奇標数の有限体上の種数 3 の超楕円曲線に対する

2 倍算アルゴリズムを例に採り, [2] に示された resultant 計算と多項式逆元計算の類似性をより明確に示す.

2 倍算アルゴリズムでは，分類計算に於いて U_1 と V_1 の resultant $r = \mathrm{res}(U_1, V_1)$ を計算し，Montgomery 同時逆元計算を利用した場合には，次に $I \equiv rV_1^{-1} \bmod U_1$ の計算が行われる．ここでは，この r と I の計算に於ける類似性について議論する.

以降では，入力因子 $\mathcal{D}_1 = (U_1, V_1)$ に対し，各多項式の j 次係数をそれぞれ u_{1j}, v_{1j} と書く．即ち，$U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$ とする．また，I の j 次係数を I_j と書く．即ち，$I = i_2X^2 + i_1X + i_0$ とする.

Resultant $r = \mathrm{res}(U_1, V_1)$ は，その定義から，Sylvester 行列

$$M := Syl(V_1, U_1) = \begin{pmatrix} v_{12} & 0 & 0 & 1 & 0 \\ v_{11} & v_{12} & 0 & u_{12} & 1 \\ v_{10} & v_{11} & v_{12} & u_{11} & u_{12} \\ 0 & v_{10} & v_{11} & u_{10} & u_{11} \\ 0 & 0 & v_{10} & 0 & u_{10} \end{pmatrix}$$

を用いて

$$r = \det M$$

と計算される.

一方，多項式逆元 $I \equiv rV_1^{-1} \bmod U_1$ は, [2] も含め，これまでは Euclid の互除法を用いて計算されていた．即ち，$r \neq 0$ のとき,

$$U_1A + V_1B = 1, \deg A < \deg V_1, \deg B < \deg U_1 \qquad (1)$$

を満足する多項式 A, B が存在するので，この B を Euclid の互除法を用いて計算すれば,

$$I = rB$$

として，I が得られる．この計算と r の計算の類似性を観るために，以下に示す Cramer の公式を用いる.

定理 1 (Cramer)**.** K を体とする．M を K 上の $n \times n$ 正則行列，x を K 上の n 次元ベクトルとする．このとき,

$$My = x$$

を満足する K 上の n 次元ベクトル $y = (y_1, \cdots, y_n)^T$ の第 j 成分は

$$y_j = \frac{\det M_j}{\det M}$$

で与えられる．ここで，M_j は M の第 j 列を x で置換して得られる行列である.

この Cramer の公式を多項式逆元 I の計算に適用するために，式 (1) を行列を用いて表現する．この表現は，式 (1) の多項式乗算を展開し，各係数毎の関係を観ることで直ちに得られる．即ち,

$$M(b_2, b_1, b_0, a_1, a_0)^T = (0, 0, 0, 0, 1)^T$$

である.

ここで，a_j, b_j は式 (1) に現れる A, B の j 次係数を各々表す．すると,

$$i_{3-j} = r \det M_j / r = \det M_j, \ j = 0, \ldots, 2$$

として，I の計算に Cramer の公式を直ちに適用可能であることが分かる．ここで，M_j は M の第 j 列を $(0, 0, 0, 0, 1)^T$ で置換して得られる行列である.

以上で見たように，resultant r は M の行列式であり，多項式逆元 I の各係数は M の 1 列を入れ換えた行列 M_j の行列式であるので，両者には明確な類似性がある. [2] は，この類似性を陰に利用し演算量の削減を行ったと考えられる.

5. 提 案 手 法

本章では，前章で示した resultant 計算と多項式逆元計算の類似性を利用した，Harley アルゴリズムの改良を示す.

改良手法の要点は以下の 2 点である.

1. Resultant 計算，多項式逆元計算の構成方法
2. Resultant 計算，多項式逆元計算の順序

即ち，本章で示す改良は Harley アルゴリズム中に現れる resultant 計算，多項式逆元計算の一般的な構成方法である．尚, 1. は [2] が実例により示した構成を一般的な構成方法としたものであり，さらに 2. の改良による変更を加えたものである.

以下では，まず 2. 即ち resultant 計算と多項式逆元計算の計算順序について述べ，その後に 1. 即ち resultant 計算の構成方法，多項式逆元計算の構成方法を述べる．また，これまでに提案された Harley アルゴリズムにこの構成方法を適用し得られたアルゴリズムの演算量を示す.

5.1 計算の順序

これまでに述べたように，Harley アルゴリズムでは，まず分類計算を行い，次に $-\mathcal{D}_O$ と同値な因子の計算が行われる．そこで, [2] の改良を含め従来手法では，Harley アルゴリズムの自然な流れに従い，まず r の計算を行い，次に I の計算を行っていた．また，r の計算過程に現れる値を I の計算に利用し演算量を削減していた．しかし，r と I を個別に計算した場合には I の方が演算量が多くなる．従って，I の演算量削減を優先するために，仮想的に I の計算を先に行いその計算過程に現れる値を r の計算に利用することで，アルゴリズム全体の演算量を削減可能であると考えられる.

そこで，以下では，多項式逆元計算，resultant 計算の順で，計算手順の構成方法とその奇標数の有限体上の種数 3 の超楕円曲線に対する 2 倍算への適用例を示す.

5.2 多項式逆元計算

[2] の方法を基として，Cramer の公式を利用した多項式逆元 I の $g - j$ 次係数の計算手順の提案構成方法を以下に示す.

1: 加算の場合には $M := Syl(U_2 - U_1, U_2)$，2 倍算の場合には $M := Syl(V_1, U_1)$ とする．
2: M の第 j 列を $(0, 0, \ldots, 1)^T$ で置換して M_j を得る．
3: M_j を第 j 列に関し余因子展開し，得られた行列をまた M_j とする．
4: M_j に第 1 要素が 1 の列がある場合，その列を用いた列操作により第 1 要素が 1 以外の列の第 1 要素を 0 にする．
5: M_j を第 1 行に関し余因子展開し，得られた行列をまた M_j とする．
6: M_j の第 1 行の全ての要素が 1 以外になるまで Step 3–4 を繰り返す．

本構成法を種数 2 の場合に適用するときは，Step 1–3 のみを行う．

多項式逆元の高次の係数から計算式を構成することで演算量の少ない式が得られることが知られており，本構成法においても高次の係数から計算式を構成することとする．

以下では，例として奇標数の有限体上の種数 3 の超楕円曲線に対する 2 倍算アルゴリズムに現れる多項式逆元 $I = i_2X^2 + i_1X + i_0$ の計算手順を構成する．

まず，構成方法に従い，I の 2 次係数 i_2 の計算式を導出する．

$$
\begin{aligned}
i_2 = rM_1 &= \begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & v_{12} & 0 & u_{12} & 1 \\ 0 & v_{11} & v_{12} & u_{11} & u_{12} \\ 0 & v_{10} & v_{11} & u_{10} & u_{11} \\ 1 & 0 & v_{10} & 0 & u_{10} \end{vmatrix} \\
&= \begin{vmatrix} 0 & 0 & 1 & 0 \\ v_{12} & 0 & u_{12} & 1 \\ v_{11} & v_{12} & u_{11} & u_{12} \\ v_{10} & v_{11} & u_{10} & u_{11} \end{vmatrix} = \begin{vmatrix} v_{12} & 0 & 1 \\ v_{11} & v_{12} & u_{12} \\ v_{10} & v_{11} & u_{11} \end{vmatrix} \\
&= \begin{vmatrix} 0 & 0 & 1 \\ t_0 & v_{12} & u_{12} \\ t_1 & v_{11} & u_{11} \end{vmatrix} = \begin{vmatrix} t_0 & v_{12} \\ t_1 & v_{11} \end{vmatrix} = v_{11}t_0 - v_{12}t_1.
\end{aligned}
$$

ここで，$t_0 = v_{11} - u_{12}v_{12}$, $t_1 = v_{10} - u_{11}v_{12}$ である．

次に，t_0, t_1 を利用し I の 1 次係数 i_1 の計算式を導出する．

$$
\begin{aligned}
i_1 = rM_2 &= \begin{vmatrix} v_{12} & 0 & 0 & 1 & 0 \\ v_{11} & 0 & 0 & u_{12} & 1 \\ v_{10} & 0 & v_{12} & u_{11} & u_{12} \\ 0 & 0 & v_{11} & u_{10} & u_{11} \\ 0 & 1 & v_{10} & 0 & u_{10} \end{vmatrix} \\
&= -\begin{vmatrix} v_{12} & 0 & 1 & 0 \\ v_{11} & 0 & u_{12} & 1 \\ v_{10} & v_{12} & u_{11} & u_{12} \\ 0 & v_{11} & u_{10} & u_{11} \end{vmatrix} \\
&= -\begin{vmatrix} 0 & 0 & 1 & 0 \\ t_0 & 0 & u_{12} & 1 \\ t_1 & v_{12} & u_{11} & u_{12} \\ -u_{10}v_{12} & v_{11} & u_{10} & u_{11} \end{vmatrix} \\
&= -\begin{vmatrix} t_0 & 0 & 1 \\ t_1 & v_{12} & u_{12} \\ -u_{10}v_{12} & v_{11} & u_{11} \end{vmatrix} = -\begin{vmatrix} 0 & 0 & 1 \\ t_2 & v_{12} & u_{12} \\ -t_3 & v_{11} & u_{11} \end{vmatrix} \\
&= -\begin{vmatrix} t_2 & v_{12} \\ -t_3 & v_{11} \end{vmatrix} = -(v_{11}t_2 + v_{12}t_3).
\end{aligned}
$$

ここで，$t_2 = t_1 - u_{12}t_0$, $t_3 = u_{10}v_{12} + u_{11}t_0$ である．

最後に，t_0, t_1, t_2, t_3 を利用し I の定数項 i_0 の計算式を導出する．

$$
\begin{aligned}
i_0 = rM_3 &= \begin{vmatrix} v_{12} & 0 & 0 & 1 & 0 \\ v_{11} & v_{12} & 0 & u_{12} & 1 \\ v_{10} & v_{11} & 0 & u_{11} & u_{12} \\ 0 & v_{10} & 0 & u_{10} & u_{11} \\ 0 & 0 & 1 & 0 & u_{10} \end{vmatrix} \\
&= \begin{vmatrix} v_{12} & 0 & 1 & 0 \\ v_{11} & v_{12} & u_{12} & 1 \\ v_{10} & v_{11} & u_{11} & u_{12} \\ 0 & v_{10} & u_{10} & u_{11} \end{vmatrix} = \begin{vmatrix} 0 & 0 & 1 & 0 \\ t_0 & v_{12} & u_{12} & 1 \\ t_1 & v_{11} & u_{11} & u_{12} \\ -u_{11}v_{12} & v_{10} & u_{10} & u_{11} \end{vmatrix} \\
&= \begin{vmatrix} t_0 & v_{12} & 1 \\ t_1 & v_{11} & u_{12} \\ -u_{11}v_{12} & v_{10} & u_{11} \end{vmatrix} = \begin{vmatrix} 0 & 0 & 1 \\ t_2 & t_0 & u_{12} \\ -t_3 & t_1 & u_{11} \end{vmatrix} \\
&= \begin{vmatrix} t_2 & t_0 \\ -t_3 & t_1 \end{vmatrix} = t_1t_2 + t_0t_3.
\end{aligned}
$$

以上により，奇標数の有限体 $\mathbb{F}_q$ 上の種数 3 の超楕円曲線に対する 2 倍算アルゴリズムに現れる多項式逆元 $I = i_2X^2 + i_1X + i_0$ を $\mathbb{F}_q$ 上の 11 回の乗算と 8 回の加算で実現する計算手順が得られた．（実際に用いられる $\mathbb{F}_q$ の性質を考慮し，加法逆元の計算も加算回数に加えている．）

5.3 Resultant 計算

以下に [2] の方法を基にした resultant r の計算手順の提案構成方法を示す．

1: 加算の場合には $M := Syl(U_2 - U_1, U_2)$，2 倍算の場合には $M := Syl(V_1, U_1)$ とする．
2: M に第 1 要素が 1 の列がある場合，その列を用いた列操作により第 1 要素が 1 以外の列の第 1 要素を 0 にする．
3: M を第 1 行で余因子展開し，得られた行列をまた M とする．
4: M の第 1 行のすべての要素が 1 以外になるまで Step 2–3 を繰り返す．
5: M を最下行に関し余因子展開する．

本構成法を種数 2 の場合に適用するときは，Step 1–2 の後，Step 5 を行う．

以下では，例として奇標数の有限体上の種数 3 の超楕円曲線に対する 2 倍算アルゴリズムに現れる resultant r の計算式を構成する．

構成に於いては多項式逆元計算で得られた $t_0 = v_{11} - u_{12}v_{12}$, $t_1 = v_{10} - u_{11}v_{12}$, $t_2 = t_1 - u_{12}t_0$, $t_3 = u_{11}v_{12} + u_{11}t_0$ を利用する．

$$
\begin{aligned}
r \;=\; \det M &= \begin{vmatrix} v_{12} & 0 & 0 & 1 & 0 \\ v_{11} & v_2 & 0 & u_{12} & 1 \\ v_{10} & v_1 & v_2 & u_{11} & u_{12} \\ 0 & v_{10} & v_1 & u_{10} & u_{11} \\ 0 & 0 & v_0 & 0 & u_{10} \end{vmatrix} \\
&= \begin{vmatrix} 0 & 0 & 0 & 1 & 0 \\ v_{11} - u_{12}v_{12} & v_{12} & 0 & u_{12} & 1 \\ v_{10} - u_{11}v_{12} & v_{11} & v_{12} & u_{11} & u_{12} \\ -u_{10}v_{12} & v_0 & v_1 & u_{10} & u_{11} \\ 0 & 0 & v_{10} & 0 & u_{10} \end{vmatrix} \\
&= -\begin{vmatrix} t_0 & 0 & 0 & 1 \\ t_1 & t_0 & v_{12} & u_{22} \\ -u_{10}v_{12} & t_1 & v_{11} & u_{11} \\ 0 & -u_{10}v_{12} & v_0 & u_{10} \end{vmatrix} \\
&= -\begin{vmatrix} 0 & 0 & 0 & 1 \\ t_1 - u_{12}t_0 & t_0 & v_{12} & u_{12} \\ -u_{10}v_{12} - u_{11}t_0 & t_1 & v_{11} & u_{11} \\ -u_{10}t_0 & -u_{10}v_{12} & v_{10} & u_{10} \end{vmatrix} \\
&= \begin{vmatrix} t_2 & t_1 & v_{12} \\ -t_1 & t_0 & v_{11} \\ -u_{10}t_0 & -u_{10}v_{10} & v_0 \end{vmatrix} \\
&= t_2 \begin{vmatrix} t_1 & v_{11} \\ -u_{10}v_{12} & v_{10} \end{vmatrix} - t_1 \begin{vmatrix} t_0 & v_{12} \\ -u_{10}v_{12} & v_{10} \end{vmatrix} - u_{10}t_0 \begin{vmatrix} t_0 & v_{12} \\ t_1 & v_{11} \end{vmatrix} \\
&= t_2 i_2 - t_1 i_1 - u_{10}t_0 i_0.
\end{aligned}
$$

上式より，奇標数の有限体 $\mathbb{F}_q$ 上の種数 3 の超楕円曲線に対する 2 倍算アルゴリズムに現れる resultant r は $\mathbb{F}_q$ 上の 4 回の乗算と 2 回の加算で計算可能である．また，2 倍算アルゴリズムに現れる resultant r が，多項式逆元計算中に現れる計算式を用いて効率的に計算されることが判る．

以上の構成によって，奇標数の有限体 $\mathbb{F}_q$ 上の種数 3 の超楕円曲線に対する 2 倍算アルゴリズムに現れる resultant と多項式逆元を $\mathbb{F}_q$ 上の 15 回の乗算と 10 回の加算で計算可能である．また，同様の構成により加算アルゴリズムに現れる resultant と多項式逆元を $\mathbb{F}_q$ 上の 15 回の乗算と 13 回の加算で計算可能である．一方, [2] はこれらを 2 倍算に対して 16 回の乗算と 11 回の加算，加算に対して 16 回の乗算と 14 回の加算で実現している．

5.4 既存アルゴリズムへの適用結果

これまでに提案されている Harley アルゴリズムに提案構成法を適用した．提案構成法を適用した結果，演算量が削減されたアルゴリズムに関し，加算アルゴリズムの演算量比較を表 3 に，2 倍算アルゴリズムの演算量比較を表 4 に示す．表 3, 4 では $\mathbb{F}_q$ 上の元 a, b に対する $a+b$, $1/a$, ab, a^2 の計算に必要な演算量を，各々A, I, M, S で表す．また，演算量の評価方法は各々の論文に記載された方法に従った．

表 3　The result on applying the proposed method to the previous addition algorithms

Genus	char($\mathbb{F}_q$)	Previous work	This work
3	even	$I + 64M + 4S$ [2]	$\mathbf{I + 63M + 4S}$
	odd	$I + 70M$ [2]	$I + 69M$
		$I + 70M + 113A$ [1]	$\mathbf{I + 67M + 110A}$
		$I + 72M + 111A$ [1]	$I + 69M + 108A$
		$I + 79M + 83A$ [1]	$I + 76M + 80A$
4	general	$2I + 160M + 4S$ [14]	$2I + 154M + 4S$

表 4　The result on applying the proposed method to the previous doubling algorithms

Genus	char($\mathbb{F}_q$)	Previous work	This work
3	even	$I + 64M + 5S$ [2]	$\mathbf{I + 63M + 5S}$
	odd	$I + 70M$ [2]	$I + 69M$
	odd	$I + 71M + 107A$ [1]	$\mathbf{I + 68M + 104A}$
	odd	$I + 73M + 101A$ [1]	$I + 70M + 98A$
	odd	$I + 78M + 83A$ [1]	$I + 75M + 80A$
4	general	$2I + 193M + 16S$ [14]	$2I + 188M + 16S$

表 3 に示された，標数 2 の有限体上の種数 3 の超楕円曲線に対する加算演算量 $I + 63M + 4S$，奇標数の有限体上の種数 3 の超楕円曲線に対する加算演算量 $I + 67M + 110A$ と，表 4 に示された，標数 2 の有限体上の種数 3 の超楕円曲線に対する 2 倍算演算量 $I + 63M + 5S$，奇標数の有限体上の種数 3 の超楕円曲線に対する 2 倍算演算量 $I + 68M + 104A$ は，$M + S$ を演算量評価パラメータとしたときの，各々の場合のこれまでに知られる最少演算量である．

奇標数の有限体上の種数 3 の超楕円曲線に対する，演算量が $I + 67M + 110A$ の加算アルゴリズムを表 1 に，演算量が $I + 68M + 104A$ の 2 倍算アルゴリズムを表 2 に示す．

種数 4 の超楕円曲線に対しては, [14] よりも効率的なアルゴリズムが [6] に示されており，これに提案構成法を適用することでより効率的なアルゴリズムが構成可能であると考え

表 1　Addition algorithm on genus 3 HEC for the most frequent case

In.	Genus 3 HEC $C: Y^2 = F(X) = X^7 + f_5X^5 + f_4X^4 + f_3x^3 + f_2X^2 + f_1X + f_0$, Reduced divisors $\mathcal{D}_1 = (U_1, V_1)$, $\mathcal{D}_2 = (U_2, V_2)$, where $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$, $U_2 = X^3 + u_{22}x^2 + u_{21}X + u_{20}$, and $V_2 = v_{22}X^2 + v_{21}X + v_{20}$	
Out.	Reduced divisor $\mathcal{D}_O = (U_O, V_O) = \mathcal{D}_1 + \mathcal{D}_2$, where $U_O = X^3 + u_{O2}X^2 + u_{O1}X + u_{O0}$, and $V_O = v_{O2}X^2 + v_{O1}X + v_{O0}$	
Step	Procedure	Cost
1	[Compute the resultant r of U_1 and U_2] $t_0 = u_{10} - u_{20}$; $t_1 = u_{11} - u_{21}$; $t_2 = u_{12} - u_{22}$; $t_3 = t_1 - u_{22}t_2$; $t_4 = t_0 - u_{21}t_2$; $t_5 = t_4 - u_{22}t_3$; $t_6 = u_{20}t_2 + u_{21}t_3$; $t_7 = t_4t_5 + t_3t_6$; $t_8 = -(t_2t_6 + t_1t_5)$; $t_9 = t_1t_3 - t_2t_4$; $r = u_{20}(t_3t_9 + t_2t_8) - t_0t_7$;	$15M + 13A$
2	[If $r = 0$ then call the Cantor algorithm]	–
3	[Compute the pseudo-inverse $I = i_2x^2 + i_1x + i_0 \equiv r/U_1 \bmod U_2$] $i_2 = t_9$; $i_1 = t_8$; $i_0 = t_7$;	–
4	[Compute $S' = s'_2x^2 + s'_1x + s'_0 = rS \equiv (V_2 - V_1)I \bmod U_2$ (Karatsuba, Toom)] $t_1 = v_{10} - v_{20}$; $t_2 = v_{11} - v_{21}$; $t_3 = v_{12} - v_{22}$; $t_4 = t_2i_1$; $t_5 = t_1i_0$; $t_6 = t_3i_2$; $t_8 = u_{22}t_6$; $t_8 = t_4 + t_6 + t_7 - (t_2 + t_3)(i_1 + i_2)$; $t_9 = u_{20} + u_{22}$; $t_{10} = (t_9 + u_{21})(t_8 - t_6)$; $t_9 = (t_9 - u_{21})(t_8 + t_6)$; $s'_0 = -(u_{20}t_8 + t_5)$; $s'_2 = t_6 - (s'_0 + t_4 + (t_1 + t_3)(i_0 + i_2) + (t_{10} + t_9)/2)$; $s'_1 = t_4 + t_5 + (t_9 - t_{10})/2 - (t_7 + (t_1 + t_2)(i_0 + i_1))$;	$10M + 31A$
5	[If $s'_2 = 0$ then call the Cantor algorithm]	–
6	[Compute S, w and $w_i = 1/w$ s.t. $wS = S'/r$ and S is monic] $t_1 = (rs'_2)^{-1}$; $t_2 = rt_1$; $w = t_1s'^2_2$; $w_i = rt_2$; $s_0 = t_2s'_0$; $s_1 = t_2s'_1$;	$I + 7M$
7	[Compute $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$] $t_6 = s_0 + s_1$; $t_1 = u_{10} + u_{12}$; $t_2 = t_6(t_1 + u_{11})$; $t_3 = (t_1 - u_{11})(s_0 - s_1)$; $t_4 = u_{12}s_1$; $z_0 = u_{10}s_0$; $z_1 = (t_2 - t_3)/2 - t_4$; $z_2 = (t_2 + t_3)/2 - z_0 + u_{10}$; $z_3 = u_{11} + s_0 + t_4$; $z_4 = u_{12} + s_1$;	$4M + 15A$
8	[Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} = (S(Z + 2w_iV_1) - w_i^2((F - V_1^2)/U_1))$] $t_1 = s_0z_3$; $u_{t3} = z_4 + s_1 - u_{22}$; $t_5 = s_1z_4 - u_{22}u_{t3}$; $u_{t2} = z_3 + s_0 + t_5 - u_{21}$; $t_3 = u_{21}u_{t2}$; $t_4 = t_1 - t_3$; $t_2 = (u_{22} + u_{21})(u_{t3} + u_{t2})$; $u_{t2} = z_3 + s_0 + t_5 - u_{21}$; $u_{t1} = z_2 + t_6(z_4 + z_3) + w_i(2v_{12} - w_i) - (t_5 + t_2 + t_4 + u_{20})$; $u_{t0} = z_1 + t_4 + s_1z_2 + w_i(2(v_{11} + s_1v_{12}) + w_iu_{12}) - (u_{22}u_{t1} + u_{20}u_{t3})$;	$13M + 26A$
9	[Compute $V_t = v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wZ + V_1 \bmod U_t$] $t_1 = u_{t3} - z_4$; $v_{t0} = w(t_1u_{t0} + z_0) + v_{10}$; $v_{t1} = w(t_1u_{t1} + z_1 - u_{t0}) + v_{11}$; $v_{t2} = w(t_1u_{t2} + z_2 - u_{t1}) + v_{12}$; $v_{t3} = w(t_1u_{t3} + z_3 - u_{t2})$	$8M + 11A$
10	[Compute $U_O = X^3 + u_{O2}X^2 + u_{O1}X + u_{O0} = (F - V_t^2)/U_t$] $t_1 = 2v_{t3}$; $u_{O2} = -(u_{t3} + v_{t3}^2)$; $u_{O1} = f_5 - (u_{t2} + u_{O2}u_{t3} + t_1v_{t2})$; $u_{O0} = f_4 - (u_{t1} + v_{t2}^2 + u_{O2}u_{t2} + u_{O1}u_{t3} + t_1v_{t1})$;	$7M + 11A$
11	[Compute $V_O = v_{O2}x^2 + v_{O1}x + v_{O0} \equiv -V_t \bmod U_O$] $v_{O2} = v_{t2} - u_{O2}v_{t3}$; $v_{O1} = v_{t1} - u_{O1}v_{t3}$; $v_{O0} = v_{t0} - u_{O0}v_{t3}$;	$3M + 3A$
Total		$I + 67M + 110A$

表 2　Doubling algorithm on genus 3 HEC for the most frequent case

In.	Genus 3 HEC $C: Y^2 = F(X) = x^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$, Reduced divisor $\mathcal{D}_1 = (U_1, V_1)$, where $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, and $V_1 = v_{12}X^2 + v_{11}X + v_{10}$	
Out.	Reduced divisor $\mathcal{D}_O = (U_O, V_O) = 2\mathcal{D}_1$, where $U_2 = X^3 + u_{O2}X^2 + u_{O1}x + u_{O0}$, and $V_O = v_{O2}X^2 + v_{O1}X + v_{O0}$	
Step	Procedure	Cost
1	[Compute the resultant r of U_1 and V_1] $t_1 = v_{11} - u_{12}v_{12}$; $t_2 = v_{10} - u_{11}v_{12}$; $t_3 = t_2 - u_{12}t_1$; $t_4 = u_{10}v_{12} + u_{11}t_1$; $t_5 = t_2t_3 + t_1t_4$; $t_6 = -(v_{11}t_3 + v_{12}t_4)$; $t_7 = v_{11}t_1 - v_{12}t_2$;$r = v_{10}t_5 - u_{10}(t_1t_7 + v_{12}t_6)$;	15M+10A
2	[If $r = 0$ then call the Cantor Algorithm]	–
3	[Compute the pseudo-inverse $I = i_2x^2 + i_1x + i_0 \equiv r/V_1 \bmod U_1$] $i_2 = t_7$; $i_1 = t_6$; $i_0 = t_5$;	–
4	[Compute $Z = z_2X^2 + z_1X + z_0 \equiv (F - V_1^2)/U_1 \bmod U_1$] $t_1 = 2u_{10}$; $t_2 = 2u_{11}$; $t_3 = u_{12}^2$; $t_4 = f_4 - (t_1 + v_{12}^2)$; $t_5 = f_5 + t_3 - t_2$; $t_{10} = 2v_{12}$; $z_2 = t_5 + 2t_3$; $z_1 = u_{12}(t_2 - t_5) + t_4$; $z_0 = f_3 + t_3(t_5 - u_{11}) + u_{12}(t_1 - t_4) + u_{11}(u_{11} - f_5) - t_{10}v_{11}$;	$7M + 18A$
5	[Compute $S' = s'_2x^2 + s'_1x + s'_0 = 2rS \equiv ZI \bmod U_1$ (Karatsuba, Toom)] $t_1 = i_1z_1$; $t_2 = i_0z_0$; $t_3 = i_2z_2$; $t_4 = u_{12}t_3$; $t_5 = (i_2 + i_1)(z_2 + z_1) - (t_1 + t_3 + t_4)$; $t_6 = u_{10}t_5$; $t_7 = u_{10} + u_{12}$; $t_8 = t_7 + u_{11}$; $t_9 = t_7 - u_{11}$; $t_7 = t_8(t_3 + t_5)$; $t_{11} = t_9(t_5 - t_3)$; $s'_2 = t_1 + t_6 + (i_2 + i_0)(z_2 + z_0) - (t_2 + t_3 + (t_7 + t_{11})/2)$; $s'_1 = t_4 + (i_0 + i_1)(z_1 + z_0) + (t_{11} - t_7)/2 - (t_1 + t_2)$; $s'_0 = t_2 - t_6$;	$10M + 28A$
6	[If $s'_2 = 0$ then call the Cantor algorithm]	–
7	[Compute S, w and $w_i = 1/w$ s.t. $wS = S'/(2r)$ and S is monic] $t_1 = 2r$; $t_2 = (t_1s'_2)^{-1}$; $t_3 = t_1t_2$; $w = t_2s'^2_2$; $w_i = t_1t_3$; $s_0 = t_3s'_0$; $s_1 = t_3s'_1$;	$I + 7M + A$
8	[Compute $G = X^5 + g_4X^4 + g_3X^3 + g_2X^2 + g_1X + g_0 = SU_1$ (Toom)] $t_1 = t_8(s_1 + s_0)$; $t_2 = t_9(s_0 - s_1)$; $t_3 = u_{12}s_1$; $g_0 = u_{10}s_0$; $g_1 = (t_1 - t_2)/2 - t_3$; $g_2 = u_{10} + (t_1 + t_2)/2 - g_0$; $g_3 = t_3 + u_{11} + s_0$; $g_4 = u_{12} + s_1$;	$4M + 12A$
9	[Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} = ((G + w_iV_1)^2 - w_i^2F)/U_1^2$] $u_{t3} = 2s_1$; $u_{t2} = s_1^2 + 2s_0$; $u_{t1} = u_{t3}s_0 + w_i(t_{10} - w_i)$; $u_{t0} = s_0^2 + 2w_i((s_1 - u_{12})v_{12} + v_{11} + w_iu_{12})$;	$7M + 10A$
10	[Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wG + V_1 \bmod U_t$] $t_1 = u_{t3} - g_4$; $v_{t0} = w(t_1u_{t0} + g_0) + v_{10}$; $v_{t1} = w(t_1u_{t1} + g_1 - u_{t0}) + v_{11}$; $v_{t2} = w(t_1u_{t2} + g_2 - u_{t1}) + v_{12}$; $v_{t3} = w(t_1u_{t3} + g_3 - u_{t2})$;	$8M + 11A$
11	[Compute $U_O = X^3 + u_{O2}X^2 + u_{O1}X + u_{O0} = (F - V_t^2)/U_t$] $t_1 = 2v_{t3}$; $u_{O2} = -(u_{t3} + v_{t3}^2)$; $u_{O1} = f_5 - (u_{t2} + u_{O2}u_{t3} + t_1v_{t2})$; $u_{O0} = f_4 - (u_{t1} + v_{t2}^2 + u_{O2}u_{t2} + u_{O1}u_{t3} + t_1v_{t1})$;	$7M + 11A$
12	[Compute $V_O = v_{O2}x^2 + v_{O1}x + v_{O0} \equiv V_t \bmod U_O$ (Karatsuba)] $v_{O2} = v_{t2} - u_{O2}v_{t3}$; $v_{O1} = v_{t1} - u_{O1}v_{t3}$; $v_{O0} = v_{t0} - u_{O0}v_{t3}$;	$3M + 3A$
Total		$I + 68M + 104A$

られる．また，提案構成法を用いて構成した resultant 計算の [8], [10], [12], [15] に記載された種数 2 の超楕円曲線に対するアルゴリズムへの適用も試みたが，大きな効果は観られなかった．以上より，提案方法は resultant 計算により多くの演算量を必要とする高種数の超楕円曲線に対して効果のある方法であると考えられる．

6. 実装結果

前章で得られた [1] に示された実装と同一の環境で [1] の改良アルゴリズムを実装した．即ち，$p = 2^{61} - 1$ とし，$\mathbb{F}_p$ 上の種数 3 の超楕円曲線上の Harley アルゴリズムを Alpha EV68 上に実装した．有限体上の演算には, [1] で用いられた関数群の乗算と逆元計算に改良を施したものを利用した．

実装結果を表 5 に示す．また，比較のため [1] に示された実装結果を併せて示す．

表 5 中，“Toom” は [1] に示された $I+70M+113A$ の加算アルゴリズム，$I+71M+107A$ の 2 倍算アルゴリズム，本実装に用いた $I+67M+110A$ の加算アルゴリズム，$I+68M+104A$

表 5　Performance results on Alpha EV68 1.25GHz

	Addition	Doubling	Scalar mul.
[This work]			
Toom	$862ns$	$865ns$	$171\mu s$
Karatsuba	$880ns$	$863ns$	$171\mu s$
Classical	$822ns$	$832ns$	$163\mu s$
[Previous work [1]]			
Toom	$919ns$	$916ns$	$180\mu s$
Karatsuba	$920ns$	$897ns$	$177\mu s$
Classical	$888ns$	$875ns$	$172\mu s$

の 2 倍算アルゴリズムを，“Karatsuba” は [1] に示された $I+72M+111A$ の加算アルゴリズム，$I+73M+101A$ の 2 倍算アルゴリズム，本実装に用いた $I+69M+108A$ の加算アルゴリズム，$I+70M+98A$ の 2 倍算アルゴリズムをそれぞれ表し，また “Classical” は [1] に示された $I+79M+83A$ の加算アルゴリズム，$I+78M+83A$ の 2 倍算アルゴリズム，本実装に用いた $I+76M+80A$ の加算アルゴリズム，$I+75M+80A$ の 2 倍算アルゴリズムをそれぞれ表す．また，“Addition,” “Doubling,” “Scalar mul.” は，それぞれ加算，2 倍算，160-bit 整数倍算の計時結果を表す．

本実装が [1] に示された実装と比較し 5%程度の高速化を実現していることが表 5 から判る．

謝　　辞

本研究の一部は文部科学省の 21 世紀 COE プログラム「電子社会の信頼性向上と情報セキュリティ」プロジェクトによる援助を受けた．

文　　献

[1] 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, January 2005. 89-96.

[2] C. Guyot, K. Kaveh, and V. M. Patankar. Explicit algorithm for the arithmetic on the hyperelliptic Jacobians of genus 3. *Journal of the Ramanujan Math. Soc.*, Vol. 19, No. 2, pp. 75–115, June 2004.

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

[4] R. Harley. Counting points with the arithmetic-geometric mean. Rump talk at EUROCRYPT 2001, 2001. (joint work with J.-F. Mestre and P. Gaudry).

[5] M. Katagi, T. Akishita, I. Kitamura, and T. Takagi. Efficient hyperelliptic curve cryptosystems using Theta divisors. *IEICE Japan Trans. Fundamentals*, Vol. E89-A, No. 1, pp. 151–160, 2006.

[6] Y. Kitamura, M. Kawazoe, and T. Takahashi. Improvement of explicit addition and doubling formulae for genus-4 HECC. In *Proc. of SCIS2006*, 2006. 4C1-1.

[7] J. Kuroki, M. Gonda, K. Matsuo, J. Chao, and S. Tsujii. Fast genus three hyperelliptic curve cryptosystems. In *Proc. of SCIS2002*, pp. 503–507, January 2002.

[8] 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/`.

[9] T. Lange. Inversion-free arithmetic on genus 2 hyperelliptic curves. Cryptology ePrint Archive, Report 2002/147, 2002. `http://eprint.iacr.org/`.

[10] T. Lange. Weighted coordinates on genus 2 hyperelliptic curves. Cryptology ePrint Archive, Report 2002/153, 2002. `http://eprint.iacr.org/`.

[11] K. Matsuo, J. Chao, and S. Tsujii. Fast genus two hyperelliptic curve cryptosystems. Technical Report ISEC2001-31, IEICE Japan, July 2001.

[12] 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, January 2002. in Japanese.

[13] J. Pelzl, T. Wollinger, J. Guajardo, and C. Paar. Hyperelliptic curve cryptosystems: Closing the performance gap to elliptic curves. In *Cryptographic Hardware and Embedded Systems - CHES 2003*, No. 2779 in Lecture Notes in Computer Science, pp. 351–365. Springer-Verlag, 2003.

[14] J. Pelzl, T. Wollinger, and C. Paar. Low cost security: Explicit formulae for genus 4 hyperelliptic curves. In *Proc. of 10th Annual International Workshop on Selected Areas in Cryptography (SAC2003), Carleton University, Ottawa*, No. 3006 in Lecture Notes in Computer Science. Springer-Verlag, 2003.

[15] 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, May 2002.

[16] M. Takahashi. Improving Harley algorithms for Jacobians of genus 2 hyperelliptic curves. In *Proc. of SCIS2002*, pp. 155–160, 2002. in Japanese.

[17] T. Wollinger, J. Pelzl, and C. Paar. Cantor versus harley: Optimization and analysis of explicit formulae for hyperelliptic curve cryptosystems. *IEEE Trans. Computers*, Vol. 54, No. 7, pp. 861–872, 2005.