SCIS2000 The 2000 Symposium on Cryptography and Information Security
Okinawa, Japan, January 26-28, 2000

逐次拡大 OEF

OEF Using a Successive Extension

小林 鉄太郎* 青木 和麻呂 * 星野 文学 *
Tetsutaro Kobayashi Kazumaro Aoki Fumitaka Hoshino

Abstract— We propose a new class of OEF. The OEFs are constructed by using a successive extension. Our analysis shows that our inversion algorithm requires about 1.5 GF(p^m) multiplications and 1 GF(p) inversion, while Bailey and Paar algorithm requires about 3 GF(p^m) multiplications and 1 GF(p) inversion.

Keywords: OEF, successive extension, tower field, elliptic curve cryptosystem, software implementation

1 はじめに

近年、速くて鍵長の短いことから、楕円曲線暗号が注目されている。楕円曲線暗号は楕円曲線上の群演算を用いるが、これは有限体上の演算に帰着される。楕円曲線上の点の表現には、大きく分けてアフィン座標表現と、射影座標表現の 2 つがある。アフィン座標表現は有限体上の逆元を必要とし、射影座標表現は必要としない代わりに多くの乗算を必要とする、という違いがある [3]。したがって、逆元が高速に実装できれば、アフィン座標表現は射影座標表現よりも優れていることになる。

CRYPTO'98 において、Bailey と Paar がソフトウェアによる実装に適し、従来法よりもひと桁程度高速となる OEF (optimal extension field) を提案した [1]。さらに、1999 年 7 月の Conference on The Mathematics of Public-Key Cryptography において、OEF の逆元アルゴリズムを提案している [2]。このアルゴリズムは、正規基底を用いた拡大体上の伊東 - 辻井アルゴリズム [5] を応用したものである。伊東 - 辻井アルゴリズムは、Frobenius 写像を高速に求められる場合に、有限体上の乗算を用いて逆元を行なう方法である。Bailey らの OEF は多項式基底を用いており、正規基底とくらべて Frobenius 写像を行なうのにすこし時間がかかるが、それでも有限体上の逆元を効率良く求めることが出来る。このアルゴリズムを用いれば、楕円曲線上の加算はアフィン座標の方が射影座標より高速に求めることが出来る。

一方、2 の拡大体が逐次拡大によって構成されている場合には、超高速な逆元アルゴリズムがあり、たとえば、Fan らが [4] で発表している。我々は、このアイデアを OEF に適用する。

新しい OEF のクラスを提案する。この OEF は、逐次拡大によって構成される。この方法では、逆元を 1 回の GF(p) 逆元と約 1.5 回の GF(p^m) 乗算と同等の計算量行なうことが出来る。1 回の GF(p) 逆元を $m \geq 3$ の場合に、最低 3 回の GF(p^m) 乗算と必要とする、Bailey らの方式と比べて、より実用的である。

2 従来法

2.1 OEF

Bailey らによって提案された OEF[1] は、以下の条件を見たす有限体 GF(p^m) である。

- $n = \log_2 p$ は、計算機の処理するワード長以下でそれに近い値、
- $p = 2^n \pm c$, ただし $\log_2 c \leq n/2$、
- 既約二項式 $f(x) = x^m - \omega$ が存在する。

OEF 上の演算は、計算機の加減乗除命令を有効に用いて実装することが出来るため、ソフトウェア実装に適している。

2.2 Bailey's Method

Bailey らによって提案された、Frobenius Map を用いる逆元演算法を示す。これは、伊東 - 辻井アルゴリズムの応用である。

* NTT 情報流通プラットフォーム研究所, 〒 239-0847 神奈川県横須賀市光の丘 1-1, NTT Information Sharing Platform Laboratories, 1-1 Hikarinooka Yokosuka-Shi Kanagawa 239-0847 Japan

[Bailey の逆元アルゴリズム]

Input: $A(x) \in \mathrm{GF}(p^m)$
Output: $B(x)$ $(A(x)B(x) \equiv 1 \pmod{f(x)})$

Step 1: $B(x) \longleftarrow A(x)$
Step 2: $B(x)^{r-1}$ を求める。
Step 3: $c_0 \leftarrow B(x)A(x)$
Step 4: $c \leftarrow c_0^{-1}$
Step 5: $B(x) \leftarrow B(x)c$

ただし、 $r = \dfrac{p^m - 1}{p - 1}$ である。Step 2 の $B(x)^{r-1}$ を求める部分は、[5] で示されている、最適な Addition Chain を用いればよい。

3 提案法

3.1 逐次拡大表現

逐次拡大を用いる OEF では、通常の OEF と異なり、既約二項式が存在している必要はないが、拡大次数 m が合成数である必要がある。m が多くの因数を含んでいるほど効率が良い。

ここでは、もっとも効率の良い $m = 2^k$ の場合について説明する。ただし、 $q_i = p^{2^i}, 0 \leq i \leq k$ とする。

$a \in \mathrm{GF}(q_k)$ を表現するのに、(α_k, β_k) を用いて

$$a = a_0\alpha_k + a_1\beta_k$$

と表す。ただし $a_i \in \mathrm{GF}(q_{k-1})$。

同様にして、 $a_i \in \mathrm{GF}(q_j)$ を $a_{i,0}, a_{i,1} \in \mathrm{GF}(q_{i-1})$ を用いて

$$a_i = a_{i,0}\alpha_j + a_{i,1}\beta_j$$

と表す。

これを繰り返すことによって、最終的に $\mathrm{GF}(q_k)$ の元を $\mathrm{GF}(p)$ の元 2^k 個によって表すことができる。

$\alpha = 1$ で、β が $\mathrm{GF}(q_{k-1})$ 上の 2 次 既約多項式 $f(x)$ の根の場合は、多項式基底表現の逐次拡大となり、α が正規基底の生成元で, $\beta = \alpha^{q_{k-1}}$ の場合は正規基底表現の逐次拡大となる。

3.2 逐次拡大演算

この章では、正規基底表現の逐次拡大体の演算方法を示す。

正規基底の生成元を α および β として、正規多項式 $f(x) = (x - \alpha)(x - \beta) = x^2 + v_1x + v_0$ とおく。

乗算は以下のように行なう。

[逐次拡大体 乗算アルゴリズム]

Input: $A = (a_0, a_1), B = (b_0, b_1) \in GF(q_k)$
Output: $C = AB = (c_0, c_1) \in GF(q_{k-1})$

Step 1: $t = (a_0 - a_1)(b_0 - b_1)\dfrac{v_0}{v_1}$ を求める。
Step 2: $c_0 = a_0b_0v_1 - t$ を求める。
Step 3: $c_1 = a_1b_1v_1 - t$ を求める。
Step 4: $C = (c_0, c_1)$ を出力する。

$\mathrm{GF}(q_k)$ 上の乗算のコストを $M(k)$ とすると、$\mathrm{GF}(q_{k-1})$ 上の乗算、加算、定数倍コスト $M(k-1), A(k-1), X(k-1)$ を用いて

$$M(k) \quad = \quad 3M(k-1) + 4A(k-1) + 3X(k-1)$$

と表すことが出来る。

このコストは、KO 法 [6] とほぼ同じであるが、v_0, v_1 を適当に選べば加算や定数倍を少なくすることができる。

逐次拡大による逆元は以下のように行なう。

[逐次拡大体 逆元アルゴリズム]

Input: $A = (a_0, a_1) \in GF(q_k)$
Output: $C = A^{-1} = (c_0, c_1) \in GF(q_{k-1}))$

Step 1: $n' = (a_0a_1v_1^2 + (a_0 - a_1)^2v_0)^{-1}$ を求める。
Step 2: $C = (a_1n', a_0n')$ を求め、出力する。

$\mathrm{GF}(q_k)$ 上の乗算のコストを $I(k)$ とすると、$\mathrm{GF}(q_{k-1})$ 上の乗算、自乗、逆元、加算、定数倍コスト $M(k-1), S(k-1), I(k-1), A(k-1), X(k-1)$ を用いて

$$I(k) = 3M(k-1)+S(k-1)+I(k-1)+2A(k-1)+2X(k-1)$$

と表すことが出来る。

ただし、アルゴリズム中の v_1^2 や $\dfrac{v_0}{v_1}$ は、あらかじめ求めておくことができるため、演算コストに含めていない。

3.3 逐次拡大構成法

この章では、3.2章で用いた正規多項式の生成法について述べる。

$\mathrm{GF}(q)$ $(2 \nmid q)$ 上の 2 次式

$$f(x) = ax^2 + bx + c \quad (a \neq 0) \qquad (1)$$

の既約性について考察する。

判別式

$$D = b^2 - 4ac$$

が $\mathrm{GF}(q)$ に平方根を持たないことが、式 (1) が $\mathrm{GF}(q)$ 上既約であるための必要十分条件である。

定理 1 GF(q) ($2 \nmid q$) 上の2次式既約多項式 $f(x) = ax^2 + bx + c$ において、$b \neq 0$ であれば、$f(x)$ は正規多項式である。

証明) $f(x)$ の2つの根 $\alpha = \dfrac{-b+\sqrt{D}}{2a}$, $\beta = \dfrac{-b-\sqrt{D}}{2a}$ が一次独立でないと仮定する。

$\alpha = x\beta$ がなりたつ $x \in \mathrm{GF}(q)$ が存在することから、

$$b(1-x) = (1+x)\sqrt{D}$$

$\sqrt{D}$ は GF(q) の元ではないので、$b = (1+x) = 0$ または $(1-x) = (1+x) = 0$。GF(p) の標数は 2 より大なので、$b = 0$ である。

したがって、$b \neq 0$ ならば $f(x)$ は正規多項式であることが導かれる。 □

以下では D の平方剰余性について考察する。

定義 1 GF(q) 上の関数 $\chi_{\mathrm{GF}(q)} : \mathrm{GF}(q) \to \mathrm{GF}(q)$ を以下のように定義する。

$$\chi_{\mathrm{GF}(q)}(x) = \begin{cases} 1 & x \text{ が GF}(q) \text{ での平方剰余} \\ 0 & x = 0 \\ -1 & x \text{ が GF}(q) \text{ での平方非剰余} \end{cases}$$

補題 1 GF(q) の乗法群の原始根を g とおく。このとき

$$\chi_{\mathrm{GF}(q)}(g^n) = (-1)^n$$

が成り立つ。

定理 2

$$\chi_{\mathrm{GF}(q)}(x) = x^{\frac{q-1}{2}}$$

定義 2 (Frobenius 写像)

$$\phi_q : \overline{\mathrm{GF}(q)} \to \overline{\mathrm{GF}(q)}; \quad x \mapsto x^q$$

定義 3 (Norm)

$$\begin{aligned} N_{\mathrm{GF}(q^2)/\mathrm{GF}(q)}(x) &= \prod_{i=1}^{[\mathrm{GF}(q^2):\mathrm{GF}(q)]} \phi_q^i(x) \\ &= x^{q+1} \end{aligned}$$

定理 3

$$\chi_{\mathrm{GF}(q^2)}(x) = \chi_{\mathrm{GF}(q)}(N_{\mathrm{GF}(q^2)/\mathrm{GF}(q)}(x))$$

証明) 定理2より、

$$\begin{aligned} \chi_{\mathrm{GF}(q^2)}(x) &= x^{\frac{q^2-1}{2}} \\ &= x^{(q+1)\frac{q-1}{2}} \\ &= \chi_{\mathrm{GF}(q)}(N_{\mathrm{GF}(q^2)/\mathrm{GF}(q)}(x)) \end{aligned}$$

となる。 □

定理 4 $h(X) = dX^2 + eX + f$ を GF(q) 上既約な二次式とし、$\mathrm{GF}(q^2) \cong \mathrm{GF}(q)[X]/(h(X))$ と表現されていたとする。また $h(\alpha) = 0$ とする。

このとき、$u, v \in \mathrm{GF}(q)$ に対して、以下の式が成り立つ。

$$N_{\mathrm{GF}(q^2)/\mathrm{GF}(q)}(u + v\alpha) = d^{-1}(du^2 - euv + fv^2)$$

証明)

$$\begin{aligned} \phi_q(h(\alpha)) &= \phi_q(d\alpha^2 + e\alpha + f) \\ &= d\phi_q(\alpha)^2 + e\phi_q(\alpha) + f \\ &= h(\phi_q(\alpha)) \\ &= 0 \end{aligned}$$

であり、$\alpha \notin \mathrm{GF}(q)$ であるので、$\alpha \neq \phi_q(\alpha)$ であるから、α と $\phi_q(\alpha)$ は $h(X) = 0$ の相異なる解である。従って、解と係数の関係より $\phi_q(\alpha) = -d^{-1}e - \alpha$ となる。

よって、

$$\begin{aligned} &N_{\mathrm{GF}(q^2)/\mathrm{GF}(q)}(u + v\alpha) = \\ &= (u + v\alpha)(u + v\phi_q(\alpha)) \\ &= (u + v\alpha)(u + v(-d^{-1}e - \alpha)) \\ &= u^2 - d^{-1}euv - d^{-1}ev^2\alpha - v^2\alpha^2 \\ &= u^2 - d^{-1}euv + d^{-1}fv^2 \\ &= d^{-1}(u^2 - euv + fv^2) \end{aligned}$$

となる。 □

4 評価

逐次拡大法による演算コストの評価を表1に示す。GF(p^m) 上の乗算、逆元を、GF(p) 上の乗算、逆元のコスト M, I を用いて表している。

ただし、GF(p^m) 上の加減算は、どの方法を用いても同じなので、省略した。

表 1: 演算コストの比較

乗算	
提案法 (逐次拡大 OEF)	$m^{1.58}M$
KO 法 (OEF)	$m^{1.58}M$
教科書法 (OEF)	m^2M

逆元	
提案法 (逐次拡大 OEF)	$I + \frac{4}{3}m^{1.58}M$
伊東 - 辻井法	$I + \log_2(m-1)m^{1.58}M$
ユークリッド互除法 (IP)[8]	$mI + (2m^2 + 3m)M$
Lim 法 (IM)[8]	$I + (3m^2 + 4m)M$

この表によって、逐次拡大 OEF は、従来の OEF と比べて逆元を高速に行なえることがわかる。乗算、自乗に

関しても同等の速度であり、同じ有限体 $GF(p^m)$ を実装する場合は、逐次拡大を用いた方が効率が良い。

また、我々は、逐次拡大 OEF の効率を確かめるために、楕円曲線加算を提案アルゴリズムに基づいて実装した。その結果を、表 2 に示す。この実装に用いたパラメータは以下の通りである。

$$\begin{aligned} p &= 2^{31} \\ f(X_1) &= X_1^2 - X_1 - 1 \\ g(X_2) &= X_2^2 - X_2 + \alpha_1 \\ h(X_3) &= X_3^2 - X_3 + \alpha_1\alpha_2 \end{aligned}$$

ただし、α_1 は $f(X_1)$ の根、α_2 は $g(X_2)$ の根であり、

$$\begin{aligned} \mathrm{GF}(p^2) &\simeq \mathrm{GF}(p)[X_1]/f(X_1) \\ \mathrm{GF}(p^4) &\simeq \mathrm{GF}(p^2)[X_2]/g(X_2) \\ \mathrm{GF}(p^8) &\simeq \mathrm{GF}(p^4)[X_3]/h(X_3) \end{aligned}$$

によって 248 bit の有限体 $\mathrm{GF}(p^8)$ を構成している。

表 2: 実装データ

提案法 (逐次拡大 OEF)

bit 長	楕円 2 倍	楕円加算	座標系
248 bit	11.14 μ s	9.66 μ s	アフィン座標

従来の OEF

bit 長	楕円 2 倍	楕円加算	座標系
217 bit	13.2 μ s	19.7 μ s	射影座標

比較対象は、我々が Eurocrypt'99 で発表した、OEF を用いた楕円演算 [7] である。逆元を高速に行なえるため、楕円演算も高速化することができた。

この表から、逐次拡大 OEF の方では、有限体のサイズが大きい場合にもより高速な実装が可能であることが分かる。

5 まとめ

新しい OEF のクラスを提案した。この OEF は、逐次拡大によって構成される。

この方法では、逆元を 1 回の $\mathrm{GF}(p)$ 逆元と約 1.5 回の $\mathrm{GF}(p^m)$ 乗算と同等の演算コストで行なうことが出来、従来最速であった Bailey らの方式に比べて、より実用的である。

また、逐次拡大の基底を求めるためのアルゴリズムを提案し、逐次拡大 OEF を実装することにより、従来法よりも楕円加算が約 2 倍、楕円 2 倍も約 20 % 高速化する効果があることを確認した。

参考文献

[1] D. V. Bailey and C. Paar, "Optimal Extension Fields for Fast Arithmetic in Public-Key Algorithms," Advances in Cryptology – CRYPTO '98, Lecture Notes in Computer Science 1462, pp.472-485, Springer, 1998.

[2] D. V. Bailey and C. Paar, "Inversion in Optimal Extension Fields," Conference on The Mathematics of Public Key Cryptography. The Fields Institute for Research in the Mathematical Sciences, Toronto, Ontario, 1999.

[3] H. Cohen, A. Miyaji and T. Ono, "Efficient Elliptic Curve Exponentiation Using Mixed Coordinates," Advances in Cryptology – ASIACRYPT'98, Lecture Notes in Computer Science 1514, pp.51-65, Springer-Verlag, 1998.

[4] J. L. Fan, C. Paar, "On Efficient Inversion in Tower Fields of Characteristic Two", 1997 IEEE International Symposium on Information Theory, June 29 - July 4, 1997, Ulm, Germany.

[5] T. Itoh and S. Tsujii. "A Fast algorithm for Computing Multiplicative Inverses in Finite Fields Using Normal Basis," IEICE Transactions Fundamentals of Electronics, Communications and Computer Sciences (Japan), Vol.J 70-A No.11, 1987.

[6] D. E. Knuth. Seminumerical Algorithms, Vol. 2 of The Art of Computer Programming. Addison Wesley, third edition, 1997.

[7] T. Kobayashi, H. Morita, K. Kobayashi and F. Hoshino, "Fast Elliptic Curve Algorithm Combining Frobenius Map and Table Reference to Adapt to Higher Characteristic," Advances in Cryptology – EUROCRYPT'99, Lecture Notes in Computer Science 1592, pp.176-189, Springer-Verlag, 1999.

[8] C. H. Lim and H. S. Hwang. "Fast Implementation of Elliptic Curve Arithmetic in $\mathrm{GF}(p^n)$," Public Key Cryptography — Third International Workshop on Practice and Theory in Public Key Cryptosystems, PKC2000, Lecture Notes in Computer Science 1751, pp. 405-421. Springer-Verlag, 2000.