Vol.2012-DPS-150 No.33
Vol.2012-CSEC-56 No.33
2012/3/1

MICAz への Vandermonde 行列乗算法の適用について

白　勢　政　明[†1]

ペアリング計算の高速化のために提案された Vandermonde 行列乗算法は，実際に PC 上でのペアリング計算には効果があることが報告されている．しかしながら，センサノード (MICAz) 上でのペアリング実装には，Gorla 等の方法を除く Vandermonde 行列乗算法は効果がなかった．本稿はその原因について考察する．

On Application of Vandermonde Multiplications to MICAz

Masaaki Shirase[†1]

Vandermonde multiplications proposed for improvement in the speed of pairing calculation actually has an effect on pairing calculation on PC. However, Vandermonde multiplications except one proposed by Gorla et al. don't have an effect on pairing calculation on sensor node (MICAz). This paper considers the cause.

1. はじめに

ワイヤレス・センサ・ネットワーク (WSN) は，各種のセンサ，通信機能, 演算処理機能, センシング機能を有するセンサノード，及び基地局から構成され，センサノードはセンサから得られるデータを基地局や他のセンサノードに通信することができる[6]．低消費電力，低価格及び小型という特徴により，センサノードはユビキタス・ネットワークに最適なデバイスであると言われている[11]．また，センサノードの計算資源を活用するために，プログラミング言語である NesC 言語とそのコンパイラ，TinyOS と呼ばれる OS が開発されている．その結果，データ転送，モニタリングといったアプリケーションがセンサノードに数十 KB の ROM 使用で実装可能となった[10]．

ところで，WSN ではセンサノードが管理者不在の場所への設置されるため，攻撃が容易であると指摘されてきた．更に，ネットワーク共通の攻撃以外にも，ノード選択攻撃や複製ノードの設置等の WSN 特有の攻撃法が存在する[7]．従って，WSN の安全性は重要な課題であり，公開鍵暗号技術の応用はその解決策となる[20]．そのため，センサノードへの公開鍵暗号の実装も近年盛んになっており，RSA 暗号，楕円曲線暗号，ペアリング暗号の実装例がある．これらのほとんどの実装例では，センサノードとして MICAz[26] やその後継機である IRIS が使用されている．

MICAz へのペアリング高速実装のため，白勢等は有限体乗算を効率的行う Block-comb 法を提案した．Block-comb 法とは，MICAz(と IRIS) が保有する 32 ワードの汎用レジスタを最大限活用することで，ストア命令とロード命令を削減し (ペアリング計算に必要な) 多倍長の乗算を高速に行う手法である[23]．但し，Block-comb 法の実装にはアセンブリで記述する必要がある．北村等は，Block-comb 法を使用してペアリング暗号を実装したところ，計算時間が 3.92 秒 (NesC 言語での実装) から 2.15 秒へと大幅に短縮することができた[14),15)]．

著者は更なるペアリング計算の高速化のために，Vandermonde 乗算法の採用を試みた．Vandermonde 乗算法は，ペアリング暗号のほとんどの場合で必要となる拡大体での高速乗算法であり，実際に PC 上のペアリング計算には効果がある．しかしながら，Vandermonde 乗算法 (特に白勢等の方法と佐々木等の方法) は，MICAz や IRIS 上へのペアリング計算には効果がなかった．その原因として，MICAz や IRIS が採用している CPU(ATmega128 シリーズ) が 8 ビット CPU であること，及びシフト演算が相対的に遅いことが考えられる．

2. センサーノード MICAz と IRIS

MICAz は Crossbow 社が開発したセンサノードの一つであり，研究用に最も使われているセンサネットワーク端末の一つである．IRIS は MICAz の後続機種である．これらは CPU として，ATMega128 シリーズを搭載している．

ATmega128 シリーズでは，レジスタ (8 ビット) 間加減算や論理演算は 1 クロックサイクル，レジスタ内データのシフト処理は 1 クロックサイクル，(1 レジスタ)×(1 レジスタ)=(2 レジスタ) 乗算は 2 クロックサイクル，ストア/ロード命令は 2 クロックサイクルで処理される．従って，一般的な汎用 CPU では高速だとされるシフト処理は，ATMega シリーズでは加減算と同じクロックサイクル数やストア/ロードや乗算の 1/2 のクロックサイクル数

†1 公立はこだて未来大学
Future University Hakodate

Vol.2012-DPS-150 No.33
Vol.2012-CSEC-56 No.33
2012/3/1

表 1 IRIS,MICAz 仕様比較

	IRIS	MICAz
クロック周波数 (MHz)	7.37	
ROM(KB)	128	
SRAM(KB)	8	4

を要するため，ATMega128 シリーズでは，シフト処理は特別高速な処理とは言えない．

3. η_T ペアリング

ペアリングと呼ばれる双線型写像を使用することで，ID ベース鍵交換[21], ID ベース暗号[3], ID ベース署名[19], リング署名[25], キーワード検索暗号[2], 効率的な放送型暗号[4], 証明書不要公開鍵暗号[1], タイムリリース暗号[5]，属性ベース暗号[9] といった様々な暗号プロトコルを効率的に実現できるようになっている．しかしながら，ペアリングの計算量は十分に小さいとは言えないため，ペアリング計算の削減の研究もなされている．

ペアリング計算のアルゴリズムは，Miller ループと最終べきから成っている．

η_T ペアリング

$$\eta_T : E(\mathbb{F}_q) \times E(\mathbb{F}_q) \to \mathbb{F}_{q^k}$$

は，Miller ループのループ数が少ない計算が高速なペアリングの一つである[⋆1]．ここで，$(q,k) = (2^n, 4)$ または $(3^n, 6)$ であり，E は超特異楕円曲線である．$\mathbb{F}_{2^n}$ 上の η_T ペアリングの計算は，Miller ループにおいて $\mathbb{F}_{2^{4n}}$($\mathbb{F}_{2^n}$ の 4 次拡大体) 上の乗算を繰り返し行う必要があり，同様に $\mathbb{F}_{3^n}$ 上の η_T ペアリングの計算は，Miller ループにおいて $\mathbb{F}_{3^{6n}}$($\mathbb{F}_{3^n}$ の 6 次拡大体) 上の乗算を繰り返し行う必要がある．

3.1 拡 大 体

有限体 $\mathbb{F}_q$ と自然数 k に対して，体の構造を持つ (つまり四則演算が定義される)$\mathbb{F}_q$ の k 次ベクトル空間 $\mathbb{F}_{q^k}$ を $\mathbb{F}_q$ の k 次拡大という．本稿では，$\mathbb{F}_{q^k}$ は $\mathbb{F}_q$ 上多項式基底で表現されているとする．この場合，$\mathbb{F}_{q^k}$ は $\mathbb{F}_q$ 係数の $k-1$ 次以下の多項式の全体として考えることができる．$\mathbb{F}_{q^k}$ 上の加減算には，多項式として加減算を行えば良い．また，$\mathbb{F}_{q^k}$ 上の乗算を行うためには，k 既約次多項式 f を一つ固定し，$a, b \in \mathbb{F}_{q^k}$ に対して普通に多項式としての乗算 ab を計算し，その結果に対して f での剰余を計算する．この場合，$\mathbb{F}_{q^k} = \mathbb{F}_q[\sigma]/(f(\sigma))$ (σ は多項式の変数) と表記する．数学的には f は k 既約次多項式であれば何でも良いが，実装を考慮すると効率的に剰余計算が行われる f を選ぶ必要がある．拡大体の乗算は本質的には多項式の乗算であるということが，本稿では重要となる．除算は Euclid の拡張アルゴリズムや Fermat の小定理を用いて計算できる．

k が合成数 $k = k_1 k_2$ の場合，$\mathbb{F}_{q^k}$ の構成にあたり，初めに $\mathbb{F}_q$ 係数の k_1 次多項式 $f(\sigma)$ を用意して $\mathbb{F}_{q^{k_1}} = \mathbb{F}_q[\sigma]/f(\sigma)$ と構成し，更に $\mathbb{F}_{q^{k_1}}$ 係数の k_2 次多項式 $g(\rho)$ を用意して $\mathbb{F}_{q^k} = \mathbb{F}_{q^{k_1}}[\rho]/g(\rho)$ と構成することもできる．

3.1.1 $\mathbb{F}_{2^{4n}}$ と $\mathbb{F}_{3^{6n}}$ の一般的な実装法

$\mathbb{F}_{2^n}$ 上の η_T ペアリングを実装するにあたり，$\mathbb{F}_{2^n}$ 上既約 2 次多項式 $f(\sigma)$ を用いて $\mathbb{F}_{2^{2n}} = \mathbb{F}_{2^n}[\sigma]/f(\sigma)$ と構成し，$\mathbb{F}_{2^{2n}}$ 上既約 2 次多項式 $g(\rho)$ を用いて $\mathbb{F}_{2^{4n}} = \mathbb{F}_{2^{2n}}[\rho]/g(\rho)$ と実装することが一般的である．この場合，(加減算のコストは無視すると)Karatsuba 法[13] により，$\mathbb{F}_{2^{2n}}$ の乗算は $\mathbb{F}_{2^n}$ の乗算 3 回で計算でき，$\mathbb{F}_{2^{4n}}$ の乗算は $\mathbb{F}_{2^{2n}}$ の乗算 3 回で計算できる．従って，$\mathbb{F}_{2^{4n}}$ の乗算は $\mathbb{F}_{2^n}$ の乗算 9 回で計算できる．

$\mathbb{F}_{3^n}$ 上の η_T ペアリングを実装するにあたり，$\mathbb{F}_{3^n}$ 上既約 2 次多項式 $f(\sigma)$ を用いて $\mathbb{F}_{3^{2n}} = \mathbb{F}_{3^n}[\sigma]/f(\sigma)$ と構成し，$\mathbb{F}_{3^{3n}}$ 上既約 3 次多項式 $g(\rho)$ を用いて $\mathbb{F}_{3^{6n}} = \mathbb{F}_{3^{2n}}[\rho]/g(\rho)$ と実装することが一般的である．この場合，(加減算のコストは無視すると)Karatsuba 法により，$\mathbb{F}_{3^{2n}}$ の乗算は $\mathbb{F}_{3^n}$ の乗算 3 回で計算でき，$\mathbb{F}_{3^{6n}}$ の乗算は $\mathbb{F}_{3^{2n}}$ の乗算 6 回で計算できる．従って，$\mathbb{F}_{3^{6n}}$ の乗算は $\mathbb{F}_{3^n}$ の乗算 18 回で計算できる．

4. ペアリング暗号のセンサノードへの既存実装

ここでは，MICAz 上へのペアリング暗号の実装例について説明する．

(1) TinyTate は 2007 年に Oliveira 等によって MICAz 上に実装された Tate ペアリングである[17]．素体 $\mathbb{F}_p$(p を 256 ビットの素数) 上の埋め込み次数 2 を用いた Tate ペアリングであり, 楕円曲線は $y^2 = x^3 + x$ を用いている．演算時間は 30.21 秒.

(2) TinyPBC は 2007 年に Oliveira 等によって実装された η_T ペアリングである[18]．有限体 $\mathbb{F}_{2^{271}}$ 上の埋め込み次数 4 を用いた η_T ペアリングで, 楕円曲線 $y^2 + y = x^3 + x^2$ を用いている．演算時間は 5.45 秒.

(3) 石黒等によって η_T ペアリングが 2007 年に実装された[12]．有限体 $\mathbb{F}_{3^{97}}$ 上の埋め込み次数 6 を用いた η_T ペアリングで, 楕円曲線 $y^2 = x^3 - x + 1$ を用いている．演算時間は 5.97 秒.

⋆1 η_T ペアリングと同様に¡Miller ループのループ数が少ないペアリングとして，Ate ペアリング (やその改良版) がある．Ate ペアリングの場合は，$\mathbb{F}_p$ 上の通常楕円曲線を用いる．

Vol.2012-DPS-150 No.33
Vol.2012-CSEC-56 No.33
2012/3/1

(4) NanoECC は 2008 年に Szczechowiak 等によって実装された η_T ペアリングである[24]．有限体 $\mathbb{F}_{2^{271}}$ 上の埋め込み次数 4 を用いた η_T ペアリングで, 楕円曲線 $y^2+y=x^3+x^2$ を用いている．演算時間は 10.96 秒.

(5) 宮崎等によって η_T ペアリングが 2009 年に実装された[23]．有限体 $\mathbb{F}_{2^{239}}$ 上の埋め込み次数 4 を用いた η_T ペアリングで, 楕円曲線 $y^2+y=x^3+x+1$ を用いている．演算時間は 1.93 秒．Block comb 法を適用.

(6) 北村等によって η_T ペアリングが 2009 年に実装された[14]．有限体 $\mathbb{F}_{3^{97}}$ 上の埋め込み次数 6 を用いた η_T ペアリングで, 演算時間は 3.92 秒.

(7) 北村等によって η_T ペアリングが 2010 年に実装された[15]．有限体 $\mathbb{F}_{3^{97}}$ 上の埋め込み次数 6 を用いた η_T ペアリングで, 演算時間は 2.15 秒．Block comb 法を適用.

5. Vandermonde 行列乗算法

Gorla 等によって，$\mathbb{F}_{3^n}$ 上の η_T ペアリングの計算の高速化を目的とした Vandermonde 行列を用いた $\mathbb{F}_{3^{6n}}$ 上乗算法が提案された[8]．その後，白勢等[23] や佐々木等[22] が Gorla 等の方法の改良を行った．なお,

$$V=\begin{pmatrix} 1 & e_0 & \cdots & e_0^m \\ 1 & e_1 & \cdots & e_1^m \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e_m & \cdots & e_m^m \end{pmatrix}$$

という形の行列を Vandermonde 行列という.

5.1 Gorla 等の方法

$\mathbb{F}_{3^n}$ 上の η_T ペアリングに必要な $\mathbb{F}_{3^{6n}}$ を多項式基底により

$$\begin{aligned} \mathbb{F}_{3^{2n}} &= \mathbb{F}_{3^n}[\sigma]/(\sigma+1), \\ \mathbb{F}_{3^{6n}} &= \mathbb{F}_{3^{2n}}[\rho]/(f(\rho)) \end{aligned}$$

と実装する場合を考える．ここで，$f(\rho)$ は $\mathbb{F}_{3^{2n}}$ 上 3 次既約多項式である．すると，$\mathbb{F}_{3^{6n}}$ の元は $a_0+a_1\rho+a_2\rho^2$ $(a_0,a_1,a_2\in\mathbb{F}_{3^{2n}})$ の形で書ける．また，σ は 1 の $\mathbb{F}_{3^{6n}}$ 上の原始 4 乗となるから，$\sigma^2=-1,\sigma^3=-\sigma$ が成り立つ.

$a(z)=a_0+a_1z+a_2+z^2, b_0+b_1z+b_2+z^2, a_i,b_j\in\mathbb{F}_{3^{2n}}$ とし，$a(z)\cdot b(z)$ の (多項式としての) 乗算結果を $c(z)=c_0+c_1z+c_2z^2+c_3z^3+c_4z^4$ とする．すると，$a(\rho),b(\rho)\in\mathbb{F}_{3^{6n}}$ であり，$\mathbb{F}_{3^{6n}}$ の積の結果は $c(\rho) \bmod f(\rho)$ で与えられる．また，$c_4=a_2b_2$ であることが分かる．Vandermonde 行列 V を

$$V=\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \sigma & \sigma^2 & \sigma^3 \\ 1 & \sigma^2 & \sigma^4 & \sigma^6 \\ 1 & \sigma^3 & \sigma^6 & \sigma^9 \end{pmatrix}=\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \sigma & \sigma^2 & \sigma^3 \\ 1 & -1 & 1 & -1 \\ 1 & -\sigma & -1 & \sigma \end{pmatrix} \quad (1)$$

とすると,

$$V\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}=\begin{pmatrix} a(1)b(1)-c_4 \\ a(\sigma)b(\sigma)-c_4 \\ a(-1)b(-1)-c_4 \\ a(-\sigma)b(-\sigma)-c_4 \end{pmatrix}$$

が成り立つので,

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix}=\underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -\sigma & -1 & \sigma \\ 1 & -1 & 1 & -1 \\ 1 & \sigma & -1 & -\sigma \end{pmatrix}}_{=V^{-1}}\begin{pmatrix} (a_0+a_1+a_2)(b_0+b_1+b_2)-c_4 \\ (a_0+a_1\sigma-a_2)(b_0+b_1\sigma-b_2)-c_4 \\ (a_0-a_1\sigma+a_2)(b_0-b_1\sigma+b_2)-c_4 \\ (a_0-a_1\sigma-a_2)(b_0-b_1\sigma-b_2)-c_4 \end{pmatrix} \quad (2)$$

によって計算できる.

5.1.1 Gorla の方法の計算コスト

式 (1) のような Vandermonde 行列を用いると，σ と $\mathbb{F}_{3^{2n}}$ の元との積の計算コスト，及び式 (2) における V^{-1} と列ベクトルとの積の計算コストは，σ と ρ が拡大体の基底であるため，ほとんど無視できる．また，$f(\rho)$ での剰余計算コストを無視し (実際に無視できる), 更に加減算のコストも無視すると，$\mathbb{F}_{3^{6n}}$ の乗算は，$c_4=a_2b_2$ の計算と式 (2) の列ベクトルの計算に必要な 5 回の $\mathbb{F}_{3^{2n}}$ 乗算となる.

$\mathbb{F}_{3^{2n}}$ 乗算は，Karatsuba 法を用いると 3 回の $\mathbb{F}_{3^n}$ 乗算で計算できる．従って，$\mathbb{F}_{3^{6n}}$ の乗算コストは，15 回の $\mathbb{F}_{3^n}$ 乗算となる．(3.1.1 節で説明したように，Karatsuba 法のみを用いる従来の方法では，$\mathbb{F}_{3^{6n}}$ の乗算コストは 18 回の $\mathbb{F}_{3^n}$ 乗算である．)

5.2 白勢等の方法

$\mathbb{F}_{2^{4n}}$ は $\mathbb{F}_{2^n}$ 上多項式基底 $\{1,z,z^2,z^3\}$ を用いて構成されているとし，$\mathbb{F}_{2^n}$ は $\mathbb{F}_2$ 上多項式基底 $\{1,x,x^2,x^3,\cdots,x^{n-1}\}$ を用いて構成されているとする.

Vol.2012-DPS-150 No.33
Vol.2012-CSEC-56 No.33
2012/3/1

$a(z) = a_0 + a_1 z + a_2 z^2 + a_3 z^3, b(z) = b_0 + b_1 z + b_2 z^2 + b_3 z^3 \in \mathbb{F}_{2^{4n}}$ に対して，(多項式としての) それらの積を $c(z) = c_0 + c_1 z + c_2 z^2 + c_3 z^3 + c_4 z^4 + c_5 z^5 + c_6 z^6$ とする．すると，直ちに $c_0 = a_0 b_0, c_6 = a_3 b_3$ であることが分かる．

Vandermonde 行列

$$V = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ x & x^2 & x^3 & x^4 & x^5 \\ x+1 & (x+1)^2 & (x+1)^3 & (x+1)^4 & (x+1)^5 \\ x^2 & x^4 & x^6 & x^8 & x^{10} \\ (x+1)^2 & (x+1)^4 & (x+1)^6 & (x+1)^8 & (x+1)^{10} \end{pmatrix}$$

を考える．すると，

$$V \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} = \underbrace{\begin{pmatrix} a(1)\cdot b(1) + a_0 \cdot b_0 + a_3 \cdot b_3 \\ a(x)\cdot b(x) + a_0 \cdot b_0 + a_3 \cdot b_3 \cdot x^6 \\ a(x+1)\cdot b(x+1) + a_0 \cdot b_0 + a_3 \cdot b_3 \cdot (x+1)^6 \\ a(x^2)\cdot b(x^2) + a_0 \cdot b_0 + a_3 \cdot b_3 \cdot x^{12} \\ a((x+1)^2)\cdot b((x+1)^2) + a_0 \cdot b_0 + a_3 \cdot b_3 \cdot (x+1)^{12} \end{pmatrix}}_{=D_P}.$$

が成り立つことが分かる．

$$\begin{aligned}
\alpha_{00} &= x^8 + x^6 + x^5 + x^3, & \alpha_{03} &= x^4 + x^3 + x^2 + x, \\
\alpha_{10} &= x^6 + x^5 + x^4 + x^3 + x^2 + x, & \alpha_{13} &= x^4 + x^3 + x^2 + 1, \\
\alpha_{20} &= x^6 + x^5 + x^4 + x^3 + 1, & \alpha_{23} &= x^2 + x + 1, \\
\alpha_{30} &= 0, & \alpha_{33} &= x^2 + 1, \\
\alpha_{40} &= x^2 + x + 1, & \alpha_{43} &= 1, \\
\alpha_{01} &= x^7 + x^3, & \alpha_{04} &= x^4 + x^3, \\
\alpha_{11} &= x^7 + x^6 + x^5 + x^4 + x^3 + x, & \alpha_{14} &= x^4 + x^3 + x, \\
\alpha_{21} &= x^6 + x^5 + x^4 + x^3 + x^2 + x, & \alpha_{24} &= x^2 + x + 1, \\
\alpha_{31} &= x^3 + x, & \alpha_{34} &= x^2, \\
\alpha_{41} &= x^2 + x, & \alpha_{44} &= 1, \\
\alpha_{02} &= x^7 + x^6 + x^5 + x^4, & \beta &= x^8 + x^6 + x^5 + x^3, \\
\alpha_{12} &= x^7 + x^2, & & \\
\alpha_{22} &= x^6 + x^5 + x^4 + x^3 + x^2 + x, & & \\
\alpha_{32} &= x^3 + x^2 & & \\
\alpha_{42} &= x^2 + x, & &
\end{aligned}$$

とおくと，$V^{-1} = \frac{1}{\beta}(\alpha_{ij})$ が成り立つ．(この等式は標数 2 の体のみで成り立つ．)

注意 1: $\mathbb{F}_{2^n}$ 上の η_T ペアリングの Miller ループの計算において，$\mathbb{F}_{2^{4n}}$ の元の乗算 $a(z)b(z)$ の計算の代わりに $\gamma a(z)b(z)$ $(\gamma \in \mathbb{F}_{2^n}, \gamma \neq 0)$ を計算しても，最終べきの効果により同じ値が得られる．

$d(z) = \beta c(z)$ とおくと，$d(z)$ は

$$\begin{cases} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{pmatrix} = (\alpha_{ij}) D_P \\ d_0 = \beta \cdot a_0 \cdot b_0, \\ d_6 = \beta \cdot a_3 \cdot b_3. \end{cases} \tag{3}$$

によって，計算できる．

5.2.1 白勢等の方法の計算コスト

$\mathbb{F}_{2^n}$ の乗算を，シフトアド法によって計算する場合を考える．

$\mathbb{F}_{2^n}$ の元 $e(x) = \sum_{i=0}^{n-1} e_i x^i$ に対して，$deg(e)$ を e_i が 0 でない最大の i と定義する．すると，$e(x), f(x) \in \mathbb{F}_{2^n}$ に対して，$deg(b) = m$ ならば，シフトアド法に必要な加算の回数は m となる．$deg(b)$ が分かっている時，乗算に必要な加算の回数は一般の場合の乗算の $\frac{deg(b)}{n-1}$ 倍となる．

この事実から，

(1) シフトアド法におけるシフト処理のコストを無視できる，

(2) $(\alpha_{ij}$ のように$)deg(f)$ の小さい場合，$e(x)f(x)$ の計算コストは $\frac{deg(f)}{n-1}$ 倍となる，

と仮定する．すると，$\gamma a(z)b(z)$ の計算コストは $7 + 90/(n-1)$ 回の $\mathbb{F}_{2^n}$ 乗算となる．(3.1.1 節で説明したように，Karatsuba 法のみを用いる従来の方法では，$\mathbb{F}_{2^{4n}}$ の乗算コストは 9 回の $\mathbb{F}_{2^n}$ 乗算である．)

5.3 佐々木等の方法

Gorla 等の方法とは異なり，$\mathbb{F}_{3^{6n}}$ は $\mathbb{F}_{3^n}$ 上多項式基底 $\{1, z, z^2, z^3, z^4, z^5\}$ を用いて構成されているとする．また，$\mathbb{F}_{3^n}$ は $\mathbb{F}_3$ 上多項式基底 $\{1, x, x^2, x^3, \cdots, x^{n-1}\}$ を用いて構成されているとする．

Vol.2012-DPS-150 No.33
Vol.2012-CSEC-56 No.33
2012/3/1

$a(z)=a_0+a_1z+a_2z^2+a_3z^3+a_4z^4+a_5z^5, b(z)=b_0+b_1z+b_2z^2+b_3z^3+b_4z^4+b_5z^5 \in \mathbb{F}_{3^{6n}}$ に対して，(多項式としての) それらの積を $c(z)=c_0+c_1z+c_2z^2+c_3z^3+c_4z^4+c_5z^5+c_6z^6+c_7z^7+c_8z^8+c_9z^9+c_{10}z^{10}$ とする．すると，直ちに $c_0=a_0b_0, c_9=a_4b_5+a_5b_4, c_{10}=a_5b_5$ であることが分かる．

$(z_1,z_2,z_3,z_4,z_5,z_6,z_7,z_8)=(1,2,x,x+1,x+2,-x,-(x+1),-(x+2))$ にとおき，

$$V=\begin{pmatrix} 1 & z_1 & \cdots & z_1^6 & z_1^7 \\ 1 & z_2 & \cdots & z_2^6 & z_2^7 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & z_7 & \cdots & z_7^6 & z_7^7 \\ 1 & z_8 & \cdots & z_8^6 & z_8^7 \end{pmatrix}$$

という Vandermonde 行列を考える．すると，

$$V\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_7 \\ d_8 \end{pmatrix}=\underbrace{\begin{pmatrix} A(1)B(1)-d_0-d_9-d_{10} \\ A(2)B(2)-d_0+d_9-d_{10} \\ \vdots \\ A(-(x+1))B(-(x+1))-d_0+d_9(x+1)^9-d_{10}(x+1)^{10} \\ A(-(x+2))B(-(x+2))-d_0+d_9(x+2)^9-d_{10}(x+2)^{10} \end{pmatrix}}_{D_P}$$

が成り立つことが分かる．

$$\begin{aligned}
\gamma_{11} &= -(x^6+x^4+x^2) & \gamma_{12} &= x^6+x^4+x^2 \\
\gamma_{21} &= -(x^6+x^4+x^2) & \gamma_{22} &= -(x^6+x^4+x^2) \\
\gamma_{31} &= 1 & \gamma_{32} &= -1 \\
\gamma_{41} &= 1 & \gamma_{42} &= 1 \\
\gamma_{51} &= 1 & \gamma_{52} &= -1 \\
\gamma_{61} &= 1 & \gamma_{62} &= 1 \\
\gamma_{71} &= 1 & \gamma_{72} &= -1 \\
\gamma_{81} &= 1 & \gamma_{82} &= 1
\end{aligned}$$

$$\begin{aligned}
\gamma_{13} &= -(x^5+x^3+x) & \gamma_{14} &= -(x^5+2x^4+2x^3+x^2) \\
\gamma_{23} &= -(x^4+x^2+1) & \gamma_{24} &= -(x^4+x^3+x^2) \\
\gamma_{33} &= x^5 & \gamma_{34} &= (x+1)^5 \\
\gamma_{43} &= x^4 & \gamma_{44} &= (x+1)^4 \\
\gamma_{53} &= x^3 & \gamma_{54} &= (x+1)^3 \\
\gamma_{63} &= x^2 & \gamma_{64} &= (x+1)^2 \\
\gamma_{73} &= x & \gamma_{74} &= (x+1) \\
\gamma_{83} &= 1 & \gamma_{84} &= 1 \\
\gamma_{15} &= -(x^5+x^4+2x^3+2x^2) & \gamma_{16} &= x^5+x^3+x \\
\gamma_{25} &= -(x^4+2x^3+x^2) & \gamma_{26} &= -(x^4+x^2+1) \\
\gamma_{35} &= (x+2)^5 & \gamma_{36} &= -x^5 \\
\gamma_{45} &= (x+2)^4 & \gamma_{46} &= x^4 \\
\gamma_{55} &= (x+2)^3 & \gamma_{56} &= -x^3 \\
\gamma_{65} &= (x+2)^2 & \gamma_{66} &= x^2 \\
\gamma_{75} &= (x+2) & \gamma_{76} &= -x \\
\gamma_{85} &= & \gamma_{86} &= 1 \\
\gamma_{17} &= x^5+2x^4+2x^3+x^2 & \gamma_{18} &= x^5+x^4+2x^3+2x^2 \\
\gamma_{27} &= -(x^4+x^3+x^2) & \gamma_{28} &= -(x^4+2x^3+x^2) \\
\gamma_{37} &= -(x+1)^5 & \gamma_{38} &= -(x+2)^5 \\
\gamma_{47} &= (x+1)^4 & \gamma_{48} &= (x+2)^4 \\
\gamma_{57} &= -(x+1)^3 & \gamma_{58} &= -(x+2)^3 \\
\gamma_{67} &= (x+1)^2 & \gamma_{68} &= (x+2)^2 \\
\gamma_{77} &= -(x+1) & \gamma_{78} &= -(x+2) \\
\gamma_{87} &= 1 & \gamma_{88} &= 1 \\
\beta &= x^6+x^4+x^2 &&
\end{aligned}$$

とおくと，$V^{-1}=\frac{1}{\beta}(\alpha_{ij})$ が成り立つ．(この等式は標数 3 の体のみで成り立つ．)

注意 2:

注意 1 と同様に，$\mathbb{F}_{3^n}$ 上の η_T ペアリングの Miller ループの計算において，$\mathbb{F}_{3^{6n}}$ の元の乗算 $a(z)b(z)$ の計算の代わりに $\gamma a(z)b(z)$ $(\gamma \in \mathbb{F}_{3^n}, \gamma \neq 0)$ を計算しても，最終べきの効果により同じ値が得られる．

Vol.2012-DPS-150 No.33
Vol.2012-CSEC-56 No.33
2012/3/1

$d(z) = \beta c(z)$ とおくと，

$$\begin{cases} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{pmatrix} = (\alpha_{ij}) D_P \\ d_0 = \beta \cdot a_0 \cdot b_0, \\ d_9 = \beta \cdot (a_4 \cdot b_5 + a_5 \cdot b_4), \\ d_{10} = \beta \cdot a_5 \cdot b_5. \end{cases} \tag{4}$$

によって，計算できる．

5.3.1 佐々木等の方法の計算コスト

5.2.1 節と同様に，

(1) シフトアド法におけるシフト処理のコストを無視できる，

(2) $(\alpha_{ij}$ のように$)deg(f)$ の小さい場合，$e(x)f(x)$ の計算コストは $\frac{deg(f)}{n-1}$ 倍となる，

と仮定する．すると，$\gamma a(z)b(z)$ の計算コストは $12+132/(n-1)$ 回の $\mathbb{F}_{3^n}$ 乗算となる．(従来の Karatsuba 法では場合は 18 回，Gorla の方法では 15 回．)

実際に，PC 上では $\mathbb{F}_{3^{97}}$ 上での η_T ペアリングの計算時間は，5.01ms から 4.76sm に削減されたことが報告されている[22]．

6. MICAz 上でのペアリング計算における Vandermonde 行列乗算法の適用

Vandermonde 乗算法のうち，Gorla の方法は既に一般的になっており，4 節で説明した $\mathbb{F}_{3^n}$ 上 η_T ペアリングの実装で用いられている．従って，本稿の目的は白勢等の方法と佐々木等の方法の適用を考察することである．

NesC 言語を用いて，$\mathbb{F}_{2^{239}}$ 上 η_T ペアリングで必要となる $\mathbb{F}_{2^{4\cdot 239}}$ 乗算を IRIS に実装した結果，$\mathbb{F}_{2^{4\cdot 239}}$ の乗算に Karatsuba 法を用いた場合の計算時間は 13.93 ミリ秒，白勢等の方法を用いた場合の計算時間は 14.02 ミリ秒となり，白勢等の方法を用いた場合の方が計算時間は長くなった．

NesC 言語を用いて，$\mathbb{F}_{3^{6\cdot 97}}$ 上 η_T ペアリングで必要となる $\mathbb{F}_{3^{6\cdot 97}}$ 乗算を IRIS に実装した結果，$\mathbb{F}_{3^{6\cdot 97}}$ の乗算に Karatsuba 法を用いた場合の計算時間は 12.3 ミリ秒，佐々木等の方法を用いた場合の計算時間は 14.1 ミリ秒となり，佐々木等の方法を用いた場合の方が計算時間は長くなった．

5.2.1 節や 5.2.2 節を考慮すると，白勢等の方法や佐々木等の方法を用いた方が高速になることが期待されたが，実際にはそうはならなかった．5.2.1 節や 5.2.2 節の計算量の見積もりにおいて，シフト処理は無視できると仮定しているが，IRIS ではこの仮定が成り立たないためと考えられる．

$\mathbb{F}_{2^{239}}$ の元には，32 ビット CPU では 8 ワードを使用する．従って，$\mathbb{F}_{2^{239}}$ の元を 1 回シフトするには，CPU の命令として 8 回のシフト命令が必要となる．

これに対して，MICAz や IRIS が搭載している ATMega128 シリーズのような 8 ビット CPU では $\mathbb{F}_{2^{239}}$ の元には，30 ワードを使用する．そのため，$\mathbb{F}_{2^{239}}$ の元を 1 回シフトするには，CPU の命令とし 30 回のシフト命令が必要となる．

更に ATMega128 シリーズでは，そもそも 1 ワードのためのシフト命令が加減算と同じだけ時間を要する命令であり，この事実が $\mathbb{F}_{2^{239}}$ の元のシフトを相対的に高コストとしている．

数値計算において 2 倍をする代わりに左シフトを，2 で割る代わりに右シフトを使うという手法は，(PC 上の) 高速実装のために良く用いられる一般的手法であるが，MICAz や IRIS では効果がないこととなる．

逆を考えると，64 ビット CPU では $\mathbb{F}_{2^{239}}$ の元は 4 ワードで表現でき，$\mathbb{F}_{2^{239}}$ の元のシフトには，CPU の命令とし 4 回のシフト命令だけで済む．従って，64 ビット CPU では $\mathbb{F}_{2^{239}}$ の元のシフト処理は，相対的により低コストな処理となる．従って，白勢等の方法や佐々木等の方法は，64 ビット CPU において効果を発すると考えられる．

今回は，白勢等の方法や佐々木等の方法の実装にあたり NesC 言語を用いており，アセンブリによる実装は行っていない．従って，シフト処理をアセンブリで実装すれば，これらの方法は IRIS 上の計算においても効果があるかもしれず，その調査は今後の課題としたい．

同様の考察は，$\mathbb{F}_{3^{97}}$ の元のシフトについても成り立つ．

7. ま と め

シフト処理を無視できる場合，Vandermonde 行列乗算法である白勢等の方法や佐々木等の方法は，実際に η_T ペアリングにおける拡大体乗算の計算コストを削減できる．しかしながら，これらの手法を用いてセンサノード (IRIS) に実装したところ，計算時間はかえって

Vol.2012-DPS-150 No.33
Vol.2012-CSEC-56 No.33
2012/3/1

遅くなった．IRIS が搭載している CPU は 8 ビットの ATMega128 シリーズであるが，計算時間が遅くなる原因として，

(1) 8 ビット CPU を用いるとき，32 ビットや 64 ビット CPU と比較して，有限体の元のシフトに必要なシフト命令の回数が増大する，

(2) ATmega128 シリーズでは，そもそもシフト命令は加減算と同じ時間を要し，速い処理とは言えない

であることが判明した．

WSN の普及に伴って，センサノードへのアプリケーションの実装の必要性が高まると思われるが，シフト処理は速いという常識は注意が必要である．

謝　辞

本稿は，公立はこだて未来大学の 2010 年度の卒業生，瀬尾孝幸さんと寺井明日実さんの卒業研究の成果を基にしています．両氏に感謝いたします．また，本研究は科研費 (若手 B21700083) の助成を受けて行われました．

参 考 文 献

1) S. Al-Riyami and K. Peterson, “Certificateless public keycryptography,” ASIACRYPT 2003, LNCS 2894, pp.452-474, 2003.
2) D. Boneh, G. Crescenzo, R. Ostrovsky, and G. Persiano, “Public Key Encryption with Keyword Search,” EUROCRYPT 2004, LNCS 3027, pp.506-522, 2004.
3) D. Boneh and M. Franklin, “Identity based encryption from The Weil pairing,” SIAM Journal of Computing, Vol. 32, No. 3, pp.586-615, 2003.
4) D. Boneh, G. Gentry, and B. Waters, “Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys,” CRYPTO 2005, LNCS 3621, pp.258-275, 2005.
5) K. Chalkias and G. Stephanides, “Timed Release Cryptography from Bilinear Pairings Using Hash Chains,” CMS 2006, LNCS 4237, pp.130-140, 2006.
6) D. Estrin, R. Govindan, J. Heidemann, and S. Kumar “Next Century Challenges: Scalable Coordination in Sensor Networks,” Mobicom99, pp.263-270, 1999.
7) V. Gligor, “Advances in Aensor and Ad-hoc Network Security: Perspective and Status,” SASN 2005, pp.68, 2005.
8) E. Gorla, C. Puttmann, and J. Shokrollahi, “Explicit Formulas for Efficient Multiplication in $\mathbb{F}_{3^{6m}}$,” SAC 2007, LNCS 4876, pp.173-183, 2007.
9) V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based Encryption for Fine-grained Access Control of Encrypted Data,” CCS 2010, pp.89-98, 2006.
10) B. Greenstein, E. Kohler, and D. Estrin, “A sensor network application, construction kit (SNACK),” 2nd ACM SenSys Conference, pp. 69-80, 2004.
11) J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System Architecture Directions for Networked Sensors,” ASPLOS IX, pp.93-104, 2000.
12) 石黒司, 白勢政明, 高木剛, “ATmega128L 上でのペアリング暗号の高速実装,” CSS2007 予稿集, pp.187-192, 2007.
13) A. Karatsuba, and Y. Ofman, “Multiplication of Multidigit Numbers on Automata,” Soviet Physics-Doklady, Vol.7, pp.595-596, 1963.
14) 北村晃輔，白勢政明，“MICAz 上での η_T ペアリング高速実装,” CSS2009 予稿集, pp. 349-354, 2009.
15) 北村晃輔，白勢政明，“MICAz 上における標数 3 の η_T ペアリングの高速実装,” SCIS2010 予稿集, 3F1-4, 2010.
16) 宮崎行規, 白勢政明, 高木剛，“MICAz 上での高速ペアリング暗号実装,” CSS2008 予稿集, pp. 199-204, 2008.
17) L. Oliveira, D. Aranha, E. Morais, F. Daguano, J. Lopez, and R, Dahab, “TinyTate: Identity-Based Encryption for Sensor Networks,” Cryptology ePrint Archive 2007/020, 2007.
18) L. Oliveira, M. Scott, J. Lopez, and R. Dahab, “TinyPBC: Pairings for Authenticated Identity-Based Non-Interactive Key Distribution in Sensor Networks,” Cryptology ePrint Archive 2007/482, 2007.
19) K. Paterson, “ID-based Signatures rom Pairings on Elliptic Curves,” Cryptology ePrint Archive 2002/004, 2002.
20) A. Perrig, J. Stankovic, and D. Wagner, “Security in Wireless Sensor Networks,” Communications of The ACM, Vol.47 No.6, 2004.
21) R. Sakai, K. Ohgishi, and M. Kasahara, “Cryptosystems based on pairing,” SCIS2000, 2000.
22) Y. Sasaki, S. Nishina, M. Shirase, and T, Takagi, “An Efficient Residue Group Multiplication for The η_TPairing over $\mathbb{F}_{3^m}$,” SAC2009, LNCS5867, pp.364-375, 2009.
23) M. Shirase, Y. Miyazaki, T. Takagi, D.-G, Han, D. Choi, “Efficient Implementation of Pairing-Based Cryptography on a Sensor Node,” IEICE Transactions 92-D(5), pp.909-917, 2009.
24) P. Szczechowiak, L. Oliveira, M. Scott, M. Collier, and R. Dahab, “NanoECC:Testing The Limits of Elliptic Curve Cryptography in Sensor Network,” EWSN 2008, LNCS 4913, pp.305-320, 2008.
25) F. Zhang and K. Kim, “ID-Based Blind Signature and Ring Signature from Pairings,” ASIACRYPT 2002, LNCS 2501, pp.629-637, 2002.
26) ZigBee 版無線センサネットワーク MOTO MICAz, http://www.xbow.jp/motemica.html