情報処理学会論文誌 Vol. 50 No. 7 1745–1756 (July 2009)

素体上の超特異楕円曲線におけるペアリング暗号の効率的な計算手法[⋆1]

中島俊哉[†1,⋆1] 伊豆哲也[†2] 高木剛[†3]

本稿では標数 5 以上の素体 GF(p) 上の超特異楕円曲線でのペアリング計算の高速化を行う．この曲線での高速化は，素数 p に適切な条件を設定することで可能となる．すなわち，$p+1$ の最大素因数 r が低 Hamming 重みであれば Miller アルゴリズムが高速化され，p が低 Hamming 重みであれば Miller アルゴリズム内部の Montgomery 乗算が高速化される．一方，低 Hamming 重みの素数での離散対数問題については最近提案された攻撃手法が存在する．したがってペアリング計算を高速化する素数 p はこの攻撃に対する安全性も持つ必要があるが，そのような素数の探索アルゴリズムはこれまで提案されていない．本稿では標数 5 以上の素体上の超特異楕円曲線でのペアリングを対象として，ペアリング計算速度の高速化と，上記の新規攻撃法に対する安全性，および p，r のビット長に関する従来の安全性の全てを満たす素数の探索手法を提案する．また，探索で見い出された素数を用いることにより，安全性を満たすと同時に Miller アルゴリズムの計算速度を約 22%高速化した結果を示す．

An Efficient Algorithm for Pairing Cryptography with Supersingular Elliptic Curves over Prime Fields

Toshiya Nakajima,[†1,⋆1] Tetsuya Izu[†2] and Tsuyoshi Takagi[†3]

The theme of this paper is to achieve fast pairing computation with supersingular elliptic curves over prime field GF(p) where p is greater than or equal to 5. Fast pairing computation for these curves is possible by using suitably chosen prime number p. That is, if largest prime factor r of $p+1$ has low Hamming weight then the Miller algorithm runs efficiently, and if p has low Hamming weight then the Montgomery multiplication inside the Miller algorithm also runs efficiently. On the other hand, a new attack to discrete logarithm problem with prime numbers of low Hamming weight has been proposed recently. Thus it is necessary to find prime numbers that realize both efficient pairing computation and security against the attack. However, no algorithms to find such prime numbers have been proposed ever. In this paper we propose a search method to find prime numbers that realize both fast pairing computation for the curves and security against the new attack (the well-known security conditions regarding bitlength of p and r are also satisfied). With a prime number found by the method we achieved speeding up of about 22% in computation of the Miller algorithm as well as security against the new attack.

1. はじめに

楕円曲線上の点のペアリングが持つ双線形性を利用することにより，ID ベース暗号をはじめ多くの暗号プロトコルが実現可能となる．ペアリング暗号の実装対象は，近年，専用ハードウェア[3] から携帯電話等の組み込みシステム用 CPU[17] まで広範囲にわたる．実装では高速計算の実現が重要であるため，ペアリング計算アルゴリズムとしては，η_T ペアリング[1] および Ate ペアリング[7] が多く用いられる．これらは，Tate ペアリングの計算アルゴリズムとして最初に提案された Miller アルゴリズム[10] を基として新規に考案された高速アルゴリズムである．また，これらをさらに高速化するため，有限体乗算の高速化手法[12] 等，多数の研究も行われている．

他方，ペアリング暗号を実装するためには既存リソースの再利用も重要であり，Miller アルゴリズムを使用する Tate ペアリングは，楕円曲線暗号でのスカラー倍算へのコード追加で実現できる利点がある．実際に，η_T および Ate ペアリングの適用対象でない曲線のうち，パラメータの選択により Tate ペアリングの高速計算を可能とする曲線が最近注目されており，標準化が進められている．それらの曲線は素体 GF(p)（$p \geq 5$）上の超特異楕円曲線に属する，以下の (a)，(b) の曲線であり，(b) の曲線は Internet Engeneering Task Force (IETF) の RFC5091 で Type-1 曲線として提案されている[2]．

(a) $y^2 = x^3 + x$; $p \equiv -1 \pmod 4$,

†1 富士通株式会社
Fujitsu Ltd.
†2 株式会社富士通研究所
Fujitsu Laboratories Ltd.
†3 公立はこだて未来大学大学院システム情報科学研究科
Graduate School of Systems Information Science, Future University-Hakodate
⋆1 現在，富士通マイクロエレクトロニクス株式会社出向中
Presently with Fujitsu Microelectronics Ltd.
⋆1 本稿の preliminary 版[13] は，2008 年暗号と情報セキュリティシンポジウム（SCIS2008）において発表された．

© 2009 Information Processing Society of Japan

(b) $y^2 = x^3 + 1$;　$p \equiv -1 \pmod{12}$.

これらの曲線はペアリング計算の高速化を可能とする次の 2 つの特長を持つ．(1) 各項の係数が 1 のため曲線上の点の演算における係数の乗算を省略できる．(2) 曲線の群位数が $p+1$ であり上記 mod の条件と合わせて，素数の半数がいずれかの曲線の群位数に対応する．したがって高速化に適した群位数が設定できれば，その位数に対応する素数 p を選べる可能性が高いことが期待できる．

一方，ペアリング暗号の安全性については，従来から楕円曲線離散対数問題および離散対数問題の対象となる群の位数のビット長により見積もられている[14] が，最近 Schirokauer による攻撃手法[15] が提案されている．これは，楕円曲線の定義体 GF(p) の素数 p が低 Hamming 重みの場合に有効な攻撃手法であり，この場合 p のビット長が長くとも安全とは限らない．したがってペアリング計算の高速化等の目的で低 Hamming 重みの p を用いる場合には，安全性条件としてビット長に加えて Schirokauer 攻撃に対する安全性も持つことが必要となる．Schirokauer 攻撃に対しては Hamming 重みの増加によって安全性が向上するが，他方で高速化の効果が低下する可能性もある．したがって安全性と高速化が両立する素数を用いることが望まれるが，そのような素数を見い出すアルゴリズムはこれまで提案されていなかった．

1.1　本稿での貢献

本稿では (a) の曲線でのペアリング計算を高速化する素数 p の探索において，Schirokauer 攻撃に対する安全性条件を組み込んだ探索アルゴリズムを提案し，見い出された素数を用いたペアリング計算での高速化を実証する．

ペアリング暗号の安全性には，ペアリング計算の入力側での楕円曲線離散対数問題（ECDLP）における安全性と，出力側での離散対数問題（DLP）における安全性とが存在する．Schirokauer 攻撃は低 Hamming 重み素数を用いる DLP に対する攻撃手法である．ペアリング暗号において Schirokauer 攻撃を考慮した安全性はこれまで提案されていないことから，本稿では Schirokauer 攻撃に対する安全性条件を新規に導出する．

各種の攻撃手法はそれらに固有の時間計算量を持つため，安全性のためにはこの計算量が十分大きくなるパラメータ値を選択する必要がある．Schirokauer 攻撃では，p の符号付き二進表現（以下 SD 表現）から定義される値 γ が，安全性を判断するための重要なパラメータとなる[⋆1]．Schirokauer 攻撃の計算量は γ の関数であり，γ は p の各 SD 表現ごとに計算される値であるため，全ての γ について攻撃計算量は，安全性の限界となる計算量を上回る必要がある．しかしながら，p の l 桁の SD 表現を全て求めるために 3^l のオーダの組合せを試す方法は指数時間の計算量が必要であり，後続の章で述べるとおり p が 512 ビットの場合には現実的ではない．

そのため本稿では p の全ての SD 表現について γ を求める代わりに，全ての γ についての共通の上界値 γ_{ubound} を用いる．γ_{ubound} は実際に導出が可能であり，これが本稿の主要結果の 1 つとなる．続いて，γ_{ubound} を用いて Schirokauer 攻撃に対する安全性条件を p の探索条件に組み込み，高速性と安全性を持つ p の探索アルゴリズムを提案する．さらに，この探索アルゴリズムで見い出された素数に特化した Montgomery 乗算を実装し，Schirokauer 攻撃に対する安全性を満たすと同時にペアリング計算速度を約 22%高速化した結果を示す．

2.　Tate ペアリングの計算アルゴリズム

標数 5 以上の素体を以下では GF(p) とする．本稿では，GF(p) 上の超特異楕円曲線[5] に属する次の曲線 E_{ss} を対象とする．

$$E_{ss}/\mathrm{GF}(p):\quad y^2 = x^3 + x;\quad p \equiv -1 \pmod{4}. \tag{1}$$

E_{ss} を用いるペアリング計算では，p を適切に選択することで Miller アルゴリズムおよび Montgomery 乗算[11] の高速化が可能になる．本章ではこれらの高速化を実現する p の条件について述べる．

2.1　Tate ペアリングの定義

q を素数の冪，E を有限体 GF(q) 上定義された楕円曲線とする．Tate ペアリングは写像

$$\langle \cdot, \cdot \rangle_r :\quad E(\mathrm{GF}(q))[r] \times E(\mathrm{GF}(q^k))/rE(\mathrm{GF}(q^k)) \quad \rightarrow \quad \mathrm{GF}(q^k)^*/(\mathrm{GF}(q^k)^*)^r \tag{2}$$

として定義される．ここで r は $\#E(\mathrm{GF}(q))$ の最大素因数，k は埋め込み次数と呼ばれ，$r|q^k - 1$ を満たす最小の正整数を表す．写像の右辺が商集合であることから，写像先の元は一意ではない．実際の Tate ペアリングの計算では写像先の元に最終冪と呼ばれる演算を適用して商集合の代表元を求め，これをペアリングの値とすることが多い．これは reduced Tate ペアリングと呼ばれる．reduced Tate ペアリングの値は GF(q^k)* の元となる．

$e(P, Q)$ を reduced Tate ペアリング，$P \in E(\mathrm{GF}(q))[r]$，$Q \in E(\mathrm{GF}(q^k))/rE(\mathrm{GF}(q^k))$

⋆1 本稿では非負整数 n について「通常の二進表現」「符号付き二進表現」の用語を次のとおり定義する．$n = \sum_{i=0}^{l} d_i 2^i$（$l > 0$ のとき $d_l \neq 0$）と展開し n を $d_l d_{l-1} \cdots d_0$ で記述するとき，d_i として 0，1 のみを用いる場合に $d_l d_{l-1} \cdots d_0$ を「通常の二進表現」と定義し，単に「二進表現」とも呼ぶ．二進表現は $(n)_2$ または $(d_l d_{l-1} \cdots d_0)_2$ とも記述する．他方，d_i として 0，1，$-1 =: \bar{1}$ のみを用いる場合に $d_l d_{l-1} \cdots d_0$ を「符号付き二進表現」と定義し，単に「SD 表現」とも呼ぶ．SD は signed digit の略である．SD 表現は一意ではなく，l も変化する（例：$n = 3 = 10\bar{1} = 1\bar{1}0\bar{1}$）．

© 2009 Information Processing Society of Japan

Algorithm 1 E_{ss} での Miller アルゴリズム

入力：素数 $p \geq 5$，GF(p) 上の楕円曲線 $E_{ss}: y^2 = x^3 + x$，
E_{ss} の群位数 $p+1$ の最大素因数 $r = (1, r_{l-2}, \ldots, r_0)_2$，
位数 r の部分群 $E_{ss}(\mathrm{GF}(p))[r]$ に属する 2 点 P，Q，
埋め込み次数 $k = 2$

出力：ペアリング値 $e(P, \phi(Q)) \in \mathrm{GF}(p^k)^*$

1: $f_1 \leftarrow 1$，$f_2 \leftarrow 1$，$T \leftarrow P$
2: $Q \leftarrow \phi(Q)$
3: **for** $i = l-2$ **downto** 0 **do**
4: 　$f_1 \leftarrow f_1^2\ l_{T,T}(Q)$，　$f_2 \leftarrow f_2^2\ v_{[2]T}(Q)$
5: 　$T \leftarrow [2]T$
6: 　**if** $r_i = 1$ **then**
7: 　　$f_1 \leftarrow f_1\ l_{T,P}(Q)$，　$f_2 \leftarrow f_2\ v_{T+P}(Q)$
8: 　　$T \leftarrow T + P$
9: **retrun** $(f_1/f_2)^{(p^k-1)/r}$

とすると，双線形性は $e([c]P, Q) = e(P, [c]Q) = e(P, Q)^c$（$c \in \mathbb{Z}$）で表される．以下ではペアリングを reduced Tate ペアリングの意味で用いる．

2.2 Miller アルゴリズム

標数 5 以上の素体上の超特異楕円曲線 E_{ss} では，ペアリングの計算には現時点では Miller アルゴリズムが用いられる．E_{ss} についての Miller アルゴリズム[5] を Algorithm 1 に示す．

Algorithm 1 のステップ 2 での ϕ は distortion map と呼ばれ，$Q = (x, y)$ を式 (2) での集合 $E(\mathrm{GF}(p^k))/rE(\mathrm{GF}(p^k))$ の元に写す．E_{ss} では $\phi(Q) = (-x, iy)$; $i^2 = -1$ となる．distortion map により入力の P，Q の両方を位数 r の部分群 $E_{ss}(\mathrm{GF}(p))[r]$ の元とすることができる．ステップ 5，8 は各々楕円曲線上の点の 2 倍算，加算を表す．また，ステップ 4 の $l_{T,T}$，$v_{[2]T}$ は各々 T を通る E_{ss} の接線，$[2]T$ を通る y 軸に平行な直線，ステップ 7 の $l_{T,P}$，v_{T+P} は各々 T と P を通る直線，$T+P$ を通る y 軸に平行な直線を表す．ステップ 9 の冪乗は最終冪を表す．なお，次章で述べるとおり，r と p のビット長はペアリング暗号の安全性条件から決まる．

Algorithm 1 は，入力として二進表現でのビット 1 の個数が最小の r を選び，ステップ 6 での分岐回数を最小化することで高速化が可能となる．さらに Algorithm 1 は，r の符号付き二進表現を用い，各項の係数 $+1$，-1 に対応した追加処理を行うアルゴリズムに拡張することができる．この拡張アルゴリズムは RFC5091 あるいはペアリング計算ライブラリの PBC[9] で使用されていることから，Miller アルゴリズムを高速化する r として，符号付き二進表現での ± 1 の係数の個数（重み）が最小の r を用いる．

Algorithm 2 Montgomery 乗算

入力：$X = xR \bmod p$，$Y = yR \bmod p$，$n = X, Y$ の最大ワード長，
$W =$ CPU のワード幅（ビット長），$p' = -p^{-1} \bmod 2^W$
（p は奇素数，R は 2 の冪乗，$x, y \in \mathrm{GF}(p)$，$R > p$）

出力：$Z = XYR^{-1} \bmod p = xyR \bmod p$

（X, Y, Z の i 番目のワードを X_i, Y_i, Z_i で表す．最下位ワードは $i = 0$）
1: $Z \leftarrow 0$
2: **for** $i = 0$ **to** $n-1$ **do**
3: 　$u = (Z_0 + X_i * Y_0) * p' \bmod 2^W$
4: 　$Z \leftarrow (Z + X_i * Y + u * p)/2^W$
5: **if** $Z \geq p$ **then** $Z \leftarrow Z - p$
6: **return** Z

2.3 Montgomery 乗算

Montgomery 乗算の計算アルゴリズム[6] を Algorithm 2 に示す．Montgomery 乗算は GF(p) 乗算の高速化で標準的に使用されている．E_{ss} を用いるペアリング計算は GF(p) 上で行われ，計算時間の大部分は GF(p) での乗算が占めることから，Montgomery 乗算の高速化によりペアリング計算全体も高速化される．

Algorithm 2 を高速化するため，p について次の Mont1–2 の条件を設定する．

Mont1: $p' = 1$ として p' の乗算を省略する（ステップ 3）．

Mont2: $u * p$ のワード単位の乗算回数を最小化する（ステップ 4）．

条件 Mont1 から $p \bmod 2^W = -1$ であり，$p = 2^v sr - 1$（s：奇数，$r : p+1$ の最大素因数）と置くと $v \geq W$．したがってほとんどの CPU にあてはまる $W \geq 4$ とすると，$p \equiv -1 \pmod 4$ となる．

条件 Mont2 については，s のビット長を最大 1 ワードとする．このとき，u が 1 ワードにより $u * p = 2^v(u * s)r - u$ でのワード単位の乗算回数は，$a := u * s$ の最大 1 回で済む．また a と r の乗算は，r が低重み（3 または 4 等）であれば a（または a の 2 の補数）のビットパターンを適切なビット位置に配置することで $a * r$ を作成できる場合が多い．したがって a 同士が占めるビット位置が重ならずに配置できる r が望ましい．

3. ペアリング暗号の安全性

楕円曲線 E_{ss} のペアリング暗号の安全性条件として，従来のビット長条件および Schirokauer 攻撃に対する安全性条件を考慮する．本章ではこれらの条件を記述するが，特に，

© 2009 Information Processing Society of Japan

Schirokauer 攻撃に対する安全性条件を考察する．

3.1 ビット長条件

本稿での楕円曲線 E_{ss} は，群位数が $p+1=2^v sr$（s：奇数，r：$p+1$ の最大素因数）であり，ペアリングの埋め込み次数は 2 である．

位数 r の部分群に属する点をペアリングの入力とすることから，ECDLP に対する安全性は r のビット長で評価される．また，埋め込み次数 2 によりペアリングの出力が $\mathrm{GF}(p^2)^*$ に属することから，DLP に対する安全性は位数 p^2 のビット長で評価される．

NIST によるこれらのビット長の推奨値[14] は，現時点では ECDLP について 160 ビット以上，DLP について 1024 ビット以上である．したがって r のビット長は 160 以上，p のビット長は 512 以上となる．

3.2 Schirokauer 攻撃の概要

素因数分解問題の解法としての数体篩法（NFS）は，DLP の解法としても用いることができる[16]．Schirokauer 攻撃は NFS を用いた DLP の解法の 1 つであり，NFS で必要となる smooth な数を選び出す集合の位数の上界を，従来手法による上界よりも引き下げる手法である．

奇数 N を DLP のモジュラスとし，N の符号付き二進展開（SD 展開）を以下で表す．

$$N=\epsilon_w 2^{c_w}+\cdots+\epsilon_1 2^{c_1}\ . \tag{3}$$

ここで $c_w>c_{w-1}>\ldots>c_1=0$; $\epsilon_i\in\{1,\ -1\}$ であり，N の全ての SD 展開の中で最小の w を重みと呼ぶ．以下では最初に N を素数として素体上の Schirokauer 攻撃の計算量について述べ，最後の段落で本稿の対象である 2 次拡大体についての計算量が，素体上の計算量から求められることを述べる．

正整数 e を与えて冪指数 c_i（$i=1,\ldots,w$）についての剰余 $\bar{c}_i\ :=c_i \bmod e$ をとる．これらの $\bar{c}_i$ と e を非減少順に並べ替えた列

$$\hat{c}_1=0\le\hat{c}_2\le\ \cdots\le\hat{c}_w<\hat{c}_{w+1}=e \tag{4}$$

における最大間隙，すなわち隣接する値同士の差の絶対値の最大値を γ とする．列の中に e が存在することから $\gamma>0$ である．次に，DLP を解くために必要となる smooth な数の候補（smoothness candidate）を生成する整数係数多項式 $f(x)$ の次数を d とする．

以上の e, γ, d が Schirokauer 攻撃での主要なパラメータとなる．これらを用いて，Schirokauer 攻撃では smoothness candidate の個数の上界 $U_{Sch}(d)$ が次式で表され，これが攻撃のための計算量（攻撃計算量）となる．

$$U_{Sch}(d)=2dM_1^{d+1}2^{2e-\gamma}. \tag{5}$$

ここで $M_1>0$ は smoothness candidate を集める領域を決めるパラメータであり，d の関数となる．M_1 は閉形式では与えられないが，条件

$$M_1u^{-u}\ge\pi^2(d+1)/12$$

を満たす最小の M_1 として計算される．ここで $u=(\log\ z)/(\log M_1)$，$z=U_{Sch}(d)$ である．また，e の範囲は $[\lfloor c_w/(d+1)\rfloor,\ \lfloor c_w/(d-1)\rfloor]$ で与えられ，式 (5) の計算には右辺の指数 $2e-\gamma$ を最小化する e が用いられる．

以上から，$U_{Sch}(d)$ が従来の手法より小さければ DLP に対する新たな攻撃方法であるといえる．Schirokauer 攻撃が対象とする従来手法は Joux-Lercier 攻撃[8] であり，この場合の smoothness candidate の個数の上界 $U_{JL}(d)$ は次式で表される．

$$U_{JL}(d)=((d+4)(d+2)/4)M_2^{d+1}N^{2/(d+2)}\ . \tag{6}$$

ここで M_2 は $U_{Sch}(d)$ の M_1 と同様に計算される．ただし $z=U_{JL}(d)$ とする．

なお本稿の E_{ss} は埋め込み次数が 2 であることから，DLP の対象は 2 次拡大体 $\mathrm{GF}(p^2)$ となるが，この場合の $U_{Sch}(d)$, $U_{JL}(d)$ は式 (5)，(6) の各右辺について M_i^{d+1}（$i=1,2$）以外を 2 乗した値

$$U_{Sch}(d)=4d^2M_1^{d+1}2^{2(2e-\gamma)},\ \ U_{JL}(d)=((d+4)(d+2)/4)^2M_2^{d+1}(N^2)^{2/(d+2)} \tag{7}$$

が用いられる．

3.3 Schirokauer 攻撃に対する安全性

Schirokauer 攻撃と Joux-Lercier 攻撃は U_{Sch} と U_{JL} の小さい方が攻撃手法として優れているが，これらは d の関数であるため d により大小関係が入れ替わる可能性もある．そのため，与えられた範囲の d における最小計算量を各々$\min_d\{U_{Sch}\}$，$\min_d\{U_{JL}\}$ とし，Schirokauer 攻撃に対する安全性を次式で定義する．

$$\min_d\{U_{Sch}\}\ge\min_d\{U_{JL}\}\ . \tag{8}$$

式 (8) の安全性は，Schirokauer 攻撃と Joux-Lercier 攻撃との相対的な安全性であるが，DLP の解法として漸近的に最も高速なアルゴリズムは数体篩法（NFS）であり，Joux-Lercier 攻撃を考慮しても素因数分解に NFS を用いる場合と計算量はほぼ同等とされる[16]．したがって p^2 が素因数分解に安全なビット長を持ち，式 (8) を満たす p は，Joux-Lercier 攻撃を考慮した DLP と Schirokauer 攻撃とに対して安全といえる．d の範囲の設定については

© 2009 Information Processing Society of Japan

次節で考察する．

3.4　次数 d の範囲の設定

Schirokauer 攻撃では，攻撃のための時間計算量を最小化する次数 d が次式で表される．

$$d = \left(\frac{(3\theta + o(1)) \log N}{2 \log\log N} \right)^{1/3}. \tag{9}$$

ここで $\theta = (2w-3)/(w-1)$．log は自然対数．

式 (9) により 512 ビットの N での d の範囲を求めると，$o(1) = 0$ として，$\theta \in [1,\ 2)$ ($w \in [2,\ \infty)$) に対応して $d \in [4.49,\ 5.66)$ となる．したがって最小計算量を与える d は 5 となるが，式 (9) は漸近値であるため個々の N については $d = 5$ から外れる可能性もある．そのため d の範囲として $[2,\ d_{max}]$; $d_{max} = 10$ に設定する．なお，最小値の $d = 2$ は e の範囲が $[\lfloor c_w/(d+1) \rfloor,\ \lfloor c_w/(d-1) \rfloor]$ であること（3.2 節）による．

4. 提 案 手 法

本章ではペアリング計算を効率化し，かつ Schirokauer 攻撃を含めて安全な素数 $p = 2^v sr - 1$ の探索手法を提案する．

4.1　効率性と安全性の条件

前章で導出した効率性と安全性の各条件を以下にまとめる．E は効率性の条件，S は安全性の条件を表す．

E1： 素数 r は符号付き二進表現での重みが最小（2.2 節）
E2： 指数 v の値はワード幅（ビット長）以上（2.3 節）
E3： 奇数 s のビット長は 1 ワード以下（2.3 節）
E4： r は符号付き二進表現での隣接する 2 の冪指数の差が $u * s$ のビット長以上（2.3 節）
S1： r のビット長 $= 160$ または 161，p のビット長 $= 512$（3.1 節）
S2： $d \in [2,\ d_{max}]$ の範囲で $\min_d\{U_{Sch}\} \geq \min_d\{U_{JL}\}$（3.3 節）
S3： $d_{max} = 10$（3.4 節）

上記において，E1 は Miller アルゴリズムの効率化条件，E2–E4 は Montgomery 乗算の効率化条件，E4 は $u * p$ の計算で $u * s$ がオーバラップしないための条件に対応し，S1 は ECDLP，DLP に対する従来の安全性条件，S2 は Schirokauer 攻撃に対する安全性条件に対応する．S3 は S2 での次数の上限値となる．なお r については条件 E1 と S1 を満たす重み 2 の素数（Mersenne 数または Fermat 数）は存在しないことから，条件 E1 で可能性のある最小重みは 3 となる．以下，r の重みは 3 とする．

上記の各条件は個別に導出されたものであるが，実際には効率性と安全性は相互に関連する．特に，s のビット長は効率性と Schirokauer 攻撃に対する安全性の両立に関して重要な値であり，4.2 節で考察する．また条件 S2 は p の全ての SD 表現での γ について計算する必要があり，1.1 節で述べたとおり現実的ではないため，実際には計算可能な条件に変更する．この変更方法を 4.3 節で述べる．これらの考察の後に，具体的な探索アルゴリズムを提案する．

4.2　s のビット長

素数 $p = 2^v sr - 1$ において，効率性条件 E3，E4 と Schirokauer 攻撃に対する安全性とは s のビット長について定性的なトレードオフの関係がある．

効率性のため r が低重みで $s = 1$ とすると p も低重みとなり，Schirokauer 攻撃に対して不利となる．他方，$s > 1$ であれば p の重みは s の重みによって増加するため Schirokauer 攻撃に対しては有利となるが，s が大であれば条件 E3，E4 に対しては不利となる．しかしながら，ソフトウェア実装では実際の計算はワード単位で行われるため，$s > 1$ のビット長が 1 ワード以下であり特殊な形を想定しなければ，条件 E3 での効率性は s のビット長によらない．したがって p の探索では，s のビット長は 1 ワード以下でできるだけ長く設定することが妥当と考えられる．

4.3　安全性条件 S2

素数 p の各 SD 表現について定まる γ の共通の上界を γ_{ubound} とする．γ，γ_{ubound} は e の関数であるから，任意の e について

$$\gamma_{ubound}(e) \geq \max\{\ p \text{ の各 SD 表現について定まる } \gamma(e) \text{ の全ての集合 }\}$$

である．

$2e - \gamma(e) \geq 2e - \gamma_{ubound}(e)$ により，$2e - \gamma_{ubound}(e)$ を最小化する $e =: e_1$ について $U_{Sch}(\gamma(e)) \geq U_{Sch}(\gamma_{ubound}(e_1))$ となる（詳細は付録 A.2 に記述）．したがって $\gamma_{ubound}(e_1)$ を用いて $\min_d\{U_{Sch}\}$ を計算し，$\min_d\{U_{Sch}\} \geq \min_d\{U_{JL}\}$ を判定する．以下，γ_{ubound} の計算のための準備と γ_{ubound} を与える命題を述べる．

最初に，非負整数 n の通常の二進表現において，0 と 1 の間（および最左ビットの左側と最右ビットの右側）に記号 | を挿入することで連続した 0 の列と連続した 1 の列とが交互に並ぶように表し，これを n のブロック分割と定義する．例として $n = 1913 = (11101111001)_2$ の場合には，n のブロック分割は |111|0|1111|00|1| と表される．ここで記号 | に挟まれた 0 の列 $|0\ldots0|$ および 1 の列 $|1\ldots1|$ をブロックと定義し，ブロック内のビットの個数をブ

© 2009 Information Processing Society of Japan

ロックの長さと定義する．

次に，条件 E4 により p の二進表現は，r の各項の符号に応じて以下のいずれかで表される．r の重みは 3 とする．

$(s)_2\ 0\ldots0\ (s)_2\ 0\ldots0\ (s-1)_2\ 1\ldots1$,

$(s-1)_2\ 1\ldots1\ (s')_2\ 0\ldots0\ (s-1)_2\ 1\ldots1$,

$(s)_2\ 0\ldots0\ (s-1)_2\ 1\ldots1\ (s'-1)_2\ 1\ldots1$,

$(s-1)_2\ 1\ldots1\ (s'-1)_2\ 1\ldots1\ (s'-1)_2\ 1\ldots1$.

ここで s' は s の 2 の補数を表す．また，$(s')_2$, $(s'-1)_2$ は先頭に 0 の列を付加して $(s)_2$ と同一ビット長とした列とする．$(s)_2$, $(s-1)_2$ には 0 の列は付加しない．

s は奇数，s', $s'-1$ は各々$s-1$, s のビット反転により，p の上記 4 種類の二進表現のブロック分割は次式で表される．

$$|\ s_3\ |\ g_3\ |\ s_2\ |\ g_2\ |\ s_1\ |\ g_1\ |\ . \tag{10}$$

ここで s, $s-1$, s', $s'-1$ はこれら自体のブロック分割を省略して $|s_i|$（$i=1,2,3$）で表し，各 $|s_i|$ を 1 つのブロックと見なす．したがって $|s_i|$ には 0，1 の両方が含まれていてもよい．また $|g_i|$ は $|0\ldots0|$ または $|1\ldots1|$ であり，長さ 0 の場合を含む．

以上の準備により次の命題 1 が成り立ち，γ_{ubound} が求まる．証明は付録 A.1 に記述する．

命題 1　γ_{ubound} は次式で表される．

$$\gamma_{ubound} = \max\{\ \gamma_r,\ \gamma_s\ \}\ . \tag{11}$$

ここで γ_r, γ_s は各々以下の (a)，(b) で定義される．

(a) γ_r：式 (10) において $|g_3|=|0\ldots0|$，$|g_2|=|0\ldots0|$，$|g_1|=|0\ldots0|1|$，$|s_i|=|1\ldots1|$（$i=1,2,3$）と置き換えて計算した γ（いずれの置き換えでも長さは $|g_i|$，$|s_i|$ の長さと等しくとる）．

(b) γ_s：$(s)_2$, $(s-1)_2$ の内部の 0 の列の長さの最大値を各々$L_s(0)$, $L_{s-1}(0)$，1 の列の長さの最大値を各々$L_s(1)$, $L_{s-1}(1)$ とするとき，

$\gamma_s := \max\{\ L_s(0),\ L_{s-1}(0),\ L_s(1),\ L_{s-1}(1)\ \}+1$.

4.4　探索アルゴリズム

提案する探索アルゴリズムを Algorithm 3 に示す．Algorithm 3 は効率性条件 E1–E4 を満たす素数の探索部分と，見い出された素数について安全性条件 S2 を検証する部分で構成される．安全性条件 S1，S3 は入力で与える．また今回は各条件を満たす素数を 1 つ見い出すことを目標としたため，このアルゴリズムは素数 p が最初に見い出された時点で終了するが，見い出す個数を指定するための変更は容易である．

Algorithm 3　提案探索アルゴリズム

入力：r のビット長 l_r，p のビット長 l_p，ワード幅 W，
　　r の最大重み $wmax$，最大次数 d_{max}（3.4 節）
出力：効率性条件 E1–E4 と安全性条件 S1–S3 を満たす素数 p
　　または “Not found”

/* ステップ 1–15 は効率性条件を満たす p の探索 */
1: $w \leftarrow 2$
2: **while** $w \le wmax$ **do**
3:　重み w で l_r ビットの奇数 r を新規に生成.
　　生成できなければ $w \leftarrow w+1$ としてステップ 2 へ
4:　r が合成数であればステップ 3 へ
5:　$l_s \leftarrow W$
6:　**while** $l_s \ge 1$ **do**
7:　　$l_s > 1$ と r の組が条件 E4 を満たさなければ $l_s \leftarrow l_s - 1$
　　としてステップ 6 へ
8:　　$l_s = 1$ と r の組が条件 E4 を満たさなければステップ 3 へ
9:　　$v \leftarrow\ l_p - l_r - l_s$
10:　　$v \ge W$ でなければ $l_s \leftarrow\ l_s - 1$ としてステップ 6 へ
11:　　ビット長 l_s の奇数 s をランダムに生成.
　　生成できなければ $l_s \leftarrow\ l_s - 1$ としてステップ 6 へ
12:　　$p \leftarrow 2^v sr - 1$ が素数であればステップ 16 へ
13:　　$l_s \leftarrow l_s - 1$
14:　$w \leftarrow w+1$
15: “Not found” を出力し終了
/* ステップ 16–25 は安全性条件の検証 */
16: V_{JL}, $V_{Sch} \leftarrow$ 100000（十分大きな値）
17: **for** $d=2$ **to** d_{max} **step** 1 **do**
18:　$tmp \leftarrow$ 100000（$2c_w$ より大きな値）
19:　**for** $e_{tmp} = \lfloor c_w/(d+1) \rfloor$ **to** $\lfloor c_w/(d-1) \rfloor$ **step** 1 **do**
20:　　γ_{ubound} を計算（式 (11)）
21:　　**if** $2e_{tmp} - \gamma_{ubound} < tmp$ **then**
　　$e \leftarrow e_{tmp}$, $tmp \leftarrow 2e_{tmp} - \gamma_{ubound}$
22:　U_{Sch}, U_{JL} を計算（式 (7)）
23:　**if** $U_{JL} < V_{JL}$ **then** $V_{JL} \leftarrow U_{JL}$
24:　**if** $U_{Sch} < V_{Sch}$ **then** $V_{Sch} \leftarrow U_{Sch}$
25: **if** $V_{Sch} < V_{JL}$ **then** ステップ 11 へ　**else** p を出力し終了

5.　素数 p の探索結果と実装データ

本章では効率性と安全性を満たす素数 p の実例を探索し，見い出された素数を用いた Miller アルゴリズムの高速化効果を Pentium 4 での実装により示す．探索で見い出された素数を

© 2009 Information Processing Society of Japan

提案素数と呼ぶ.

5.1　p の探索結果

$l_r = 160$ または 161，$l_p = 512$，$W = 32$，$wmax = 3$，$d_{max} = 10$ として探索を実施した結果得られた提案素数の例を以下に示す．この素数を p_{Sch1} で表す．なお (x) を 2^x の略記とし，`0x` 以降の表記は 16 進数を表す.

$r = (160) + (110) - 1$，　$s =$ `0x93c02bdd`，　$v = 320$,

$p_{Sch1} = (480)s + (430)s - (320)s - 1$

$= $ `0x93c02bdd 000024f0 0af73fff ffffffff ffffffff 6c3fd422`
`ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff`
`ffffffff ffffffff ffffffff ffffffff`.

p_{Sch1} は Schirokauer 攻撃に対する安全性条件 S2 （$\min_d\{U_{Sch}\} \geq \min_d\{U_{JL}\}$）を満たす.

一方，$\min_d\{U_{Sch}\} < \min_d\{U_{JL}\}$ である素数の例 p_{Sch2} を以下に示す.

$r = (160) - (91) - 1$，　$s =$ `0xe60d248f`，　$v = 320$,

$p_{Sch2} = (480)s - (411)s - (320)s - 1$

$= $ `0xe60d248f ffffffff f8cf96db 87ffffff ffffffff 19f2db70`
`ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff`
`ffffffff ffffffff ffffffff ffffffff`.

p_{Sch1}，p_{Sch2} に対する Schirokauer 攻撃と Joux-Lercier 攻撃との比較を**図 1** に示す．図 1 は次数 d に対する各攻撃の計算量のビット長をグラフ表示している．p_{Sch1} への Schirokauer 攻撃の計算量は $d \leq 7$ において Joux-Lercier 攻撃の計算量を上回っており，最小計算量同士（Schirokauer 攻撃：$d = 5$ での計算量，Joux-Lercier 攻撃：$d = 4, 5$ での計算量）においても Schirokauer 攻撃の計算量が上回る．対して p_{Sch2} では $d = 5, 6$ 以外で Schirokauer 攻撃の計算量が Joux-Lercier 攻撃の計算量を上回っているが，双方が最小計算量となる $d = 5$ では逆に Schirokauer 攻撃の計算量が Joux-Lercier 攻撃の計算量を下回る．これらの結果により，ペアリングに用いる素数として p_{Sch1} を採用し，p_{Sch2} は棄却する.

5.2　実装データ

以下では Algorithm 2 の Montgomery 乗算を p_{Sch1} に用いた場合と，p_{Sch1} に特化した高速化 Montgomery 乗算を用いた場合のペアリング計算速度の測定結果を示す.

（1）速度測定の条件

Miller アルゴリズムの実装のベースとしてペアリング計算用の PBC ライブラリ[9]（v0.4.18）を使用し，速度測定ツールとして PBC ライブラリに付属のベンチマーク用コード（`benchmark.c`）を使用した．Montgomery 乗算については，高速化 Montgomery 乗算を PBC ライブラリ内部に実装し，通常の Montgomery 乗算は PBC ライブラリに既存のアルゴリズム（Algorithm 2 と同一）を使用した.

また，PBC ライブラリに必要な多倍長演算ライブラリとして GMP[4]（v4.2.2）を使用した．CPU は 2.8 GHz Pentium 4 を用いたが，組み込み用途向け等 Pentium 4 以外の CPU 環境でも高速化の効果を類推可能とするため，GMP ライブラリは generic 指定でインストールした．この指定により，SSE 等 Pentium 4 固有の高速演算機能を用いず，一般的な CPU が持つ機能に対応した命令のみ使用する．なお主記憶容量は 512 MB，OS は Red Hat Linux 9 であり，コンパイルと最適化は `g++ -O2` で実施した.

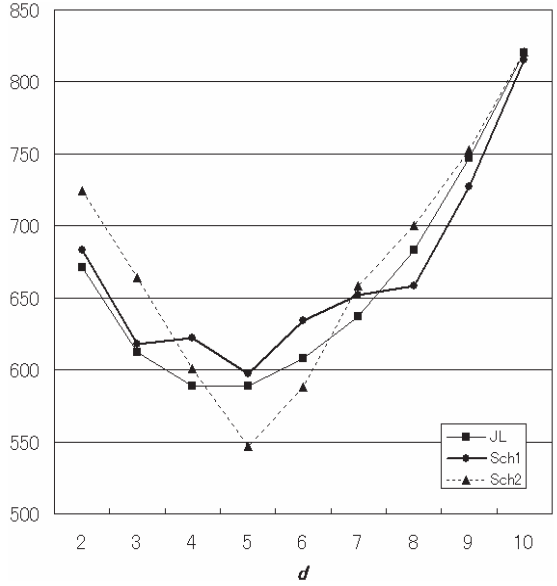


図 1　Schirokauer 攻撃に対する安全性
Fig. 1　Security against Schirokauer's attack.

d：多項式の次数（3.2 節），縦軸：攻撃計算量のビット長，JL：Joux-Lercier 攻撃，Sch1：p_{Sch1} に対する Schirokauer 攻撃，Sch2：p_{Sch2} に対する Schirokauer 攻撃
JL との最小計算量同士（$d = 5$ での値）の比較により Sch1 は安全，Sch2 は安全でない

© 2009 Information Processing Society of Japan

表 1　探索結果の p_{Sch1} を用いた Miller アルゴリズムの計算速度
Table 1　Timings of Miller algorithm with found prime p_{Sch1}.

	前処理なし	前処理あり
従来手法（PBC ライブラリ[9]）	55.1 msec	23.4 msec
提案手法	43.1 msec	18.3 msec

CPU：2.8 GHz Pentium 4

（2）速度測定結果

p_{Sch1} を用いたペアリング計算速度の測定結果を**表 1** に示す．各計算時間は 10 万回のペアリング計算の平均値を表す．表中の前処理は，PBC ライブラリにおいて Miller アルゴリズム内で必要なパラメータ（直線の係数）をあらかじめ計算する処理を表す．

表 1 に示されるとおり，提案手法では高速化 Montgomery 乗算を用いることにより，従来手法に対して約 22%のペアリング計算の高速化を達成した．

6. ま と め

標数 5 以上の素体上の超特異楕円曲線でのペアリングを対象として，ペアリング計算の高速化と Schirokauer 攻撃に対する安全性とを実現するための素数探索手法を提案した．探索手法で見い出される提案素数は，ペアリング計算での Miller アルゴリズムとその内部の Montgomery 乗算を高速化し，かつ Schirokauer 攻撃に対する安全性を持つ．提案素数の探索に要する時間は十分短く，高速性と安全性が両立する素数を現実的な時間内で見い出すことが可能となる．提案素数を用いた実装の結果，Miller アルゴリズムの計算速度について約 22%の高速化を実現した．この結果はソフトウェア実装によるものであるが，CPU に対して特殊な命令を前提にしてはいないため，提案手法による高速化は組み込み用途向けを含めた広範囲の CPU に適用可能である．また，本稿では Schirokauer 攻撃に安全な素数の探索アルゴリズムを提案したが，式 (9) で与えられる d の値は漸近的な値であるため，今後の解析の進展により d の最適値が変わる可能性もある．しかし提案探索アルゴリズムを用いれば，そのような変化にも容易に対応可能である．

謝辞　初期投稿版について有益なコメントをいただきました査読者の方々に感謝いたします．

参 考 文 献

1) Barreto, P.S.L.M., Galbraith, S.D., Ó'hÉigeartaigh, C. and Scott, M.: Efficient Pairing Computation on Supersingular Abelian Varieties, *Designs, Codes and Cryptography*, Vol.42, No.3, pp.239–271 (2007).
2) Boyen, X. and Martin, L.: Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems, Internet Engeneering Task Force (online). available from http://tools.ietf.org/html/rfc5091 (accessed 2008-10-22)
3) Beuchat, J.-L., Shirase, M., Takagi, T. and Okamoto, E.: An Algorithm for the η_T Pairing Calculation in Characteristic Three and its Hardware Implementation, *18th IEEE International Symposium on Computer Arithmetic*, *ARITH-18*, pp.97–104 (2007).
4) Free Software Foundation: GNU Multiple Precision Arithmetic Library, FSF (online). available from http://gmplib.org/ (accessed 2008-10-22)
5) Galbraith, S.: Pairings, Blake, F., Seroussi, G. and Smart, N.P. (Eds.), *Advances in Elliptic Curve Cryptography*, LMS 317, Cambridge Univ. Press (2005).
6) Menezes, A.J., Van Oorschot, P.C. and Vanstone, S.A.: *Handbook of Applied Cryptography*, CRC Press (1996).
7) Hess, F., Smart, N.P. and Vercauteren, F.: The Eta Pairing Revisited, *IEEE Trans. Information Theory*, Vol.52, No.10, pp.4595–4602 (2006).
8) Joux, A. and Lercier, R.: Improvement to the General Number Field Sieve for the Discrete Logarithms in Prime Finite Field, *Math. Comp.*, Vol.72, pp.953–967 (2003).
9) Lynn, B.: The Pairing-Based Cryptography Library, Stanford Univ. (online). available from http://crypto.stanford.edu/pbc/ (accessed 2008-10-22)
10) Miller, V.S.: Short Programs for Functions on Curves, Unpublished manuscript (1986), Stanford Univ. (online). available from http://crypto.stanford.edu/miller/miller.pdf (accessed 2008-10-22)
11) Montgomery, P.: Modular Multiplication without Trial Division, *Math. Comp.*, Vol.48, pp.519–521 (1985).
12) Nakajima, T., Izu, T. and Takagi, T.: Reduction Optimal Trinomials for Efficient Software Implementation of the η_T Pairing, *Trans. IEICE*, Vol.E91-A, No.9, pp.2379–2386 (2008).
13) 中島俊哉，伊豆哲也，高木　剛：GF(p) 上の超特異楕円曲線におけるペアリング計算に適した素数の検討，2008 年暗号と情報セキュリティシンポジウム予稿集，3D1-2 (2008).
14) National Institute of Standard and Technology: Recommendation for Key Man-

© 2009 Information Processing Society of Japan

agement, SP 800-57 (2007), NIST (online). available from http://csrc.nist.gov/publications/PubsSPs.html (accessed 2008-10-22)
15) Schirokauer, O.: The Number Field Sieve for Integers of Low Weight, *IACR Cryptology ePrint Archive*, 2006/107 (2006), IACR (online). available from http://eprint.iacr.org/2006/107.pdf (accessed 2008-10-22)
16) 内山成憲 : 離散対数問題の困難性に関する計算量についての調査・研究報告書，Cryptography Research and Evaluation Committees (CRYPTREC)，技術報告書 No.0602 (2007)，CRYPTREC (オンライン)．入手先 http://www.cryptrec.go.jp/estimation.html (参照 2008-10-22)
17) Yoshitomi, M., Takagi, T., Kiyomoto, S. and Tanaka, T.: Efficient Implementation of the Pairing on Mobilephones using BREW, *Trans. IEICE*, Vol.E91-D, No.5, pp.1330–1337 (2008).

付　　録

A.1　命題 1 の証明

本節では 4.3 節の命題 1 の証明を記述する．なお，4.3 節で述べた記号およびブロックに関する定義を以降でも用いる．

命題 1　γ_{ubound} は次式で表される．

$\gamma_{ubound} = \max\{\ \gamma_r,\ \gamma_s\ \}$.

ここで γ_r, γ_s は各々以下の (a)，(b) で定義される．

(a) γ_r：式 (10) において $|g_3| = |0\dots0|$，$|g_2| = |0\dots0|$，$|g_1| = |0\dots0|1|$，$|s_i| = |1\dots1|$ ($i = 1, 2, 3$) と置き換えて計算した γ（いずれの置き換えでも長さは $|g_i|$，$|s_i|$ の長さと等しくとる）．

(b) γ_s：$(s)_2$，$(s-1)_2$ の内部の 0 の列の長さの最大値を各々$L_s(0)$，$L_{s-1}(0)$，1 の列の長さの最大値を各々$L_s(1)$，$L_{s-1}(1)$ とするとき，

$\gamma_s := \max\{\ L_s(0),\ L_{s-1}(0),\ L_s(1),\ L_{s-1}(1)\ \} + 1$.

命題 1 の証明　$G := 1$ または -1，$B := 0$ とする．任意の非負整数 n の二進表現のブロック分割において，ブロック内の各ビットを $\{0,\ 1,\ \bar{1}\}$ のいずれか任意の元で置き換える ($\bar{1} := -1$)．記号 | は変更せずにそのまま残す．

n の全てのビットを置き換えた後，| を除く 0，1，$\bar{1}$ の列が n の SD 表現であるならば，次の (C1)，(C2) が成り立つ．

(C1)　| 前後の列 $U|V$（U, $V = G$ または B）は $B|B$ ではない．

(C2)　各ブロック内のビットを置き換えた列は，ブロック両端の | を含めて

$$|B\dots B|,\ |B\dots BG|,\ \dots,\ |BG\dots G|,\ |G\dots G| \tag{12}$$

のいずれかである．

(C1)，(C2) は以下により示される．

最初に，与えられた非負整数 n と任意の非負整数 a についての恒等式 $n = (n+a) - a$ により，n の SD 表現 n_{SD} は次の式で表される．

$$n_{SD} = (n+a)_2 \mathbin{\overline{\oplus}} (a)_2\ . \tag{13}$$

ここで $\overline{\oplus}$ は左右の通常の二進表現（先頭に適宜 0 を追加してビット長を等しくする）の各ビットについて以下の演算を行う記号を表す．

$0 \mathbin{\overline{\oplus}} 0 = 0$，$1 \mathbin{\overline{\oplus}} 0 = 1$，$0 \mathbin{\overline{\oplus}} 1 = \bar{1}$，$1 \mathbin{\overline{\oplus}} 1 = 0$.

式 (13) が任意の $a \geq 0$ について n の 1 つの SD 表現を与えることは明らかであり，逆に n の任意の SD 表現について，1 を 0 に置き換えた表現での値の絶対値として a が定まる．

次に，n のビット長を $l+1$ ビットとし，n のブロック分割において，| の左側のビットは n の二進表現で第 i ビット，右側のビットは第 $i-1$ ビットであるとする（$i = 1, \dots, l$）．上記の恒等式 $n = (n+a) - a$ において，$(n)_2 + (a)_2$ の加算で第 i ビットに渡されるキャリーを c_i，第 $i-1$ ビットに渡されるキャリーを c_{i-1} とすると，| 前後の G, B は次のとおりに分類される（n, a の第 i ビットを n_i, a_i で表し，X, Y は 0 または 1，$\hat{X}$, $\hat{Y}$ は各々X, Y のビット反転，$\sigma_i := c_i \oplus n_i \oplus a_i$，$Z_i := \sigma_i \mathbin{\overline{\oplus}} a_i$ とする．第 $i-1$ ビットについても同様）．

$n_i \mid n_{i-1} = 0 \mid 1$ の場合：

$c_i \mid c_{i-1}$	$1 \mid 1$	$0 \mid 0$	$1 \mid 0$	$0 \mid 1$
$n_i \mid n_{i-1}$	$0 \mid 1$	$0 \mid 1$	$0 \mid 1$	$0 \mid 1$
$a_i \mid a_{i-1}$	$X \mid Y$	$X \mid Y$	$X \mid Y$	$X \mid Y$
$\sigma_i \mid \sigma_{i-1}$	$\hat{X} \mid Y$	$X \mid \hat{Y}$	$\hat{X} \mid \hat{Y}$	$X \mid Y$
$Z_i \mid Z_{i-1}$	$G \mid B$	$B \mid G$	$G \mid G$	$B \mid B$

したがって $Z_i \mid Z_{i-1} = B \mid B$ であるためには $c_{i-1} = n_{i-1} = 1$ かつ $c_i = 0$ が必要であるがこれは不可能．

$n_i \mid n_{i-1} = 1 \mid 0$ の場合：

$c_i \mid c_{i-1}$	$0 \mid 0$	$1 \mid 1$	$0 \mid 1$	$1 \mid 0$
$n_i \mid n_{i-1}$	$1 \mid 0$	$1 \mid 0$	$1 \mid 0$	$1 \mid 0$
$a_i \mid a_{i-1}$	$X \mid Y$	$X \mid Y$	$X \mid Y$	$X \mid Y$
$\sigma_i \mid \sigma_{i-1}$	$\hat{X} \mid Y$	$X \mid \hat{Y}$	$\hat{X} \mid \hat{Y}$	$X \mid Y$
$Z_i \mid Z_{i-1}$	$G \mid B$	$B \mid G$	$G \mid G$	$B \mid B$

© 2009 Information Processing Society of Japan

Algorithm A1 γ の計算アルゴリズム

入力：SD パターン，正整数 e（3.2 節）
出力：γ（3.2 節式 (4) の列の最大間隙）

1：SD パターンを右端から e 個ずつ区切る．
　SD パターンの長さが e の倍数でなければ左端に B の列を追加し，
　長さを e の倍数とした後に区切る．
　これらの区切りを $K_1, \ldots, K_\mu$ とする．
2：$K_1, \ldots, K_\mu$ について G を 1 と見なし，
　並列に論理和を取った列 $K_\gamma := K_1 \vee \ldots \vee K_\mu$ を作る．
3：K_γ の中の最長の B の列の長さ $+1$ を γ として出力する．

したがって $Z_i \mid Z_{i-1} = B \mid B$ であるためには $c_{i-1} = n_{i-1} = 0$ かつ $c_i = 1$ が必要であるがこれは不可能．

以上により (C1) が成り立つ．他方，第 i, $i-1$ ビットがブロックの内部にある場合は $n_i n_{i-1} = 00$ または 11 であるから，$Z_{i-1} = B$ のとき Z_i の G, B は次のとおりに分類される．

c_i c_{i-1}	1 1	0 0	0 1	1 0
n_i n_{i-1}	1 1	0 0	1 1	0 0
a_i a_{i-1}	X Y	X Y	X Y	X Y
σ_i σ_{i-1}	X Y	X Y	$\hat{X}$ Y	$\hat{X}$ Y
Z_i Z_{i-1}	B B	B B	G B	G B

したがって $Z_i Z_{i-1} = GB$ となるためには，「$c_{i-1} = n_{i-1} = 0$ かつ $c_i = 1$」または「$c_{i-1} = n_{i-1} = 1$ かつ $c_i = 0$」が必要であるがこれは不可能．よって (C2) が成り立つ．

以上における，n のビットの置き換え後の B, G の列 $Z_l Z_{l-1} \cdots Z_0$ を SD パターンと呼ぶ．SD パターンについて γ を求めるアルゴリズムを Algorithm A1 に示す．Algorithm A1 は，3.2 節式 (4) の最大間隙としての γ を求めるアルゴリズムである．

Algorithm A1 の実行後の B, G の列 (K_γ) の中の最長の B の列を B_γ とする．Algorithm A1 の論理和演算と (C1) により，B_γ は式 (10) のいずれかのブロック内での，ある B の列 B_0 の部分列である．したがって非負整数 n についての γ と，n の二進表現内の 0 の列の最大長 $L_n(0)$ および 1 の列の最大長 $L_n(1)$ との間には

$$\gamma \le \max\{\ L_n(0),\ L_n(1)\ \} + 1 \tag{14}$$

の関係が成り立つ．また．B_γ と B_0 の関係を $B_\gamma = T(B_0)$ で表す．B_0 は γ が求まるまで事前には特定されないが，長さ 0 の場合を含め存在する．

次に，B_γ の導出において B_0 と論理和演算される列が m 個存在するとし，それらの列を A_i（$i = 1, \ldots, m$）とする．なお，A_i が存在しない場合でも以下の議論は同様に行える．A_i は一般に B, G を含み，長さは B_0 に等しい．

A_i の部分列で長さが B_γ に等しく，B_γ が B_0 に占める位置と同一の位置の部分列を $T(A_i)$ とする．B_γ が論理和演算の結果としての B の列であることから，

$$B_\gamma = T(B_0) = T(A_1) = \cdots = T(A_m)$$

であり，$T(A_i)$ も B の列である．したがって B_γ（$= T(B_0)$）と同様に各 $T(A_i)$ も式 (10) の $|s_j|$, $|g_j|$（$j = 1, 2, 3$）のいずれか 1 つのブロックに属する．ここで $T(B_0)$, $T(A_i)$ が $|s_j|$ に属するとは，$T(B_0)$, $T(A_i)$ が $|s_j|$ の中のいずれかの B の列の部分列であることと定義し，同様に $|g_j|$ に属するとは，$|g_j|$ の B の列の部分列であることと定義する．

$T(B_0)$ が属するブロックを D_0，$T(A_i)$ が属するブロックを D_i とし，D_0, D_i の $m+1$ 個組（$m+1$-tuple）を

$$D := (D_0,\ D_1,\ \ldots\ , D_m)$$

とする．D は，要素としてのブロック $|s_j|$ の有無に応じて以下の case 1，2 に場合分けされる．

case 1： $\forall j \in \{1, 2, 3\}$, $|s_j|$ は D の要素でない．

case 2： $\exists j \in \{1, 2, 3\}$, s.t. $|s_j|$ は D の要素である．

case 2 は 1 個以上の j について成立するものとする．これらの各 case について γ の上界を求める．

case 1 では D に $|s_j|$ が含まれないことから，$|s_j| = |G \ldots G|$ と置いても γ は変わらない．したがって式 (10) で $|g_3| = |B \ldots B|$, $|g_2| = |B \ldots B|$, $|g_1| = |B \ldots BG|$（すなわち，含まれる B の個数が最大）と置き，$|s_j| = |G \ldots G|$ と置いて計算した γ は，p についての γ の上界である．なお SD 表現の長さは固定ではなく，二進表現での先頭ビットを $1\bar{1} \ldots \bar{1} = G \ldots G =: h$ に置き換えることも可能であるが，h の長さの増加に対して γ の上界は非増加であるから h には置き換えない．以上により $\gamma \le \gamma_r$ となる．

case 2 では

$$\gamma \le \max\{\ L_{|s_j|}(B) \mid j \in \{1, 2, 3\}\ \} + 1$$

が成り立つ．ここで $L_{|s_j|}(B)$ は $|s_j|$ に含まれる最長の B の列の長さを表す．$|s_j|$ は s, $s-1$, s', $s'-1$ のいずれかの SD 表現であるから，

$$\begin{aligned} &\max\{\ L_{|s_j|}(B) \mid j \in \{1, 2, 3\}\ \} \\ &\le\ \max\{\ L_s(0),\ L_s(1),\ L_{s-1}(0),\ L_{s-1}(1),\ L_{s'}(0),\ L_{s'}(1),\ L_{s'-1}(0),\ L_{s'-1}(1)\ \} \end{aligned}$$

© 2009 Information Processing Society of Japan

である．ここで $L_x(0)$, $L_x(1)$ は各々x に含まれる最長の 0 の列の長さと最長の 1 の列の長さを表す．さらに，s', $s'-1$ は各々$s-1$, s のビット反転であるから，式 (14) の関係と合わせて

$$\gamma \le \max\{\ L_s(0),\ L_s(1),\ L_{s-1}(0),\ L_{s-1}(1)\ \} + 1 = \gamma_s$$

となる．

case1，2 は D の場合を尽くしているから $\gamma \le \max\{\ \gamma_r,\ \gamma_s\ \}$ が成り立つ．よって

$$\gamma_{uboud} = \max\{\ \gamma_r,\ \gamma_s\ \}$$

を得る．　□

A.2　4.3 節での関係式の詳細

本節では 4.3 節での関係式 $U_{Sch}(\gamma(e)) \ge U_{Sch}(\gamma_{ubound}(e_1))$ を導出する．ここで e_1 は $2e-\gamma_{ubound}(e)$ を最小化する e である．

2 次拡大体の場合は $U_{Sch}(d) = (2d2^{2e-\gamma})^2 M_1^{d+1}$（式 (7)）である．$M_1$ は関係式

$$M_1 u^{-u} \ge \pi^2(d+1)/12 =: \alpha \qquad (15)$$

を満たす最小の M_1 として計算される．$u = (\log\ z)/(\log M_1)$，$z = U_{Sch}(d)$ であることから，式 (15) は

$$x - a \ge \left(\frac{b(y)}{x} + c\right)\ \log\left(\frac{b(y)}{x} + c\right) =: R(x,\ y)$$

と表される．ここで $x = \log M_1$，$a = \log\alpha$，$y = 2e - \gamma(e) > 0$，$b(y) = 2\log(2d2^y)$，$c = d+1$.

$R(x,\ y)$ は x について単調減少，y について単調増加である．したがって $y_1 \le y_2$ において，$x - a \ge R(x,\ y_1)$ を満たす最小の M_1 を M_{11}，$x - a \ge R(x,\ y_2)$ を満たす最小の M_1 を M_{12} とすると $M_{11} \le M_{12}$ となる．

一方，$2e - \gamma_{ubound}(e)$ を最小化する e を e_1，$2e - \gamma(e)$ を最小化する e を e_2 とすると，全ての e について $2e - \gamma_{ubound}(e) \le 2e - \gamma(e)$ により，$2e_1 - \gamma_{ubound}(e_1) \le 2e_2 - \gamma(e_2)$ である．さらに，$y_1 = 2e_1 - \gamma_{ubound}(e_1)$, $y_2 = 2e_2 - \gamma(e_2)$ とすると $M_{11} \le M_{12}$ である．したがって次式が成り立つ．

$$\begin{aligned}\min_e\{U_{Sch}(\gamma_{ubound}(e))\} &= (2d2^{2e_1-\gamma_{ubound}(e_1)})^2 M_{11}^{d+1} = U_{Sch}(\gamma_{ubound}(e_1)) \\ &\le (2d2^{2e_2-\gamma(e_2)})^2 M_{12}^{d+1} = \min_e\{U_{Sch}(\gamma(e))\} \\ &\le U_{Sch}(\gamma(e))\ .\end{aligned}$$

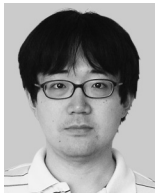
ここで $\min_e$ は e を変化させたときの最小値を表す．

以上により 4.3 節での関係式 $U_{Sch}(\gamma(e)) \ge U_{Sch}(\gamma_{ubound}(e_1))$ が得られる．

(平成 20 年 10 月 23 日受付)
(平成 21 年 4 月 3 日採録)

中島　俊哉（正会員）

1979 年東京農工大学工学部機械工学科卒業．1981 年東京大学大学院工学系研究科航空学専攻修士課程修了．1985 年東京大学大学院工学系研究科航空学専攻博士課程単位取得退学．同年富士通株式会社入社．現在に至る．2008 年より公立はこだて未来大学大学院博士（後期）課程在学中．人工知能，ニューラルネットワーク，ウェーブレット，ヒューマノイドロボット等の研究開発を経て，組み込み向け暗号システムの研究開発に従事．計測自動制御学会，日本航空宇宙学会各会員．



伊豆　哲也（正会員）

1967 年生．1992 年東京大学理学部数学科卒業．1994 年立教大学大学院理学研究科数学専攻博士前期課程修了．1997 年立教大学大学院理学研究科数学専攻博士後期課程退学．博士（工学）．1997 年より富士通株式会社および株式会社富士通研究所に勤務，現在に至る．情報セキュリティ，暗号理論の研究に従事．2001 年 Waterloo 大学（カナダ）客員研究員．1999 年暗号と情報セキュリティシンポジウム（SCIS 1999）論文賞受賞．2002 年コンピュータセキュリティシンポジウム（CSS 2002）優秀論文賞受賞．2007 年科学技術分野の文部科学大臣表彰若手科学者賞受賞．2008 年情報処理学会喜安記念業績賞受賞．電子情報通信学会，IACR 各会員．

© 2009 Information Processing Society of Japan

高木　　剛（正会員）

1993 年名古屋大学理学部数学科卒業．1995 年名古屋大学大学院理学研究科修士課程修了．同年 NTT 情報流通プラットフォーム研究所入社．2001 年理学博士（ダルムシュタット工科大学）．その後，ダルムシュタット工科大学情報科学部助教授を経て，2005 年公立はこだて未来大学システム情報科学部准教授，2008 年より同大学教授，現在に至る．2009 年より九州大学大学院数理学研究院客員教授．第 8 回船井情報科学振興賞受賞．暗号および情報セキュリティに関する研究に従事．電子情報通信学会，IACR 各会員．

© 2009 Information Processing Society of Japan