自己評価書

1 背景

楕円暗号は楕円曲線上の離散対数問題 (ECDLP) の困難さに基づく暗号である．ECDLP に対する解法として現在提案されている方法に，Pohlig-Hellman 法 [12]，Pollard-ρ 法 [13] 等の全数探索的な攻撃と，SSSA アルゴリズムと呼ばれる楕円曲線の位数の標数 p-divisible part, すなわち楕円曲線の位数の p べき部に対する攻撃 [17, 18, 1] がある．これらは，どちらも簡単に回避できる攻撃である．前者の全数探索的な攻撃は，ベースポイントの位数が大きな素数であれば回避でき，後者の SSSA 攻撃については $\mathbb{F}_p$ 上の楕円曲線ならば，トレースが 1 の場合を除けばよいので，いずれも容易に回避できる．さらに，ECDLP を k 次拡大体上の離散対数問題（DLP）に確率的多項式時間内で帰着し，DLP に対する解法を適用する FR-帰着法 [4] がある．MOV-帰着法は FR-帰着法と本質的に同じなので以下，FR-帰着法のみを述べる．現在，DLP に対する解法として提案されている最高速のアルゴリズムは数体ふるい法であるが，任意の体に適用することはできない．現実的な仮定のもとでの最高速なアルゴリズムは 2 次ふるい法で，オーダーとして $L_q[1/2, c] = exp((c + o(1))(\log q)^{1/2}(\log\log q)^{1/2})$ の準指数時間アルゴリズムとなる．このことから，拡大次数が $k > \log q$ を満たすと DLP への帰着攻撃は指数時間になり，FR-帰着法が脅威にならない [14]．したがって，曲線を構成する際には FR-帰着法が脅威にならない条件，FR-condition, i.e.

$$FR-condition : 1 \leq k \leq \log q \text{ に対して } q^k \neq 1 \pmod{n} \text{ が成り立つ}$$

ように構成する必要がある．ここで，$\mathbb{F}_q$ 上定義された楕円曲線を $E/\mathbb{F}_q$，ベースポイントを $G \in E(\mathbb{F}_q)$，その位数を $n := ord(G)$ とする．

通常，楕円曲線を構成する場合は，上で述べたような既知の攻撃に対する安全性のチェックを行う．つまり，$E/\mathbb{F}_q$，$G \in E(\mathbb{F}_q)$，$n = ord(G)$（最大素数位数）を $n \sim q$ かつ $n \neq \mathrm{ch}(\mathbb{F}_q)$，FR-condition を満たすように構成する．構成方法としては SEA[16] と呼ばれる元の個数を計算する方法，CM 法を用いる方法がある．どちらも高速に動くが，スマートカード等の小メモリサイズで実現するには何らかの工夫が必要である．通常，楕円曲線暗号を構築する際，そこで用いる楕円曲線を全ユーザで固定する場合が多い．しかし安全性の観点からは，ユーザ毎に異なる楕円曲線を用いることが望ましい．特に近年の特定の楕円曲線に対する攻撃は，全ユーザが同じ曲線を利用する危険性に対する警告ともいえる．また，近年の携帯電話をはじめとする携帯端末の普及を考えると，小メモリ，小 CPU で，楕円曲線を生成できることは，安全性，汎用性の観点からも望ましい．このような容易な楕円曲線の生成が可能になれば，ユーザ毎に異なる楕円曲線を生成する状況も十分想定できるようになるだろう．その観点からは，現在の SEA, CM-法により単純に楕円曲線を生成するには限界がある．これらの方法に，なんらかの工夫を付加する必要がある．

そこで我々は，99 年度より FR-condition について explicit な条件を与える研究を行ってきた．[9] では，FR-帰着法に対して弱い曲線の explicit な条件を示したが，今回はじめて FR-帰着法に対する安全性が保証できる explicit なトレースの条件を導き出した．本条件を用いることで FR-帰着法に対する安全性が保証された楕円曲線を容易にシステマティックに生成することができる．本提案では，この楕円曲線を用いて ECDH 鍵共有方とその一般化を実現する．

本評価書の構成は以下のとおりである．2 章で提案する基本関数である ECDLP の安全性，そして ECDH 鍵共有方及びその一般化である提案 ECDH 鍵共有方の安全性について述べる．3 章では提案楕円曲線の生成アルゴリズムについて実験的な評価を行う．4 章では，ECDH 鍵共有方及び提案鍵共有方の実装評価について述べる．3 章は，自己評価書としては必須の項目ではないが，

本提案の 1 つの特徴であるのであえて記述する．

2　安全性に対する評価

2.1　楕円曲線上の離散対数問題に関する安全性評価

2.1.1　楕円曲線上の離散対数問題への既存攻撃

p を素数とする．有限体 $\mathbb{F}_p$ 上定義された楕円曲線 $E/\mathbb{F}_p : y^2 = x^3 + ax + b, \quad a, b \in \mathbb{F}_p$ に対し，楕円曲線の位数 $n = \#E(\mathbb{F}_p)$ は素数とする．

定義 1 楕円曲線上の離散対数問題
$G \in E(\mathbb{F}_p)$，$ord(G) = n$, $R \in < G >$ が与えられたときに，$R = lG := \underbrace{G + G + \cdots + G}_{l}$ を満たす整数 $0 < l < n$ を求める問題を楕円曲線上の離散対数問題（$ECDLP$）と呼ぶ．

ECDLP に対して有効な解法（準指数時間または多項式時間で解けるアルゴリズム）として，一般的な方法は確立されておらず，特殊な曲線に対する解法のみが存在する．Pohlig-Hellman 法や Pollard-ρ 法等は，任意の ECDLP に適用できる攻撃であるが，いずれも指数時間のアルゴリズムである．したがって，安全な楕円曲線を構成する際には，そのような特殊な曲線を避けなければならない．具体的な既知の攻撃方法としては，特定の楕円曲線のクラスに限定した攻撃に，MOV-帰着法 [8]，FR-帰着法，SSSA アルゴリズム [17, 18, 1] がある．MOV-帰着法と FR-帰着法では $\mathbb{F}_p$ 上定義された ECDLP を $\mathbb{F}_{p^k}^*$ の部分群上の DLP に帰着し，SSSA アルゴリズムは定義体の加法群に帰着する．以下で各攻撃について記載する．ここで，定義体を位数 p の素体，楕円曲線の位数 $n = \#E(\mathbb{F}_p)$，トレース $t := p + 1 - \#E(\mathbb{F}_p)$ とする．

- MOV-帰着法
  $E(\mathbb{F}_{p^k}) \supset E[n] := \{P \in E(\overline{\mathbb{F}_p}) \mid nP = \mathcal{O}\}$ を満たす最小の k について，Weil Paring を用いることで，ECDLP を $\mathbb{F}_{p^k}^*$ 上の DLP に帰着する．

  曲線を supersingular に限定した場合，拡大次数が $k = 1, 2, 3, 4, 6$ のいずれかになり，群の構造が分かるので Weil Paring が簡単に計算でき，容易に攻撃される．最近の研究では，群の構造が未知の場合でも，効率良く Weil Paring を計算するアルゴリズムの提案がなされており [5, 6, 19, 20]，supersingular でない曲線に対しても MOV-帰着法を適用させる研究は進んでいる．

- FR-帰着法
  $n \mid p^k - 1$ (FR-condition) を満たす最小の k について，Tate Paring の変形を用いることで，$\mathbb{F}_{p^k}^*$ 上の DLP に帰着する．

  [2] により，FR-帰着アルゴリズムの適用範囲は，MOV-帰着アルゴリズムの適用範囲を完全に含むことが指摘されている．したがって，楕円曲線の安全性を検討する際には，FR-帰着法に対する安全性を考えれば良い．[5] では，適用範囲及び効率の観点から MOV-帰着アルゴリズムとの比較を行っている．それまで概念的に提示されていた FR-帰着アルゴリズムを実装し，アルゴリズムの立場から，FR-帰着アルゴリズムは MOV-帰着アルゴリズムよりも効率が良いことを指摘している．また，[6, 14] では，理論研究の立場から $t = 2$ の場合は FR-帰着アルゴリズムの中で利用されている写像とは異なる写像を利用することで FR-帰着アルゴリズムよりも効率の良い帰着アルゴリズムを提案している．

- SSSA アルゴリズム
  $t = 1$ である楕円曲線に対して $\mathbb{F}_p$ の加法群の DLP に帰着する．

安全性の観点から，FR-帰着法が適用された場合の埋め込まれる体 $\mathbb{F}_{p^k}$ の拡大次数 k は $k > \log q$ を満たす必要がある [15].

2.1.2 FR-帰着法に対する安全性

本章では，素体 $\mathbb{F}_p$ 上の楕円曲線に対する FR-帰着法について述べる．FR-帰着法に対する安全性を考察する際に，最も重要なポイントは帰着される有限体の拡大次数 k の大きさである．この点についてはじめて触れているのは，MOV-帰着法の提案の直後の [7] の文献である．Koblitz は [7] で，k 次拡大体上の DLP の解法として，$L_p[1/3, c] = exp((c + o(1))(\log p)^{1/3}(\log\log p)^{2/3})$ の準指数時間アルゴリズムを想定した場合，MOV-帰着法が準指数時間になるのは，$k \leq (\log p)^2$ のときであることを指摘している．さらに [2] において，素体上定義された素数位数楕円曲線をランダムに選ぶ場合，$k \leq (\log p)^2$ なる k に対して $n \mid p^k - 1$ が成り立つ確率の上界を以下のように与えている．

$$Pr \leq c\frac{(\log M)^9(\log\log M)^2}{M}$$

ここで，c, M は $\frac{M}{2} \leq p \leq M$ を満たす任意の整数である．これに対して，斎藤，内山らは [15] において上の確率評価を一般化させ，次の結果を導いている．素体上定義された素数位数楕円曲線をランダムに選ぶ場合，$0 < a < 1$ に関して，$k \leq (\log p)^a$ なる k に対して $n \mid p^k - 1$ が成り立つ確率の上界は

$$Pr \leq c\frac{(\log M)^{2a+5}(\log\log M)^2}{M} \tag{1}$$

となる．ここで，c, M は $\frac{M}{2} \leq p \leq M$ を満たす任意の整数である．

Koblitz が指摘するように，DLP の解法として $L_p[1/3, c] = exp((c+o(1))(\log p)^{1/3}(\log\log p)^{2/3})$ の準指数時間アルゴリズムを想定した場合，MOV-帰着法が準指数時間になるのは，$k \leq (\log p)^2$ のときであるが，現在のところ，そのようなアルゴリズムとして数体ふるい法と呼ばれるアルゴリズムがある．しかし，数体ふるい法が適用できるためには p に関していくつかの条件があり，斎藤，内山らが [14] で触れているように，現実的には，DLP の解法としては 2 次ふるい法と呼ばれる $L_p[1/2, c] = exp((c + o(1))(\log p)^{1/2}(\log\log p)^{1/2})$ の準指数時間アルゴリズムを想定すれば十分である．その場合，MOV-帰着法が準指数時間になるのは，$k \leq \log p$ のときとなる．この結果，MOV-帰着法が準指数時間になる確率の上界は，(1) より，

$$Pr \leq c\frac{(\log M)^7(\log\log M)^2}{M} \tag{2}$$

となる．ここで，c, M は $\frac{M}{2} \leq p \leq M$ を満たす任意の整数である．

以上より，FR-帰着法が準指数時間になるのは $k \leq \log p$ のときであり，その存在確率の上界は最大でも (2) の形で表される程度である．しかしながら，これはあくまでも確率的な議論であり，どのようなトレースの時に拡大次数が十分高いことが保証されるかなどの議論はされていない．実際これまでに知られている条件は，$t = 0$ (supersingular) と $t = 2$ のみであるが，どちらも攻撃される条件であった．

一方で我々は FR-condition に関する explicit な条件についての研究を'99 年度より行ってきた．[9] において我々は supersingular 及び $t = 2$ 以外の場合で FR-帰着法を適用したときの拡大次数 k が $k = 3, 4, 6$ となるための explicit な必要十分条件を p と n を用いて示した．しかし，残念なこと

にこれらはすべて解読される条件であり，FR-condition を満たす安全なトレースの条件は全く明らかになっていなかった．

そこで，今回，初めて FR-condition を満たす安全な素数位数楕円曲線の explicit な条件 を，トレースを用いて導いたのでそれを以下に示す．

定理 1 ([11]) $E/\mathbb{F}_p$ を $\mathbb{F}_p$ 上定義された楕円曲線，t を $E/\mathbb{F}_p$ のトレース，とする．*FR*-帰着によって，$E/\mathbb{F}_p$ 上の *ECDLP* が $\mathbb{F}_{p^k}^*$ 上の *DLP* に帰着するとする．この時，$t \geq 3$ に対して，拡大次数 k は次を満たす．

$$k > \frac{\log p}{\log (t-1)}$$

証明: FR-帰着によって，$E/\mathbb{F}_p$ 上の ECDLP が $\mathbb{F}_{p^k}^*$ 上の DLP に帰着できるのは次の条件を満たす時かつその時に限る．

$$p^k \equiv 1 \pmod{n} \tag{3}$$

$n = p + 1 - t$ を，(3) に代入して，次を得る．

$$(t-1)^k \equiv 1 \pmod{n} \tag{4}$$

ここで，$n > (t-1)^k$ の時には，$\mathbb{Z}$ 上の演算と $\mathbb{Z}/n\mathbb{Z}$ 上の演算は一致する．仮定より $t \geq 3$ なので，$n \geq (t-1)^k$ のとき，すなわち $k < \frac{\log n}{\log (t-1)}$ のとき，$(t-1)^k \neq 1 \pmod{n}$ となる．したがって，帰着する拡大次数は

$$k > \frac{\log n}{\log (t-1)}$$

を満たす．ここで，Hasse の定理より，$\log p \sim \log n$ が成り立つから，以下の結果を得る．

$$k > \frac{\log p}{\log (t-1)}$$

∎

定理 1 より，以下の系が導かれる．

系 1 ([11]) $E/\mathbb{F}_p$ を $\mathbb{F}_p$ 上定義された楕円曲線，$E/\mathbb{F}_p$ のトレースを $t = 3$ とする．*FR*-帰着によって，$E/\mathbb{F}_p$ 上の *ECDLP* が $\mathbb{F}_{p^k}^*$ 上の *DLP* に帰着するとする．この時，拡大次数 k は次を満たす．

$$k > \log p$$

このことより，特に $t = 3$ の素数位数楕円曲線では，FR-帰着による ECDLP の解法は指数時間になり，上で挙げた既知の攻撃に対する計算量的耐性を保証できる．

2.1.3 提案楕円曲線の安全性

我々の用いる楕円曲線 $E/\mathbb{F}_p$ は以下の性質を持つ．

1. 近年の Weil Decent の攻撃可能性を削減するために，$\mathbb{F}_{2^r}$ を用いずに素体上の楕円曲線 $E/\mathbb{F}_p$ とする．
2. ECDLP に対する exhaustive 攻撃に相当する Pohlig-Hellman[12]，Pollard-ρ 法 [13] に対する耐攻撃性を高めるため，素数位数 $n = \#E(\mathbb{F}_p)$ の楕円曲線 $E/\mathbb{F}_p$ とする．
3. SSSA に対する耐攻撃性を確保するため，$\#E(\mathbb{F}_p) \neq p$ とする．

4. FR-帰着法に対する計算量的耐性（i.e. 帰着拡大次数 $k > \log p$）を保証する．
5. 楕円曲線がシステマティックに容易に効率的に構成できる．

現時点で，問題となる攻撃は exhaustive 攻撃，SSSA 攻撃，FR-攻撃法である．上記 2.3 及び前章の定理より，我々の提案する楕円曲線上の ECDLP の解読には $O(\sqrt{n})$ 時間の，すなわち Hasse の定理より $O(\sqrt{p})$ 解読時間を要する．これは $\mathbb{F}_p$ 上の ECDLP で実現できる最強の安全性を持つことを意味する．特に推奨パラメータ p ： 160 ビット（仕様書 3.1.3）においてはおおざっぱで 2^{80} 回の試行回数が解読には必要であり，十分な安全性を確保できる．

2.2 汎用楕円曲線 Diffie-Hellman 鍵共有アルゴリズム

2.2.1 鍵共有アルゴリズム

ここでは本提案スキームについて述べる．本提案スキームは ECDH 鍵共有法 [3] である．ECDH は各ユーザが同じ楕円曲線を利用することを前提にする．本提案では，ユーザに対する汎用性，柔軟性を持たせるために，ユーザ毎のシステムパラメータが異なる場合についても，利用可能なように ECDH を拡張する [10].

前提

各ユーザはそれぞれ通信相手の公開鍵の正当性をあらかじめチェックすることを前提にする．公開鍵の正当性については，認証局の証明書を利用するなどの一般的な方法を用いると良い．

補助関数

各アルゴリズムでは，範囲内の任意の整数を生成する際に擬似乱数生成関数を利用する．擬似乱数生成関数については，暗号技術仕様書の 1.2.3 章に記載した性質を満たすものを利用する．ここでは，一般的に用いられている方法として，時刻（ミリ秒）やユーザーによる適当なキー入力等をもとに乱数の種（seed）を決定し，それをもとに計算したハッシュ値を擬似乱数とする．ハッシュ関数としては SHA-1 や MD-5 を用いる．

初期設定

ユーザ A は以下の初期設定を行う．

1. $t = 3$ の素数位数楕円曲線 $E_A/\mathbb{F}_{p_A}$ を生成する．
2. ベースポイント $G_A \in E_A(\mathbb{F}_{p_A})$ をランダムに選ぶ．
3. 整数 x_A $(0 < x_A < p_A - 2)$ をランダムに選び，秘密鍵とする．
4. $Y_A = x_A G_A$ を計算し，$(E_A/\mathbb{F}_{p_A}, Y_A, G_A)$ を公開鍵とする．

同様にして B の公開鍵を $(E_B/\mathbb{F}_{p_B}, Y_B, G_B)$ とする．

鍵の共有 [共通システムパラメータの場合（楕円曲線とベースポイントが同じ）]

楕円曲線を $E/\mathbb{F}_p = E_A/\mathbb{F}_{p_A} = E_B/\mathbb{F}_{p_B}$，ベースポイントを $G = G_A = G_B$ とする．

鍵共有

- A は，$K = x_A Y_B = x_A x_B G$ を計算し，K を共有する．
- B は，$K = x_B Y_A = x_B x_A G$ を計算し，K を共有する．

鍵の共有 [個別システムパラメータの場合（楕円曲線とベースポイントが異なる）]

利用者 A

1. r_A $(0 < r_A < p_B - 2)$ となる任意の整数を選ぶ.
2. $R_A = r_A G_B$ を $E_B/\mathbb{F}_{p_B}$ 上で計算する.
3. R_A を B に送る.

利用者 B

1. r_B $(0 < r_B < p_A - 2)$ となる任意の整数を選ぶ.
2. $R_B = r_B G_A$ を $E_A/\mathbb{F}_{p_A}$ 上で計算する.
3. R_B を A に送る.

鍵共有

- A は,
  $K_A = x_A R_B = x_A r_B G_A, \quad K_B = r_A Y_B = x_B r_A G_B$
  を計算し，(K_A, K_B) を共有する.
- B は,
  $K_B = x_B R_A = x_B r_A G_B, \quad K_A = r_B Y_A = x_A r_B G_A$
  を計算し，(K_A, K_B) を共有する.

注意:実際の共有鍵は (K_A, K_B) より計算できる値としてよい. 例えば各々の x 座標成分 K_{A_x}, K_{B_x} を用いて $K = K_{A_x} \oplus K_{B_x}$（$\oplus$ はビット毎の排他的論理和）とする．これはアプリケーションに応じて設定可能である.

2.2.2 安全性評価

提案するスキームは従来の楕円曲線 Diffie-Hellman 鍵共有法（ECDH 鍵共有法）を異なるシステムパラメータでも鍵共有が可能になるように拡張したものである．各ユーザの利用するシステムパラメータが同じ場合は，ECDH と一致する．IEEE P1363 で明記されているように，各々の公開鍵の正当性を各々が検証する.

我々の提案するスキームは，従来から知られている ECDH 鍵共有法であるが，ここで改めて定義する.

定義 2 *ECDH* 問題
$G \in E(\mathbb{F}_p)$，$ord(G) = n$，$P, R \in < G >$ が与えられたとき，$S \in < G >$ を求める問題である．ここで，$P = lG,\ R = mG,\ S = lmG$ である.

さらに，異なるシステムパラメータにおける鍵共有法に対する，汎用 ECDH 問題を定義する.

定義 3 汎用 *ECDH* 問題
$G_1 \in E_1(\mathbb{F}_{p_1})$，$ord(G_1) = n_1$，$P_1, R_1 \in < G_1 >$ 及び
$G_2 \in E_2(\mathbb{F}_{p_2})$，$ord(G_2) = n_2$，$P_2, R_2 \in < G_2 >$ が与えられたとき,
$S_1 \in < G_1 >$ と $S_2 \in < G_2 >$ の両方を求める問題である．ここで,
$P_1 = l_1 G_1,\ R_1 = m_1 G_1,\ S_1 = l_1 m_1 G_1$
$P_2 = l_2 G_2,\ R_2 = m_2 G_2,\ S_2 = l_2 m_2 G_2$ である.

公開鍵暗号系における攻撃方法の種類は，能動的攻撃と受動的攻撃に大別できる.

- 受動的攻撃
  盗聴や傍受等によって 2 者間の通信から何らかの情報を獲得し，それを用いて共有する鍵を復元するような，受動的攻撃が考えられる．この場合，各々のユーザの利用する楕円曲線上の ECDH 問題はそれぞれを統合した汎用 ECDH 問題に帰着する [10].

- 能動的攻撃
  攻撃者が 2 者間の通信の間に入り込み，A に対しては B になりすまし，B に対しては A になりすますという中間侵入攻撃が考えられる．それを防ぐためには，通信相手の公開鍵の正当性を仮定（もしくはユーザがチェック）するなどの認証機能が必要である．これに関しては，ECDH も汎用 ECDH も同じ安全性である．

我々の提案はあくまでも汎用的に利用されている ECDH 鍵共有法に我々の楕円曲線を適用することであり，全ての安全性の議論（公開鍵の正当性のチェック）等は ECDH 鍵共有法に準ずる．

3　楕円曲線生成アルゴリズムの評価

本提案は楕円曲線を用いた鍵共有法であるが，初期設定部についての計算時間の評価は，本提案の特徴である FR-帰着に対する計算量的安全性の保証された楕円曲線の生成に相当するので詳細に記載する．時間評価として本質的な部分はトレース 3 の楕円曲線の生成と，楕円曲線上の 1 回のべき倍演算の時間となる．

評価の対象となる項目は次の通りである．

- 初期設定部
  - トレース 3 の楕円曲線 $E_A/\mathbb{F}_{p_A}$ の生成時間
  - 秘密鍵 x_A の生成時間
  - 公開鍵 $G_A, Y_A = x_A G_A$ の生成時間

3.1　トレース 3 の曲線の生成に関する評価

トレース 3 の楕円曲線 $E_A/\mathbb{F}_{p_A}$ の生成時間については，評価として本質的な部分は双子素数の生成である．これは一般的な素数判定法を用いれば良いので，素数判定時間を評価の対象とするのではなく，双子素数の出現頻度を評価すれば良い．ここで，出現頻度とは探索回数を n_l，個数を n_p すると，$\frac{n_p}{n_l}$ で表す．双子素数が一様に分布しているとすれば，n_l が十分に大きいとき，大数の法則より上で示した出現頻度は双子素数が存在する真の確率に近づく．したがって，

$$\text{双子素数の生成時間} = \frac{n_l}{n_p} \times \text{素数判定時間}$$

として評価すれば良い．次に我々の実験結果を示す．

整数 l に対して $p = dl^2 + dl + \frac{d+9}{4}$ とおくとき，$2^{76} - 2^{20} \leq l \leq 2^{76} + 2^{20}$，すなわち $n_l = 2^{21}$ に対して双子素数 $(p, p-2)$ の出現頻度について表 1 の結果が得られている．

双子素数の出現確率は d によって異なる．最も出現確率の高いもので，160bits の双子素数の確率はおおよそ 0.00245953 程度である．したがってこの場合，双子素数の生成時間はおおよそ以下で与えられる．

$$\text{双子素数の生成時間} = 406.582 \times \text{素数判定時間}$$

実験的な議論ではあるが，これは FR-condition の判定を除いた単純な CM 法による楕円曲線の構成とほぼ同等の速度で楕円曲線の生成を意味する．

4　ソフトウェアでの実装評価

評価の対象となる項目は次の通りである．

- 鍵共有部
  - r_A の生成時間
  - $R_A = r_A G_B$ の計算時間
  - $K_A = x_A R_A$ の計算時間
  - $K_B = r_A Y_B$ の計算時間

鍵共有部についての計算時間の評価として本質的な部分は楕円曲線上の 3 回のべき倍演算の時間となる．

表 1: 双子素数の出現頻度 $n_l = 2^{21}$

d	# 双子素数の個数 n_p	双子素数の出現頻度 $\frac{n_p}{n_l}$
19	190	0.00009059
43	1,157	0.00055170
67	1,902	0.00090694
91	450	0.00021457
115	1,036	0.00049400
139	139	0.00006628
163	5,158	0.00245953
187	1,402	0.00066852
211	292	0.00013923
235	2,523	0.00120306
259	247	0.00011777
283	645	0.00030756
307	696	0.00033187
331	458	0.00021839
355	635	0.00030279
379	583	0.00027799
403	3,392	0.00161743

4.1 鍵共有に関する速度評価

ECDH 鍵共有法を用いる場合は，1 回の楕円曲線上のべき倍演算，ユーザ間のシステムパラメータが異なる場合，すなわち汎用 ECDH 鍵共有法では，3 回の楕円曲線上のべき倍演算が必要になる．

我々の提案方式ではベースポイントを固定することを仮定せずに，任意の点に関するべき倍演算で可能なアルゴリズムを利用する．効率良く計算を行うために我々は，符合付 2 進法とウィンドウ法を組み合わせた加算連鎖をヤコビ座標系を用いて実現した．

4.1.1 楕円曲線上のべき倍演算の速度評価及び，プログラムサイズと RAM サイズ

演算速度の測定方法は clock() 関数を使用して 10,000 回の処理時間の平均である．なお，多倍長演算については松下電器（株）製の多倍長演算ライブラリ，ANRI97 を使用した．

必要な RAM サイズについては，多倍長演算部 (ANRI) は除外し，プログラムソースコードから直接算出した結果以下のように合計して，おおよそ 7 KB である．

static table	4,353 Byte
stack	2,679 Byte

以上より，PentiumII 400MHz においては，152 KB のプログラムサイズで，7 KB 程度の RAM を用いて，ECDH 鍵共有法を，約 3.1msec，汎用 ECDH 鍵共有法を，約 9.3msec で実行できる．

表 2: 楕円曲線上のべき倍演算の速度評価とライブラリのサイズ

評価プラットフォーム	Sun Solaris 5.6, UltraSparc-IIi 300MHz, 512MB	MS-Windows NT 4.0 SP5, PentiumII 400MHz, 128MB
コンパイラ	gcc 2.95.2	MS-Visual C++ 6.0
1 回の楕円べき倍算	7.8 msec	3.1 msec
ECDH 鍵共有	7.8 msec	3.1 msec
汎用 ECDH 鍵共有	23.4 msec	9.3 msec
ライブラリのオブジェクトサイズ	108 KB	152 KB

4.1.2 テストベクタサンプル

以下にテストベクタのサンプルを示す．出力形式は，Big Endian 16 進表記である．
p を定義体の位数，n を楕円曲線の位数，Gx をベースポイントの x 座標，Gy をベースポイントの y 座標とする．これらがシステムパラメータである．

p : 0x80000000 0x00000005 0x53f0e64c 0xa74b83fa 0x9d08fa01
n : 0x80000000 0x00000005 0x53f0e64c 0xa74b83fa 0x9d08f9ff
Gx : 0x00000000 0x00000000 0x00000000 0x00000000 0x00000000
Gy : 0x5be84c4c 0x3ba3fb4e 0x6a6c84cb 0xa7c1fb13 0x60631734

A の秘密鍵を k とすると，公開鍵は (kGx, kGy) となる．
k : 0x3aad44ed 0x12861afb 0x5e5c4131 0x68f84d42 0x04f27b57
kGx : 0x787134cc 0x5608bd57 0x83eddccf 0x5eb93300 0x5bd1f7f0
kGy : 0x3945af93 0x7964ba1a 0xc3871f83 0x9af45e55 0x9dc280b1

B の秘密鍵を l とすると，公開鍵は (lGx, lGy) となる．
l : 0x2dd449a9 0x5f1b0da4 0x596327f3 0x011f678e 0x4cc93293
lGx : 0x7bf844d4 0xd30e66c2 0x2a7d7d9c 0xedbdd077 0x0b95207e
lGy : 0x0a7fddff 0x5f8d25a7 0x6a054f37 0x22397d81 0x364efbab

A,B は以下の情報を共有する．
klGx : 0x2cee5cd1 0x7562facf 0x2eda9ae8 0xdbdb91d7 0x01fe8086
klGy : 0x32bec425 0xcbe0c28a 0xae4df559 0xaa6b20ef 0x63c070b2

4.1.3 テストベクタファイル出力形式

テストベクタについては TestVector**.txt というファイルを 10 個添付した．1 つのファイルは各システムパラメータ，すなわち定義体，楕円曲線，ベースポイントに対応している．各ファイルには，そのシステムパラメータに基づいた出力例を 20 個出力した．出力形式は，以下の通りである．いずれも，Big Endian 16 進表記である．

********* 出力結果 *********
a : 楕円曲線の 1 次係数
b : 楕円曲線の 0 次係数

p　　: 定義体
n　　: 楕円曲線の位数
Gx　: ベースポイントの x 座標
Gy　: ベースポイントの y 座標

k　　: 整数 $0 < k < p$（秘密鍵）
kGx　: 公開鍵の x 座標
kGy　: 公開鍵の y 座標
l　　: 整数 $0 < l < p$（秘密鍵）
lGx　: 公開鍵の x 座標
lGy　: 公開鍵の y 座標
klGx : 共有情報の x 座標
klGy : 共有情報の x 座標
****
最初の 6 個のデータがシステムパラメータであり，これらに基づいた出力例が続く 8 個のデータである．各ファイルでは，この出力例を 20 組出力した．

参考文献

[1] K. Araki and T. Satoh “Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves”, *Commentarii Math. Univ. St. Pauli.*, vol. **47** (1998), 81-92.

[2] R. Balasubramanian and N. Koblitz, “The Improbability That an Elliptic Curve Has Subexponential Discrete Log Problem under the Menezes-Okamoto-Vanstone Algorithm”, *Journal of CRYPTOLOGY*, **11** (1998), 141-145.

[3] W. Diffie and M. Hellman, “New directions in cryptography”, *IEEE Trans. Inf. Theory*, **IT-22** (1976), 644–654.

[4] G. Frey and H. G. Rück, “A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves”, *Mathematics of computation*, **62**(1994), 865-874.

[5] R. Harasawa, H. Imai, J. Shikata, J. Suzuki, “Comparing the MOV and FR Reductions in Elliptic Curve Cryptography”, *Advances in Cryptology-Proceedings of EUROCRYPT ’99*, Lecture notes in Computer Science, **1592** (1999), 190-205.

[6] N. Kanayama, T. Kobayashi, T. Saito, and S. Uchiyama ”Remarks on elliptic curve discrete logarithm problems” , *IEICE Trans.*, Fundamentals. vol. E83-A, No.1 JANUARY 2000, 17-23.

[7] N. Koblitz, “Elliptic curve imiplementation of sero-knowledge blobs”, *Journal of CRYPTOLOGY*, **4** (1991), 207-213.

[8] A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field", *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing* (1991), 80–89.

[9] 宮地 充子, 高野 俊三, "Some explicit conditions for FR-reduction", **第３回代数幾何・数論及び符号・暗号**, (2000), 74-85.

[10] A. Miyaji and H. Shizuya "Integration of DLP-based cryptosystems", *IEICE Japan Tech. Rep.*, **ISEC99-48**(1999-9), 73-80.

[11] A. Miyaji, M. Nakabayashi, and S. Takano, "New relation between FR-reduction and elliptic curve traces", to appear in ISEC2000-9.

[12] S. C. Pohlig and M. E. Hellman, "An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance", *IEEE Trans. Inf. Theory*, **IT-24** (1978), 106–110.

[13] J. Pollard, "Monte Carlo methods for index computation (mod p)", *Mathematics of Computation*, **32** (1978), 918–924.

[14] T. Saitoh and S. Uchiyama, "A Note on the Discrete Logarithm Problem on Elliptic Curves of Trace Two", *Technical Report of IEICE*, ISEC98-27(1998), 51-57.

[15] 斎藤 奏一, 内山 成憲, "楕円曲線上の離散対数問題について I", *Proceedings of the SCIS'98*, **6.1.C** (1998).

[16] R. Schoof, "Counting points on elliptic curve over finite fields", *Journal de Théorie des Nombres de Bordeux*, **7** (1995), 219–254.

[17] I. A. Semaev "Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p", *Mathematics of computation*, **67** (1998), 353-356.

[18] N. P. Smart "The discrete logarithm problem on elliptic curves of trace one", to appear in J. Cryptology.

[19] J.Shikata, Y.Zheng, J.Suzuki and H.Imai "Generalizing the Menezes-Okamoto-Vanstone (MOV) algorithm to non-supersingular elliptic curves", *Proceedings of the SCIS'99*, (2000),26-29.

[20] J.Shikata, Y.Zheng, J.Suzuki and H.Imai "Realizing the Menezes-Okamoto-Vanstone (MOV) Reduction Efficiently for Ordinary Elliptic Curves", *IEICE Trans.*, Fundamentals. vol. E83-A, No.4 APRIL 2000, 756-763.