暗号技術仕様書

1　技術の背景

本提案暗号技術は，楕円曲線上の離散対数問題（ECDLP）に基づく鍵共有法である．本提案暗号技術において，暗号プリミティブとしてECDLPを用いるのは以下の理由からである．

1. 素因数分解問題（IFP），有限体上の離散対数問題（DLP）には，指数計算法 [13, 14]，数体ふるい法 [5] 等の準指数時間の攻撃が存在するのに対し，ECDLPには現時点では準指数時間の攻撃が存在しない．
2. IFPやDLPと同じ安全性を仮定したときの鍵サイズを小さくできる．

有限体上定義された楕円曲線（$E/\mathbb{F}_q$）からECDLPを安全に構成するには，$\#E(\mathbb{F}_q)$ がnon-smoothであると共に，$G \in E(\mathbb{F}_q)$，$n = ord(G)$ がFR-condition, i.e.

$$1 \leq k \leq \log q \text{ に対して } q^k \neq 1 \pmod{n}$$

を満たす必要がある．[2] より，FR-conditionを満足しない楕円曲線 $E/\mathbb{F}_p$ は，確率的に非常に少ないことが示されている．しかしあくまでも確率的な議論であり，explicitな条件は，$E/\mathbb{F}_q$ がsupersingularあるいはトレース $t = 2$ の場合を除いて，明らかにされていなかった．ここでトレースとは $t := q + 1 - \#E(\mathbb{F}_q)$ である．我々は，新たなexplicitな条件を示した [9] が，いずれもFR-帰着に対して弱い条件であり，安全な条件でexplicitな条件は全く知られていなかった．

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．つまり，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が，一般に $E/\mathbb{F}_q$ の構成はRSAの構成に比べて複雑である．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．

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．この楕円曲線の性質が利用できるスキームとしてECDH鍵共有法を提案する．

従来から，鍵共有法として知られているECDH鍵共有法 [3] では，ベースポイント，定義体，楕円曲線のパラメータ等は共通のシステムパラメータとして固定することが前提となっている．我々の提案する楕円曲線は，FR-帰着法 [4] による攻撃に対する安全性が保証され，かつ容易に生成できる特徴を有する．この特徴を用いて，システムパラメータをユーザが個別に選択／変更可能なように従来の楕円曲線Diffie-Hellman鍵共有法（ECDH鍵共有法）を拡張する [10]．この拡張により，例えばスマートカード等においてワンタイムパスワード的に楕円曲線を生成し，その楕円曲線を用いて鍵共有を実現することが可能となる．

以上の問題を鑑みて，本提案暗号技術では以下の設計方針を満たすECDLPプリミティブを用い，各種のシステムパラメータを利用者が個別に選択／変更が可能な汎用ECDH鍵共有法を提案する．

本仕様書の構成は以下のとおりである．2章に基本関数である楕円曲線及び提案スキーム，利用する補助関数各々の設計方針について述べる．3章に提案暗号スキームとして基本関数である楕円曲線生成アルゴリズムと楕円演算アルゴリズム，次に提案鍵共有アルゴリズム，そして利用する補助関数について述べる．

2 設計方針

2.1 基本関数（安全な楕円曲線の設計）

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[13]，Pollard-ρ法[14]に対する耐攻撃性を高めるため，素数位数の楕円曲線 $E/\mathbb{F}_p$ 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．

我々はFR-帰着法に対する安全なexplicitな条件として以下の結果を導いた．これは上記の設計方針を満足する楕円曲線を与える．（詳細な証明は自己評価書に記載）

定理 1 ([11]) p を素数，$E/\mathbb{F}_p$ 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．

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

系 1 ([11]) p を素数，$E/\mathbb{F}_p$ 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．

$$k > \log p$$

[15]の結果より，FR-帰着法によってDLPに帰着する拡大次数が $k > \log p$ を満たすとき，FR-帰着法は指数時間の攻撃にしかならない，つまり有効でないことが示されている．系1より，トレース3の楕円曲線はFR-帰着法に対する安全性が保証される，すなわちFR-conditionを満たす初めてのexplicitな条件となる．また，トレース3の楕円曲線は，$\mathbb{F}_p$ 上の素数位数 $\#E(\mathbb{F}_p) = p - 2$ の楕円曲線を意味し，双子素数 $(p, p-2)$ を求める問題に帰着する．双子素数の探索問題は，数論においても興味ある問題のひとつであり，その問題にFR-帰着法に対する安全性が保証された楕円曲線の探索問題が帰着するのは非常に興味深い．

我々は，FR-conditionを満たすexplicitな条件を導いたことにより，FR-conditionが成り立つかどうかのチェックを行うことなく，効率良くFR-帰着法に対して安全な楕円曲線を生成することができる．具体的に $E/\mathbb{F}_p$，$\#E(\mathbb{F}_p) = p - 2$ となる楕円曲線の構成方法については，2章で述べる．

2.2 スキーム（鍵共有）

本提案においては，標準的に利用されている ECDH[3] との互換性を重視するとともに，ユーザごとに楕円曲線等のシステムパラメータが異なるときにも柔軟に対応できることを設計方針にする．すなわち，以下のように定める．

- 固定システムパラメータのときは，ECDH 鍵共有法を用いる
- ユーザ個別システムパラメータのときは，汎用 ECDH 鍵共有方を用いる

2.3 補助関数（乱数生成関数）

本提案鍵共有方式では、秘密鍵の生成の際に乱数を用いる。乱数の利用においては、真性乱数が望ましいが、実際には擬似乱数生成関数を用いる。

擬似乱数生成関数を利用する際には、以下の性質を満たす関数を用いる [12]。

- 統計的一様性（任意の長さにおける、全系列の等頻度性）
- 無相関性
- 長周期性
- 非線形性（大きな線形複雑度）

3　提案暗号スキーム

ここでは，暗号スキームとして，楕円曲線生成アルゴリズム，楕円曲線上のべき倍演算アルゴリズム，提案 ECDH 鍵共有法について述べる．

本提案技術は，鍵共有法の提案のため，楕円曲線生成アルゴリズムは実装上本質的でない．しかし本提案は，FR-帰着法に対する安全性が保証された楕円曲線をシステマティックに生成することにも特徴を持つので，楕円曲線生成アルゴリズムについても記載する．また，楕円曲線上のべき倍演算アルゴリズムは，提案スキームの実現に必須なアルゴリズムである．前章で述べたように，ユーザ毎に楕円曲線を生成する，つまりワンタイムパスワード的に楕円曲線を利用することにより安全性を高める場合，この実用的な楕円曲線生成アルゴリズムは非常に有効である．

3.1　基本関数

3.1.1　楕円曲線生成アルゴリズム

楕円曲線生成アルゴリズムとして，理論的なアルゴリズムと実用的なアルゴリズムの 2 つを明記する．現実的には，実用的なアルゴリズムで十分である．前章に述べたトレース 3 となる素数位数の楕円曲線を生成するアルゴリズムは以下のようになる．

アルゴリズム 1 (理論的アルゴリズム)

1. $d \equiv 19 \pmod{24}$ を満たす整数 d を選ぶ.
2. 整数 l に対して $p = dl^2 + dl + \frac{d+9}{4}$ とおく.
3. $(p, p-2)$ のどちらかが合成数のとき，*1.* に戻る．両方素数のとき，*4.* に行く.
4. d によって定まる類多項式 $P_d(x)$ を求める.
5. $P_d(X) \equiv 0 \pmod{p}$ の解 j_0 を求める.
6. j_0 を *j-invariant* とする $\mathbb{F}_p$ 上楕円曲線 $\{E_{j_0}\}$ を構成する．ここで，E_{j0} とは,

$$E_{j_0} : y^2 = x^3 + a_{j_0}x + b_{j_0}$$

$$a_{j_0} = \frac{3j_0}{1728 - j_0} \pmod{p}, \qquad b_{j_0} = \frac{2j_0}{1728 - j_0} \pmod{p}$$

となる楕円曲線であり，$\{E_{j_0}\}$ はその同型な楕円曲線の集合を意味する.
7. $\{E_{j_0}\} \ni E : y^2 = x^3 + ax + b$ を，$\#E(\mathbb{F}_p) = p - 2$ かつ $a = -3$ あるいは小さい a をとる.

ステップ 4. にある類多項式の求め方について表記する.

アルゴリズム 2 (類多項式の構成アルゴリズム)

1. $P \leftarrow 1$,　$b \leftarrow d \pmod 2$,　$B \leftarrow \lfloor\sqrt{|d|/3}\rfloor$
2. $t \leftarrow \frac{b^2-d}{4}$,　$a \leftarrow max(b, 1)$
3. $a \nmid t$ の場合 *4.* に行く.
　$a \mid t$ の場合，$j \leftarrow j(\frac{-b+\sqrt{d}}{2a})$ を計算する.
　　$a = b$ または $a^2 = t$ または $b = 0$ の場合，$P \leftarrow P \cdot (X - j)$
　　それ以外の場合 $P \leftarrow P \cdot (X^2 - 2Re(j)X + |j|^2)$
4. $a \leftarrow a + 1$
　$a^2 \leq t$ の場合 *3.* に行く．$a^2 > t$ の場合 *5.* に行く.

5. $b \leftarrow b+2$
   $b \leq B$ の場合 2. に行く．$b > B$ の場合 P の係数を最も近い整数に設定し P を出力する．

ステップ 3. にあるモジュラー関数 $j()$ は以下で定義する．

$$j(\tau) = \frac{(256f(\tau)+1)^3}{f(\tau)}, \quad f(\tau) = \frac{\Delta(2\tau)}{\Delta(\tau)}$$

$$\Delta(\tau) = q\left\{1+\sum_{n=1}^{\infty}(-1)^n(q^{n(3n-1)/2}+q^{n(3n+1)/2})\right\}^{24}, \quad q = e^{2i\pi\tau}$$

理論的アルゴリズムにおいて，d を固定すると，$P_d(X)$ も固定されるので，非常に効率的に，コンパクトに実現できる．ここでは，$d = 403$ に固定した場合のアルゴリズムについて明記する．素体上定義された楕円曲線 $E/\mathbb{F}_p : y^2 = x^3 + ax + b$ において，a, b, p を出力する．

アルゴリズム 3 (実用的アルゴリズム)

1. 整数 l に対して $p = 403l^2 + 403l + 103$ とする．
2. $(p, p-2)$ のどちらかが合成数のとき，1. に戻る．両方素数のとき，3. に行く．
3. $X^2 + 24528113892293313919795200000X - 10884420340249105583308800000 \equiv 0 \pmod{p}$ の解 j_1 と j_2 を求める．すなわち，以下のように j_1, j_2 を定める．

$$j_1 = -12264056946146656959897600000 + 34014373972724674193817600\sqrt{13} \pmod{p}$$

$$j_2 = -12264056946146656959897600000 - 34014373972724674193817600\sqrt{13} \pmod{p}$$

4. 楕円曲線 E_{j_1}, E_{j_2} を得る．
   $E_{j_1} : y^2 = x^3 + a_{j_1}x + b_{j_1}$
   $a_{j_1} = \frac{3j_1}{1728-j_1} \pmod{p}, \qquad b_{j_1} = \frac{2j_1}{1728-j_1} \pmod{p}$

   $E_{j_2} : y^2 = x^3 + a_{j_2}x + b_{j_2}$
   $a_{j_2} = \frac{3j_2}{1728-j_2} \pmod{p}, \qquad b_{j_2} = \frac{2j_2}{1728-j_2} \pmod{p}$

5. $E_{j_1} \ni G \neq \mathcal{O}$ に対して，$(p-2)G \neq \mathcal{O}$ ならば $a_{j_1} \leftarrow a_{j_1}c^2, \quad b_{j_1} \leftarrow b_{j_1}c^3$ とする．
   $E_{j_2} \ni G \neq \mathcal{O}$ に対して，$(p-2)G \neq \mathcal{O}$ ならば $a_{j_2} \leftarrow a_{j_2}c^2, \quad b_{j_2} \leftarrow b_{j_2}c^3$ とする．
   ここで c は p を法とした任意の平方非剰余な数である．
6. $a_{j_1}k^2 = -3 \pmod{p}$ なる k が存在する場合

$$a_1 = -3, \quad b_1 = b_{j_1}k^3 \pmod{p}$$

   上記の k が存在しない場合，$a_{j_1}k^2 \pmod{p}$ がなるべく小さくなる k に対して，

$$a_1 = a_{j_1}k^2 \pmod{p}, \quad b_1 = b_{j_1}k^3 \pmod{p}$$

   a_1，b_1 を出力．
   同様に，$a_{j_2}k^2 = -3 \pmod{p}$ なる k が存在する場合

$$a_2 = -3, \quad b_2 = b_{j_2}k^3 \pmod{p}$$

   上記の k が存在しない場合，$a_{j_2}k^2 \pmod{p}$ がなるべく小さくなる k に対して，

$$a_2 = a_{j_2}k^2 \pmod{p}, \quad b_2 = b_{j_2}k^3 \pmod{p}$$

   a_2，b_2 を出力．

3.1.2　楕円曲線上のべき倍演算アルゴリズム

本章では楕円曲線上のべき倍演算アルゴリズムについて述べる．基本的には，加算公式，2 倍算公式の繰り返しで実現できるので，ヤコビ座標 [7] による加算，2 倍算の公式について明記した後，k 倍算を求めるアルゴリズムについて述べる．$\mathbb{F}_p$ 上定義された楕円曲線を

$$E : y^2 = x^3 + ax + b \quad (a, b \in \mathbb{F}_p \quad 4a^3 + 27b^2 \neq 0)$$

とする．ヤコビ座標系での演算を行うため，$x = X/Z^2$，$y = Y/Z^3$ と置き，

$$E_J : Y^2 = X^3 + aXZ^4 + bZ^6$$

を得る．ここで，$P = (X_1, Y_1, Z_1)$，$Q = (X_2, Y_2, Z_2)$ とする．

- ヤコビ座標による加算公式 $P + Q = (X_3, Y_3, Z_3)$ $(P \neq \pm Q)$

$$X_3 = -H^3 - 2U_1H^2 + r^2, \quad Y_3 = -S_1H^3 + r(U_1H^2 - X_3), \quad Z_3 = Z_1Z_2H$$

$$U_1 = X_1Z_2^2, \quad U_2 = X_2Z_1^2, \quad S_1 = Y_1Z_2^3, \quad S_2 = Y_2Z_1^3, \quad H = U_2 - U_1, \quad r = S_2 - S_1$$

- ヤコビ座標による 2 倍算公式 $2P = (X_3, Y_3, Z_3)$

$$X_3 = T, \quad Y_3 = -8Y_1^4 + M(S - T), \quad Z_3 = 2Y_1Z_1$$

$$S = 4X_1Y_1^2, \quad M = 3X_1^2 + aZ_1^4, \quad T = -2S + M^2$$

次に，上で定義した加算及び 2 倍算公式に基づいて，楕円曲線上のべき倍演算アルゴリズムについて述べる．ここで用いる方法は加算連鎖と呼ばれる一般的なアルゴリズムである．本提案で用いる楕円曲線上のべき倍演算ではベースポイントの固定を前提としない．そのため，固定であることを利用したアルゴリズム [6] ではなく，汎用的である符合付 2 進法とウィンドウ法との組合わせを用いる [7]．アルゴリズムは大きく分けて次の 3 部からなる．

1. 予備計算（テーブルの作成）
2. 符合付 2 進表記とウィンドウ計算（ウィンドウ幅は $w = 4$ を設定する）
3. k 倍算を計算

アルゴリズム中でウィンドウ幅は $w = 4$ を設定する．以下のアルゴリズムは楕円曲線上の点のべき倍点を $2^w - 1$ 個計算し，これらに基づいて後の k 倍点を計算する．なお，このアルゴリズムの計算結果はメモリ中に蓄えておき，k 倍点の計算終了後，破棄する．

アルゴリズム 4 (予備計算)

1. $P_1 \leftarrow P$，$P_2 \leftarrow [2]P$
2. $i = 1$ から $2^{w-1} - 1$ の間 $P_{2i+1} \leftarrow P_{2i-1} + P_2$

以下のアルゴリズムは与えられた k を符合付 2 進表記 k' に変換し，ウィンドウ W を出力する．

アルゴリズム 5 (符合付 2 進表記とウィンドウ計算を求めるアルゴリズム)

1. $n = \lfloor \log_2 k \rfloor$ を計算する．
2. $i \leftarrow 0$，$j \leftarrow 0$，$k[n+1] \leftarrow 0$ とおく．
3. $i \leq n$ に対して，以下を実行する．

- $i+w-1 \geq n-w$ の場合，
  $k'[i] \leftarrow k[i], \cdots, k'[n] \leftarrow k[n]$ とおく．
  $W[j] \leftarrow (k[i+w-1], \cdots, k[i+1], k[i])$ とおき，*3.* に行く．
- $k[i]=0$ の場合，
  $k'[i] \leftarrow k[i],\ \ i \leftarrow i+1$ とおき，*2.* に行く．
- $k[i]=1$ の場合，
  $t[j] = \sum_{t=0}^{w-1} k[i+t]2^t$ とおく．
  - $k[i+w]=0$ の場合，
    $k'[i] \leftarrow k[i], \cdots, k'[i+w] \leftarrow k[i+w]$ とおく．
    $W[j] \leftarrow (k[i+w-1], \cdots, k[i+1], k[i])$ とおく．
    $j \leftarrow j+1,\ i \leftarrow i+w+1$ とおき，*2.* に行く．
  - $k[i+w]=1$ の場合，$k[t]=0,\ (t=i+w+1, \cdots)$ が成り立つ最初の t について，
    $k'[i+w] \leftarrow 0, \cdots, k'[t-1] \leftarrow 0,\ k[t] \leftarrow 1$ とおく．
    $t[j] \leftarrow 2^w - t[j] = \sum_{t=0}^{w-1} k\prime[i+t]2^t$（*2* 進展開を行う）．
    $k'[i] \leftarrow -k'[i], \cdots, k'[i+w-1] \leftarrow -k'[i+w-1]$ とおく．
    $W[j] \leftarrow (k[i+w-1], \cdots, k[i+1], k[i])$ とおく．
    $i \leftarrow t,\ j \leftarrow j+1$ とおき，*2.* に行く．

4. $k' = \sum_{t=0} k'[i]2^t\ (k'[i]=0, \pm 1),\ W[i]\ (i=0,1,\cdots)$ を出力．

上のアルゴリズムによって，

$$k' = 2^{t_0}(2^{t_1}(\cdots 2^{t_{s-1}}(2^{t_s}W[s] + W[s-1])\cdots) + W[0])\ (0 \leq t_i)$$

$$W[s] = (k'[n], \cdots, k'[i+w])$$

と書ける．これらを用いて以下のアルゴリズムを実行する．

アルゴリズム 6 (k **倍算を求めるアルゴリズム**)

1. $Q \leftarrow P_{W[s]}$
2. $i=s$ から，*0* の間
  $Q \leftarrow [2^{t_{i+1}-t_i}]Q$
  $W[i] > 0$ ならば，$Q \leftarrow Q + P_{W[i]}$
  $W[i] \leq 0$ ならば，$Q \leftarrow Q - P_{-W[i]}$
  $i \leftarrow i-1$
3. $Q \leftarrow [2^{t_0}]Q$
4. Q を出力する．

3.1.3 パラメータ推奨値

定義体の位数 p については、ECDLP に対する安全性の観点から、少なくとも 160 ビット以上が必要である。また、実装上の観点からは、有限体演算の高速化を図るために、$\mathbb{F}_p$ のサイズは計算機に適したサイズを利用することが望ましい．すなわち，$|p| = u$ =160, 192, 224 bits を推奨する．また，剰余算の高速化を図るために $p = 2^u - c$（c は小さい整数）が望ましい．

3.2 鍵共有スキーム

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].

3.2.1 鍵共有アルゴリズム

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.

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 K 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）としてもよい．これはアプリケーションに応じて設定可能である．

3.2.2 パラメータ推奨値

楕円曲線 $E/\mathbb{F}_p$ については 3.1.3 の推奨値を用いる．また各乱数のビット数は p_A と同じビット数が望ましい．

3.3 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，の 1.2.3 章に記載した性質を満たす一般的なものを利用する．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 [8].

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] D. M. Gordon, "Discrete logarithms in $GF(p)$ using the number field sieve", *SIAM J. on Discrete Math.*, **6**(1993), 124-138.

[6] A. Miyaji, T. Ono and H. Cohen "Efficient elliptic curve exponentiation", *Advances in Cryptology-Proceedings of ICICS'97*, Lecture Notes in Computer Science, **1334**(1997), Springer-Verlag, 282-290.

[7] H. Cohen, A. Miyaji and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates", *Advances in Cryptology-Proceedings of ASIACRYPT'98*, Lecture Notes in Computer Science, **1514**(1998), Springer-Verlag, 51-65.

[8] A. Menezes, P. Oorschot and S. Vanstone, "Handbook of Applied Cryptography", *CRC Press*, (1996).

[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] 岡本 龍明, 山本 博資, "現代暗号", **産業図書**, (1998).

[13] 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.

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

[15] 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.

[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.