PSEC 暗号仕様書

1　まえがき

本資料では、秘匿通信を目的とする公開鍵暗号方式（暗号スキーム）「PSEC 暗号」の仕様について述べる。「PSEC 暗号」は、「PSEC-1 暗号」、「PSEC-2 暗号」、「PSEC-3 暗号」という 3 つのバージョンを持つ。それぞれが、単独の暗号スキームとして利用できる。それぞれのバージョンは、安全性証明のため仮定や性能が異なるため、利用環境により適したバージョンを選択することを推奨する。

「PSEC 暗号」は、楕円 ElGamal 暗号関数（基本暗号プリミティブ）を 3 種類の変換法（FO-1 法, FO-2 法, OP 法）[7, 8, 12] を用いて、高い安全性をもつ暗号スキームに変換したものである。

なお、これら変換法では、補助関数としてハッシュ関数を利用するが、これらのハッシュ関数としては、出力サイズの条件の合致した任意の衝突困難な一方向性ハッシュ関数を用いることができるため、特定のハッシュ関数に限定する必要はない。本資料では、その一実現例として、SHA-1 に基づく方式 [3] を例示する（10章参照）。なお、同時に提出するテストデータ、サンプルプログラムではここで例示したハッシュ関数を用いて作られている。

FO-1 法を用いて楕円 ElGamal 暗号関数を変換した暗号スキームが PSEC-1 暗号であり、FO-2 法を用いて楕円 ElGamal 暗号関数を変換した暗号スキームが PSEC-2 暗号であり、OP 法を用いて楕円 ElGamal 暗号関数を変換した暗号スキームが PSEC-3 暗号である。

補助アルゴリズムとしては、上で述べたハッシュ関数以外に、楕円曲線のパラメータ生成アルゴリズムなどを必要とするが、それについては、 refhojo 章を参照されたい。

また、 PSEC-2, PSEC-3 では、任意の共通鍵暗号と併用することができる。同時に提出するテストデータ、サンプルプログラムでは、共通鍵暗号としてバーナム（one-time-pad）暗号が用いられている。

2　設計方針、設計基準

実用的暗号において（むしろ実用的であればあるほど）様々な実用的な利用環境での安全性を理論的に保証することは重要である。例えば、 1998 年に起こった SSL での PKCS#1（インターネットで最も広く使われている暗号ツール）に対する暗号攻撃の成功 [4] により、以下のような教訓が広く認識されるようになった。（1）実用的な利用環境においても能動的な攻撃（適応的選択暗号文攻撃）が十分に可能である、および（2） PKCS#1 のようなヒューリスティックな設計が危険である。そのような教訓を通じて、現在知られている暗号攻撃の分類で最強の意味の安全性を持つ（つまり、適応的選択暗号文攻撃に対して強秘匿である）[1] ことが（適当な仮定のもとで）証明できるような実用的な暗号が求められるようになった。

PSEC 暗号はそのような要求条件に答えるために作られた秘匿目的の公開鍵暗号方式である。そのような最強の意味の安全性を持った実用的な暗号を設計するために我々がとっ

た方針は、基本的安全性と実用的な性能を持った基本暗号（暗号プリミティブ）をその実用性を保持したままハッシュ関数を用いて最強の意味の安全性を持つような暗号に変換することである。このようなアプローチは、1994 年の Bellare, Rogaway の OAEP の提案以来、標準的なアプローチとされているものである。

PSEC 暗号の基本暗号（暗号プリミティブ）としては、楕円曲線上で構成された ElGamal 暗号関数（楕円 ElGamal 暗号関数）を用いる。また、ハッシュ関数を用いた変換方式としては、3 種類の方法（FO-1 法, FO-2 法, OP 法）[7, 8, 12] を用いる。それぞれの変換方式に応じて 3 つの暗号方式（暗号スキーム）が構成でき、それらをそれぞれ PSEC-1, PSEC-2, PSEC-3 と呼ぶ。

このように変換された暗号方式は、そこで用いられたハッシュ関数が理想的なランダム関数と仮定し（ランダムオラクルモデル）、かつ楕円 ElGamal 暗号関数の基本的な安全性を仮定すれば、最強の意味での安全性（適応的選択暗号文攻撃に対して強秘匿であること）が証明できる。

実用的なハッシュ関数が理想的なランダム関数という仮定は強い仮定であるが、もしこのような仮定の下で安全性の証明のついた方式に対して何らかの攻撃があれば、それはそのハッシュ関数にランダム関数に無い性質（非ランダム性）の一つが見つかったことを意味する。SHA のように注意深く作られたハッシュ関数の場合、最も基本的な非ランダム性である衝突性ですら見つけることが困難であると信じられている。従って、ランダムオラクルモデルの下で安全性の証明のついた方式を破ることは非常に困難であると信じられている。これが、ランダムオラクルモデルにおける安全性のパラダイムである[1]。

典型的なパラメータ設定（$|q| = 160$）において、PSEC 暗号は以下のような特長を持つ（詳しくは、「PSEC 暗号自己評価書」を参照されたい）。

適応性

1. PSEC-1 の適応性：共通鍵暗号の鍵（高々 128 ビット）の配送に適している。
2. PSEC-2 の適応性：任意の長さの共通鍵暗号鍵の配送、および長い平文の秘匿通信、特にカプセル的な利用方法（つまり、鍵配送とデータ配送が同期している）に適している。
3. PSEC-3 の適応性：任意の長さの共通鍵暗号鍵の配送、および長い平文の秘匿通信、特にカプセル的な利用方法（つまり、鍵配送とデータ配送が完全同期）のみならずセッション的利用方法（つまり、セッション開設時における鍵配送とそれ以降の該セッション開設中の共通鍵暗号によるデータ暗号化）に適している（8.4 節参照）。

安全性

1. PSEC-1 は、楕円曲線上の部分決定 Diffie-Hellman (EC-PDDH) 仮定とランダムオラクルモデルの下で、適応的選択暗号文攻撃に対して強秘匿であることが証明できる。

[1] Canetti らは、ランダムオラクルモデルで証明ができてもどのような計算量的仮定の下でもそのモデルを実現するハッシュ関数が構成できないような暗号プロトコルを示している [5] 。しかし、この例は極めて人工的に作られたものであり、自然に構成された方式に対しては、Canetti たちの結果に基づく懸念は特に意味がないように思われる。

2. PSEC-2（共通鍵暗号としてバーナム暗号を用いたもの）は、楕円曲線上の Diffie-Hellman (EC-DH) 仮定とランダムオラクルモデルの下で、適応的選択暗号文攻撃に対して強秘匿であることが証明できる。

3. PSEC-2（一般の共通鍵暗号を用いたもの）は、ここで用いる共通鍵暗号が受動的攻撃に対して安全である場合、楕円曲線上の Diffie-Hellman (EC-DH) 仮定とランダムオラクルモデルの下で、適応的選択暗号文攻撃に対して強秘匿であることが証明できる。

4. PSEC-3（共通鍵暗号としてバーナム暗号を用いたもの）は、楕円曲線上の Gap-Diffie-Hellman (EC-GDH) 仮定とランダムオラクルモデルの下で、適応的選択暗号文攻撃に対して強秘匿であることが証明できる。

5. PSEC-3（バーナム暗号以外の共通鍵暗号を用いたもの）は、ここで用いる共通鍵暗号が受動的攻撃に対して安全である場合、楕円曲線上の Gap-Diffie-Hellman (EC-GDH) 仮定とランダムオラクルモデルの下で、適応的選択暗号文攻撃に対して強秘匿であることが証明できる。さらに、セッション的利用方法をした場合でも、上と同じ仮定の下で、セッションにおける暗号全体として最強の意味の安全性（適応的選択暗号文攻撃に対して強秘匿／頑健）が証明される。

性能 典型的なパラメータ設定において、PSEC（PSEC-1, PSEC-2, PSEC-3）は、典型的な楕円曲線暗号（楕円 ElGamal 暗号）とほぼ同等の性能を有している。特に、PSEC-1, PSEC-2 は、暗号化処理速度が楕円 ElGamal 暗号とほぼ同じで、復号化処理速度が楕円 ElGamal 暗号の約 1/2 である。一方、PSEC-3 は、暗号化、復号化のいずれも楕円 ElGamal 暗号とほぼ同じである。（なお、楕円 ElGamal 暗号には、PSEC のような安全性の証明は無いことに注意。）

3 準備および記法

PSEC 暗号は、三つ組み $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ によって規定される。ここで、$\mathcal{G}$ は鍵生成演算であり、$\mathcal{E}$ は暗号化演算、$\mathcal{D}$ は復号化演算である。

PSEC-1 は楕円 ElGamal 暗号関数とハッシュ関数を用いた暗号方式であり、PSEC-2 および PSEC-3 は楕円 ElGamal 暗号関数とハッシュ関数、共通鍵暗号を用いた暗号方式である。

本仕様書では、以下のような表記を用いる。

- $a := b$: b の値を a に代入する。もしくは、a を b として定義する。
- $\mathbf{Z}$: 整数の集合。
- $\mathbf{Z}/n\mathbf{Z} := \{0, 1, \ldots, n-1\}$。
- A, B を集合とするとき、$A \backslash B := \{x \mid x \in A \wedge x \notin B\}$。
- $(\mathbf{Z}/n\mathbf{Z})^* := \{1, 2, \ldots, n-1\} \backslash \{x \mid \gcd(x, n) \neq 1\}$。

- $\mathbf{F}_q$: 位数が q の有限体。$q = p^n$ （p: 素数）のとき、$\mathbf{Z}/p\mathbf{Z}$ 上の最小多項式 $f(x) = f_0 + f_1 x + \cdot f_n x^n$ および基底（正規基底もしくは多項式基底）を定めると、$\mathbf{F}_q$ の要素 a は、$a = (a_{n-1}, a_{n-2}, \ldots, a_0)$ （$a_i \in \mathbf{Z}/p\mathbf{Z}$）で表現される。

- $\{0,1\}^*$ は、有限長のビット列の集合。$\{0,1\}^*$ を $\mathbf{B}$ と記すこともある。

- $\{0,1\}^i$ は、i ビット長のビット列の集合。$\{0,1\}^i$ を $\mathbf{B}_i$ と記すこともある。

- $a \in \mathbf{Z}$ のとき、$\mathbf{B}_i[a]$ は、

$$a = a_0 + 2a_1 + 2^2 a_2 + \cdots 2^{i-1} a_{i-1}$$

  となるようなビット列 $(a_{i-1}, a_{i-2}, \ldots, a_0) \in \mathbf{B}_i$ を意味する。

- $a \in \mathbf{F}_q$ で、$q = p^n$ （p: 素数）のとき、$a = (a_{n-1}, a_{n-2}, \ldots, a_0)$ （$a_i \in \mathbf{Z}/p\mathbf{Z}$）と表現されているとする。ここで、$2^{k-1} \leq p \leq 2^k - 1$ とする。このとき、$\mathbf{B}_{n \cdot k}[a]$ は、ビット列 $(\mathbf{B}_k[a_{n-1}]||\mathbf{B}_k[a_{n-2}]$ $\cdots \mathbf{B}_k[a_0]) \in \mathbf{B}_{n \cdot k}$ を意味する。ここで、$|\mathbf{F}_q| := n \cdot k$ とする。

- P を $\mathbf{F}_q$ 上の楕円曲線の点とする。$\mathbf{B}_{8+2 \cdot qLen}[P]$ は、ビット列 $(00000UCY || \mathbf{B}_{qLen}[x_P]$ $||\mathbf{B}_{qLen}[y_P]) \in \mathbf{B}_{8+2 \cdot qLen}$ を意味する（$qLen := |\mathbf{F}_q|$）。ここで、x_P および y_P は、P の x- 座標、y- 座標を意味し、(UCY) は、IEEE P1363 E.2.3.2 で定められたものを用いる [9]。

- $a := (a_{i-1}, a_{i-2}, \ldots, a_0) \in \mathbf{B}_i$ のとき、$\mathbf{I}[a]$ は、

$$b = a_0 + 2a_1 + 2^2 a_2 + \cdots 2^{i-1} a_{i-1}$$

  となるような整数 $b \in \mathbf{Z}$ を意味する。

- $a \in \mathbf{B}_i$ のとき、$|a| := i$.

- $a \equiv b \pmod{n}$ は、$a - b$ が p で割り切れることを意味する。$a := b \bmod n$ は、$a \in \mathbf{Z}/n\mathbf{Z}$ かつ $a \equiv b \pmod{n}$ を意味する。

- $a \in \mathbf{B}$ かつ $b \in \mathbf{B}$ のとき、$a||b$ は a と b の結合を意味する。例えば、$(0,1,0,0)||(1,1,0) = (0,1,0,0,1,1,0)$ となる。

- $X \in \mathbf{B}$ のとき、$[X]^k$ は、X の最上位 k ビットを意味する。

- $X \in \mathbf{B}$ のとき、$[X]_k$ は、X の最下位 k ビットを意味する。

- $a \in \mathbf{B}_i$ かつ $b \in \mathbf{B}_i$ のとき、$a \oplus b$ はビット毎の排他的論理和を意味する。（つまり、$a \oplus b \in \mathbf{B}_i$ となる。）

- $a \in \mathbf{B}_i$ かつ $b \in \mathbf{B}_j$ のとき（$i < j$）、$a \oplus b$ を計算する場合は、a の上位に 0 をパディングし、j ビットとして演算を行なう。例えば、$(101) \oplus (10100) := (00101) \oplus (10100) = (10001)$。item 点 Q を楕円曲線上の点としたとき、x_Q はその x- 座標を意味するものとする。

4 基本暗号関数（暗号プリミティブ）

PSEC 暗号は、基本暗号関数（暗号プリミティブ）として、楕円 ElGamal 暗号関数を用いている。以下に、楕円 ElGamal 暗号関数（鍵生成 $\mathcal{G}$、暗号 $\mathcal{E}$、復号 $\mathcal{D}$）を示す。

4.1 鍵生成: $\mathcal{G}$

$\mathcal{G}$ の入力と出力は以下の通りである。

[入力] 正整数であるセキュリティパラメータ $k \in \mathbf{Z}$.

[出力] 公開鍵 $(\mathbf{F}_q, a, b, p, P, W, pLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^7$ と秘密鍵 $s \in \mathbf{Z}$ の対。

k を入力とする鍵生成演算 $\mathcal{G}$ は以下の通りである。

- 以下の楕円曲線 (EC) パラメータを選ぶ（IEEE P1363 A.12.4 – 12.7, [9]）. 有限体 $\mathbf{F}_q$ のパラメータ、楕円曲線 E を定めるパラメータ a と b（いずれも $\mathbf{F}_q$ の要素）、E の点の数の約数である素数 p、位数が p である E 上の点 P。なお、楕円曲線のパラメータ (a, b) は、Weierstrass の標準形に基づく。ここで、$2^{k-1} \leq p \leq 2^k - 1$ とする。また、点 P は、$\mathbf{F}_q$ 上のアフィン座標で表される（つまり $P \in (\mathbf{F}_q)^2$）。

  なお、$q = p^n$ （p: 素数）のとき、$\mathbf{F}_q$ は、以下のように表現されているものとする。$((p, n), (f_n, \cdots, f_1, f_0), b) \in \mathbf{Z}^{n+4}$。ここで、$(f_n, \cdots, f_1, f_0)$ は、$\mathbf{Z}/p\mathbf{Z}$ 上の最小多項式 $f(x) = f_0 + f_1 x + \cdot f_n x^n$ を定め、b は、基底（$b = 1$: 正規基底；$b = 2$: 多項式基底）を定める。

  有限体 $\mathbf{F}_q$ は IEEE P1363 [9] で定められている。q が素数のもの、2 のべき乗のもののいずれを使っても良い。また、OEF[14] を使うことも出来る。（OEF については付録で述べる。）

- $s \in (\mathbf{Z}/p\mathbf{Z})^*$ をランダムに定め、E 上の点 $W = sP$ を計算する。

- $pLen := k$ とし、$qLen := |\mathbf{F}_q|$ とする。

注：楕円曲線 (EC) パラメータはシステムで定め、多くの利用者で共有しても良い。

4.2 暗号: $\mathcal{E}$

$\mathcal{E}$ の入力、出力は以下の通り。

[入力] 平文 $m \in \{0,1\}^{qLen}$ 及び公開鍵 $(\mathbf{F}_q, a, b, p, P, W, pLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^9$.

[出力] 暗号文 $c = (C_1, c_2) \in \mathbf{Z}^2 \times \{0,1\}^{qLen}$.

m と $(q, a, b, p, P, W, pLen, qLen)$, を入力とする 暗号演算 $\mathcal{E}$ は以下の通り。

- $r \in (\mathbf{Z}/p\mathbf{Z})^*$ をランダムに選ぶ。

- E 上の点 Q と C_1 を以下のように計算する。

$$Q := rW, \qquad C_1 := rP.$$

- c_2 を以下のように計算する。

$$c_2 := m \oplus \mathbf{B}_{qLen}[x_Q].$$

4.3 復号: $\mathcal{D}$

$\mathcal{D}$ の入力、出力は以下の通り。

[入力] 暗号文 $c = (C_1, c_2) \in \mathbf{Z}^2 \times \{0,1\}^{qLen}$ と公開鍵 $(\mathbf{F}_q, a, b, p, P, W, pLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^9$ および秘密鍵 $s \in \mathbf{Z}$。

[出力] 平文 $m \in \{0,1\}^{qLen}$。

c と $(q, a, b, p, P, W, pLen, qLen)$, s, を入力として、復号演算 $\mathcal{D}$ は以下の通り。

- E 上の点 $Q' := sC_1$ および $m' := c_2 \oplus \mathbf{B}_{qLen}[x_{Q'}]$ を計算する。
- m' を復号結果（平文 m）として出力する。

5 補助関数

以下に、本仕様において使用される補助関数を示す。

- (k ビット）乱数生成アルゴリズムもしくは（入力された整数 k に対して）k ビットの疑似乱数を生成出力するアルゴリズム：

  本仕様においては、鍵生成ならびに暗号化処理において、ある範囲からランダムな値を選択する必要があり、そのとき乱数もしくは疑似乱数を利用する。物理的に発生された乱数源を用いる場合には、そのための補助装置が必要となる。そのような補助装置を用いない場合には、疑似乱数を利用する。疑似乱数生成アルゴリズムの例は、IEEE P1363 [9] の Annex D.6 もしくは、[11] の 6 章に示されているようなアルゴリズムを利用することができる。

- (k ビット）素数生成アルゴリズム：

  （入力された整数 k に対して）k ビットの素数を出力するアルゴリズム。例として、（k ビットの）乱数を上記、乱数生成アルゴリズムにより生成し、Miller-Rabin の素数判定テストを t 回繰り返してパスするまで、乱数の生成を繰り返す。入力された乱数が Miller-Rabin テストを t 回繰り返してパスした場合、その乱数を素数と判定し出力するアルゴリズム。ここで、Miller-Rabin の素数判定テストは、IEEE P1363 [9] Annex A.15.1 の Miller-Rabin の素数判定テストの仕様に従うものとする。なお、$k(\geq 88)$ ビットの整数が、上記の Miller-Rabin の素数判定テストを $t(> 1)$ 回パスしたにもかかわらず、合成数である確率は、高々

  $$p_{k,t} = 2^{t+4} k (2^{-\sqrt{tk}}) \sqrt{\frac{k}{t}}$$

  であることが知られている [6]。

- ハッシュ関数：

  その典型的な構成例を 10 章で示す。そこに示されるハッシュ関数の構成例においては、SHA-1 を用いる。これは、NIST によって提案された、Secure Hash Algorithm (SHA-1) を指す。SHA-1 の仕様は、FIPS 180-1 standard [15] に従うものとする。

- 共通鍵暗号 $SymE$:

  PSEC-2, PSEC-3 では、共通鍵暗号 $SymE$ を用いる。$SymE$ を実現する一つの典型的方法は、バーナム暗号である。つまり、$SymEnc(key, ptext) := key \oplus ptext$, $SymDec(key, ctext) := key \oplus ctext$。ここで、$\oplus$ はビット毎の排他的論理和を意味する。

  $SymE$ の鍵 key の長さよりも大きいサイズの平文を暗号化する場合には、$SymE$ として任意のブロック暗号もしくはストリーム暗号を利用することができる。

- 楕円曲線のパラメータ生成アルゴリズム：

  素体上（$p > 3$）の楕円曲線のパラメータ生成アルゴリズムについては、IEEE P1363 [9] の Annex A.12.4, 12.5 に従うものとする。F_{2^m} 上の楕円曲線のパラメータ生成アルゴリズムについては、IEEE P1363 [9] の Annex A.12.6, 12.7 に従うものとする。

- 楕円曲線上の群演算アルゴリズム：

  素体上（$p > 3$）の楕円曲線の群演算アルゴリズムについては、IEEE P1363 [9] の Annex A.10.1 に従うものとする。F_{2^m} 上の楕円曲線の群演算アルゴリズムについては、IEEE P1363 [9] の Annex A.10.2 に従うものとする。

- 基本的な整数演算アルゴリズム：

  IEEE P1363 [9] の Annex A.1 – A.3 などに示されているアルゴリズムに従うものとする。

6 PSEC-1 暗号仕様

6.1 鍵生成: $\mathcal{G}$

$\mathcal{G}$ の入力と出力は以下の通りである。

[入力] 正整数であるセキュリティパラメータ $k \in \mathbf{Z}$.

[出力] 公開鍵 $(\mathbf{F}_q, a, b, p, P, W, hID, pLen, mLen, hLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{13}$ と秘密鍵 $s \in \mathbf{Z}$ の対。

k を入力とする鍵生成演算 $\mathcal{G}$ は以下の通りである。

- 以下の楕円曲線 (EC) パラメータを選ぶ（IEEE P1363 A.12.4 – 12.7, [9]）. 有限体 $\mathbf{F}_q$ のパラメータ、楕円曲線 E を定めるパラメータ a と b（いずれも $\mathbf{F}_q$ の要素）、E の点の数の約数である素数 p、位数が p である E 上の点 P。なお、楕円曲線のパラメータ (a, b) は、Weierstrass の標準形に基づく。ここで、$2^{k-1} \leq p \leq 2^k - 1$ とする。また、点 P は、$\mathbf{F}_q$ 上のアフィン座標で表される（つまり $P \in (\mathbf{F}_q)^2$）。

  なお、$q = p^n$ （p: 素数）のとき、$\mathbf{F}_q$ は、以下のように表現されているものとする。$((p, n), (f_n, \cdots, f_1, f_0), b) \in \mathbf{Z}^{n+4}$。ここで、$(f_n, \cdots, f_1, f_0)$ は、$\mathbf{Z}/p\mathbf{Z}$ 上の

最小多項式 $f(x) = f_0 + f_1 x + \cdot f_n x^n$ を定め、b は、基底（$b = 1$: 正規基底；$b = 2$: 多項式基底）を定める。

有限体 $\mathbf{F}_q$ は IEEE P1363 [9] で定められている。q が素数のもの、2 のべき乗のもののいずれを使っても良い。また、OEF[14] を使うことも出来る。（OEF については付録で述べる。）

- $s \in (\mathbf{Z}/p\mathbf{Z})^*$ をランダムに定め、E 上の点 $W = sP$ を計算する。
- $pLen := k, qLen := |\mathbf{F}_q|$ と定める。$mLen$ と $rLen$ を $mLen + rLen \leq qLen$ となるように定め、$hLen \leq pLen$ とする。
- 以下のようなハッシュ関数 h: $\{0,1\}^{mLen+rLen} \longrightarrow \{0,1\}^{hLen}$ を定め、その識別番号を hID とする。hID = 1 は、本仕様書 10 章で例示するハッシュ関数とする。

注：楕円曲線 (EC) パラメータと h はシステムで定め、多くの利用者で共有しても良い。

6.2 暗号: $\mathcal{E}$

$\mathcal{E}$ の入力、出力は以下の通り。

[入力] 平文 $m \in \{0,1\}^{mLen}$ 及び公開鍵 $(\mathbf{F}_q, a, b, p, P, W, hID,\ pLen,\ mLen,\ hLen,\ rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{13}$.

[出力] 暗号文 $c = (C_1, c_2) \in \mathbf{Z}^2 \times \{0,1\}^{qLen}$.

m と $(\mathbf{F}_q, a, b, p, P, W, h, pLen, mLen, rLen, qLen)$, を入力とする 暗号演算 $\mathcal{E}$ は以下の通り。

- $r \in \{0,1\}^{rLen}$ をランダムに選び、$t := h(m||r)$ を計算する。
- $\alpha := \mathbf{I}[t]$ とし、E 上の点 Q と C_1 を以下のように計算する。

$$Q := \alpha W, \qquad C_1 := \alpha P.$$

- c_2 を以下のように計算する。

$$c_2 := (m||r) \oplus \mathbf{B}_{qLen}[x_Q].$$

6.3 復号: $\mathcal{D}$

$\mathcal{D}$ の入力、出力は以下の通り。

[入力] 暗号文 $c = (C_1, c_2) \in \mathbf{Z}^2 \times \{0,1\}^{qLen}$ と公開鍵 $(\mathbf{F}_q, a, b, p, P, W, hID, pLen, mLen, hLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{13}$ および秘密鍵 $s \in \mathbf{Z}$。

[出力] 平文 $m \in \{0,1\}^{qLen}$ もしくは出力なし。

c と $(\mathbf{F}_q, a, b, p, P, W, hID, pLen, mLen, hLen, rLen, qLen)$, s, を入力として、復号演算 $\mathcal{D}$ は以下の通り。

- E 上の点 $Q' := sC_1$ および $u := c_2 \oplus \mathbf{B}_{qLen}[x_{Q'}]$ を計算し、さらに $u' = [u]_{mLen+rLen}$ とする。
- $\alpha' := \mathbf{I}[h(u')]$ とし、以下の式が成立するかどうかを検証する。

$$C_1 = \alpha' P.$$

- もし成立すれば、$[u']^{mLen}$ を平文として出力する。成立しない場合は、何も出力しない。

注：検査に不合格の場合、「不合格」を意味する特別な出力を行ってもよい。

7 PSEC-2 暗号仕様

7.1 鍵生成: $\mathcal{G}$

$\mathcal{G}$ の入力と出力は以下の通りである。

[入力] 正整数であるセキュリティパラメータ $k \in \mathbf{Z}$。

[出力] 公開鍵 $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{15}$, と秘密鍵 $s \in \mathbf{Z}$ の対。

k を入力とする鍵生成演算 $\mathcal{G}$ は以下の通りである。

- 以下の楕円曲線 (EC) パラメータを選ぶ（IEEE P1363 A.12.4 – 12.7, [9]）. 有限体 $\mathbf{F}_q$ のパラメータ、楕円曲線 E を定めるパラメータ a と b（いずれも $\mathbf{F}_q$ の要素）、E の点の数の約数である素数 p、位数が p である E 上の点 P。なお、楕円曲線のパラメータ (a, b) は、Weierstrass の標準形に基づく。ここで、$2^{k-1} \leq p \leq 2^k - 1$ とする。また、点 P は、$\mathbf{F}_q$ 上のアフィン座標で表される（つまり $P \in (\mathbf{F}_q)^2$）。

  なお、$q = p^n$ （p: 素数）のとき、$\mathbf{F}_q$ は、以下のように表現されているものとする。$((p, n), (f_n, \cdots, f_1, f_0), b) \in \mathbf{Z}^{n+4}$。ここで、$(f_n, \cdots, f_1, f_0)$ は、$\mathbf{Z}/p\mathbf{Z}$ 上の最小多項式 $f(x) = f_0 + f_1 x + \cdot f_n x^n$ を定め、b は、基底（$b = 1$: 正規基底；$b = 2$: 多項式基底）を定める。

  有限体 $\mathbf{F}_q$ は IEEE P1363 [9] で定められている。q が素数のもの、2 のべき乗のもののいずれを使っても良い。また、OEF[14] を使うことも出来る。（OEF については付録で述べる。）

- $s \in (\mathbf{Z}/p\mathbf{Z})^*$ をランダムに定め、E 上の点 $W = sP$ を計算する。$pLen := k$, $qLen := |\mathbf{F}_q|$ と定める。$rLen \leq qLen$ となるように $rLen$ を定める。
- 以下のような 2 つのハッシュ関数 h: $\{0,1\}^{mLen+rLen} \longrightarrow \{0,1\}^{hLen}$, g: $\{0,1\}^{rLen} \longrightarrow \{0,1\}^{gLen}$ を定め、それぞれの識別番号を HID および GID とする。HID = 1 および GID =1 は、本仕様書 10 章で例示するハッシュ関数とする。
- 共通鍵暗号 $SymE$ を定め、その識別番号を SEID とする。SEID = 1 は、バーナム暗号（one-time-pad）とする。ここで、$SymE = (SymEnc, SymDec)$ は共通

鍵 K をもつ共通鍵暗号・復号アルゴリズムの対である。K の長さは $gLen$ ビットである。暗号アルゴリズム $SymEnc$ は、鍵 K と平文 $X \in \mathbf{B}_{mLen}$ を入力として暗号文 $SymEnc(K,X)\mathbf{B}_{mLen}$ を出力する。復号アルゴリズム $SymDec$ は、鍵 K と暗号文 $Y\mathbf{B}_{mLen}$ を入力し平文 $SymDec(K,Y)\mathbf{B}_{mLen}$ を出力する。ここで、任意の鍵 K に対して、$SymEnc(K,\cdot)$ は、1 対 1 関数とする。

注：楕円曲線 (EC) パラメータと h, g はシステムで定め、多くの利用者で共有しても良い。

7.2 暗号: $\mathcal{E}$

$\mathcal{E}$ の入力、出力は以下の通り。

[入力] 平文 $m \in \{0,1\}^{mLen}$、および公開鍵 $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{15}$。

[出力] 暗号文 $c = (C_1, c_2, c_3) \in \mathbf{Z}^2 \times \{0,1\}^{qLen+mLen}$。

m と $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen)$ を入力とする暗号演算 $\mathcal{E}$ は以下の通り。

- $r \in \{0,1\}^{rLen}$ をランダムに選び、$g(r)$ および $t := h(m||r)$ を計算する。
- $\alpha := \mathbf{I}[t]$ とし、E 上の点 Q および C_1 を以下のように計算する。

$$Q := \alpha W, \qquad C_1 := \alpha P.$$

- c_2, c_3 を以下のように計算する。

$$c_2 := r \oplus \mathbf{B}_{qLen}[x_Q],$$

$$c_3 := SymEnc(g(r), m).$$

補足: $SymE$ を実現する一つの典型的方法は、バーナム暗号である。つまり、

$$SymEnc(key, ptext) := key \oplus ptext,$$

$$SymDec(key, ctext) := key \oplus ctext.$$

$SymE$ の鍵 $g(r)$ の長さ $gLen$ に比べて大きなサイズ $mLen$ の平文 M を暗号化する場合には、$SymE$ として適当な共通鍵暗号（ブロック暗号もしくはストリーム暗号）を利用する。

7.3 復号: $\mathcal{D}$

$\mathcal{D}$ の入力、出力は以下の通り。

[入力] 暗号文 $c = (C_1, c_2, c_3)$ と公開鍵 $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{15}$ および秘密鍵 $s \in \mathbf{Z}$。

[出力] 平文 $m \in \{0,1\}^{mLen}$ もしくは出力なし。

c と $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen)$, s を入力として、復号演算 $\mathcal{D}$ は以下の通り。

- E 上の点 $Q' := sC_1$ および $u := c_2 \oplus \mathbf{B}_{qLen}[x_{Q'}]$ を計算し、さらに $r' := [u]_{rLen}$ とする。
- $m' := SymDec(g(r'), c_3)$ を計算する。
- $\alpha' := \mathbf{I}[h(m'||r')]$ とし、以下の式が成立するかどうかを検証する。

$$C_1 = \alpha' P.$$

- もし成立すれば、m' を平文として出力する。成立しない場合は、何も出力しない。

注：検査に不合格の場合、「不合格」を意味する特別な出力を行ってもよい。

8 PSEC-3 暗号仕様

8.1 鍵生成: $\mathcal{G}$

$\mathcal{G}$ の入力と出力は以下の通りである。

[入力] 正整数であるセキュリティパラメータ $k \in \mathbf{Z}$。

[出力] 公開鍵 $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{15}$, と秘密鍵 $s \in \mathbf{Z}$ の対。

k を入力とする鍵生成演算 $\mathcal{G}$ は以下の通りである。

- 以下の楕円曲線 (EC) パラメータを選ぶ （IEEE P1363 A.12.4 – 12.7, [9]）. 有限体 $\mathbf{F}_q$ のパラメータ、楕円曲線 E を定めるパラメータ a と b （いずれも $\mathbf{F}_q$ の要素）、E の点の数の約数である素数 p、位数が p である E 上の点 P。なお、楕円曲線のパラメータ (a, b) は、 Weierstrass の標準形に基づく。ここで、$2^{k-1} \leq p \leq 2^k - 1$ とする。また、点 P は、$\mathbf{F}_q$ 上のアフィン座標で表される（つまり $P \in (\mathbf{F}_q)^2$）。

  なお、$q = p^n$ （p: 素数）のとき、$\mathbf{F}_q$ は、以下のように表現されているものとする。$((p, n), (f_n, \cdots, f_1, f_0), b) \in \mathbf{Z}^{n+4}$。ここで、$(f_n, \cdots, f_1, f_0)$ は、$\mathbf{Z}/p\mathbf{Z}$ 上の最小多項式 $f(x) = f_0 + f_1 x + \cdot f_n x^n$ を定め、b は、基底（$b = 1$: 正規基底 ; $b = 2$: 多項式基底）を定める。

  有限体 $\mathbf{F}_q$ は IEEE P1363 [9] で定められている。q が素数のもの、2 のべき乗のもののいずれを使っても良い。また、OEF[14] を使うことも出来る。（OEF については付録で述べる。）

- $s \in (\mathbf{Z}/p\mathbf{Z})^*$ をランダムに定め、E 上の点 $W = sP$ を計算する。$pLen := k$, $qLen := |\mathbf{F}_q|$ と定める。$rLen \leq qLen$ となるように $rLen$ を定める。
- 以下のようなハッシュ関数 h: $\{0,1\}^{8+4 \cdot qLen + 2 \cdot mLen} \longrightarrow \{0,1\}^{hLen}$, g: $\{0,1\}^{qLen} \longrightarrow \{0,1\}^{gLen}$ を定め、それぞれの識別番号を HID および GID とする。HID = 1 および GID =1 は、本仕様書 10 章で例示するハッシュ関数とする。

- 共通鍵暗号 $SymE$ を定め、その識別番号を SEID とする。SEID = 1 は、バーナム暗号 (one-time-pad) とする。ここで、$SymE = (SymEnc, SymDec)$ は共通鍵 K をもつ共通鍵暗号・復号アルゴリズムの対である。K の長さは $gLen$ ビットである。暗号アルゴリズム $SymEnc$ は、鍵 K と平文 $X \in \mathbf{B}_{mLen}$ を入力として暗号文 $SymEnc(K, X)\mathbf{B}_{mLen}$ を出力する。復号アルゴリズム $SymDec$ は、鍵 K と暗号文 $Y\mathbf{B}_{mLen}$ を入力し平文 $SymDec(K, Y)\mathbf{B}_{mLen}$ を出力する。ここで、任意の鍵 K に対して、$SymEnc(K, \cdot)$ は、1 対 1 関数とする。

注：楕円曲線 (EC) パラメータと h, g はシステムで定め、多くの利用者で共有しても良い。

8.2 暗号: $\mathcal{E}$

$\mathcal{E}$ の入力、出力は以下の通り。

[入力] 平文 $m \in \{0,1\}^{mLen}$、および公開鍵 $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{15}$。

[出力] 暗号文 $c = (C_1, c_2, c_3, c_4) \in \mathbf{Z}^2 \times \{0,1\}^{qLen+mLen+hLen}$。

m と $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen)$ を入力とする暗号演算 $\mathcal{E}$ は以下の通り。

- $u \in \{0,1\}^{qLen}$ および $r \in (\mathbf{Z}/p\mathbf{Z})^*$ を一様にランダムに選ぶ。
- E 上の点 C_1 および T を以下のように計算する。

$$C_1 := rP, \qquad T := rW.$$

を計算する。

- $$c_2 := u \oplus \mathbf{B}_{qLen}[x_T]$$

および $g(u)$ を計算する。

- c_3, c_4 を以下のように計算する。

$$c_3 := SymEnc(g(u), m),$$

$$c_4 := h(\mathbf{B}_{8+2\cdot qLen}[C_1]||c_2||c_3||u||m).$$

補足: $SymE$ を実現する一つの典型的方法は、バーナム暗号である。つまり、

$$SymEnc(key, ptext) := key \oplus ptext,$$

$$SymDec(key, ctext) := key \oplus ctext.$$

ここで、$\oplus$ はビット毎の排他的論理和を意味する。

$SymE$ の鍵 $g(u)$ の長さ $gLen$ に比べて大きなサイズ $mLen$ の平文 M を暗号化する場合には、$SymE$ として適当な共通鍵暗号（ブロック暗号もしくはストリーム暗号）を利用する。

8.3 復号: $\mathcal{D}$

$\mathcal{D}$ の入力、出力は以下の通り。

[入力] 暗号文 $c = (C_1, c_2, c_3, c_4) \in \mathbf{Z}^2 \times \{0,1\}^{qLen+mLen+hLen}$ と公開鍵 $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{15}$。および秘密鍵 $s \in \mathbf{Z}$。

[出力] 平文 $m \in \{0,1\}^{mLen}$ もしくは出力なし。

c と $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen)$, s を入力として、復号演算 $\mathcal{D}$ は以下の通り。

- E 上の点 $T' := sC_1$ および $u' := c_2 \oplus \mathbf{B}_{qLen}[x_{T'}]$ を計算する。
- $m' := SymDec(g(u'), c_3)$ を計算する。
- 以下の式が成立するかどうかを検証する。

$$c_4 = h(\mathbf{B}_{8+2\cdot qLen}[C_1]||c_2||c_3||u'||m').$$

- もし成立すれば、m' を平文として出力する。成立しない場合は、何も出力しない。

注：検査に不合格の場合、「不合格」を意味する特別な出力を行ってもよい。

8.4 参考：PSEC-3 暗号のセッション的利用方法

PSEC-3 暗号では、以下のようなセッション的利用方法が可能となる。

- 送信者は、$u \in \{0,1\}^{qLen}$ および $r \in (\mathbf{Z}/p\mathbf{Z})^*$ を一様にランダムに選ぶ。
- 送信者は、$C_1 := rP$, $T := rW$, $c_2 := u \oplus \mathbf{B}_{qLen}[x_T]$ および $K := g(u)$ を計算し、(C_1, c_2) を送信する。
- （秘密鍵を保持する）受信者は、(C_1, c_2) から u を復号化し、$K := g(u)$ を計算する。[鍵配送フェーズ終]
- 以降、平文 m_i $(i = 1, 2, \ldots,)$ に対して、送信者は、$c_{3,i} := SymEnc(K, m_i)$ および $c_{4,i} := H(\mathbf{B}_{8+2\cdot qLen}[C_1]||c_2||c_{3,i}||u||m_i)$ を計算し、$(c_{3,i}, c_{4,i})$ を送信する。
- 受信者は、K を用いて m_i を復号すると同時に $c_{4,i} = H(\mathbf{B}_{8+2\cdot qLen}[C_1]||c_2||c_{3,i}||u||m_i)$ の正当性検証を行なう。[暗号通信フェーズ終]

9 推奨パラメータ

以下では、PSEC-1, PSEC-2, PSEC-3 で共通となるパラメータの推奨値を示す。

- k: 160 以上（p のサイズを 160 ビット以上）
- $hLen$: 128 以上

以下、「PSEC 暗号自己評価書」で示した「性能評価」での代表的なパラメータ値を示す。

PSEC-1, PSEC-2, PSEC-3 において共通なパラメータとして、p および q のサイズを 160 bits とする。さらに、PSEC-1 の場合、$mLen = 128$, $rLen = 32$, $hLen = 160$、PSEC-2 （バーナム暗号利用）の場合、$rLen = 160$, $gLen = 128$, $hLen = 160$、また PSEC-3 （バーナム暗号利用）の場合、$qLen = 160$, $gLen = 128$, $pLen = 160$, $hLen = 128$、とする。

10 ハッシュ関数

既に述べたように、h （および g）が理想的なランダム関数であるとき PSEC の安全性を証明することができる。一方、実際にこの暗号アルゴリズムを実現する際には理想的なランダム関数の代わりに実用的な一方向性関数（例えば SHA など）を用いる。ここでは、SHA を用いて任意のサイズ（$hLen$ ビット）を出力する関数 H の一構成例を示す。この方法は、Bellare と Rogaway により示された方法である [3]。

$\mathrm{SHA}_\sigma(x)$ は x に SHA を適用して得られた 160 ビットの出力値を意味する。但し、160 ビットの “初期値” を $ABCDE = \sigma$ とする。$\mathrm{SHA}^l_\sigma(x)$ を $\mathrm{SHA}_\sigma(x)$ の先頭 l ビットとする。また、i を 32 ビットに符号化した値を $< i >$ とする。関数 H を以下のように定める。

$$H(x) := \mathrm{SHA}^{80}_\sigma(< 0 > ||x)||\mathrm{SHA}^{80}_\sigma(< 1 > ||x)||\cdots||\mathrm{SHA}^{L_l}_\sigma(< l > ||x),$$

ここで $l = \lfloor \frac{3k}{80} \rfloor$, かつ $L_l = hLen - 80l$.

参考文献

[1] Bellare, M., Desai, A., Pointcheval, D., and Rogaway, P.: Relations Among Notions of Security for Public-Key Encryption Schemes, Proc. of Crypto'98, LNCS 1462, Springer-Verlag, pp. 26–45 (1998).

[2] Bellare, M. and Rogaway, P.: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols, Proc. of the First ACM Conference on Computer and Communications Security, pp.62–73 (1993).

[3] Bellare, M. and Rogaway, P. : Optimal Asymmetric Encryption, Proc. of Eurocrypt'94, LNCS 950, Springer-Verlag pp.92-111 (1995).

[4] Bleichenbacher, D.: Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1, Proc. of Crypto'98, LNCS 1462, Springer-Verlag, pp. 1–12 (1998).

[5] Canetti, R., Goldreich, O. and Halevi, S.: The Random Oracle Methodology, Revisited, Proc. of STOC, ACM Press, pp.209–218 (1998).

[6] Damgård, I., Landrock, P., and Pomerance, C., "Average Case Error Estimates for the Strong Probable Prime Test", Matematics of Computation 61(1993), pp.177–194.

[7] Fujisaki, E., and Okamoto, T., "How to Enhance the Security of Public-Key Encryption at Minimum Cost", IEICE Trans. Fundamentals, Vol.E83-A, NO.1 January, pp.24–32, 2000.

[8] Fujisaki, E. and Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes, Proc. of Crypto'99, Springer-Verlag, LNCS 1666, pp. 535–554 (1999).

[9] IEEE P1363 Draft (D9), http://grouper.ieee.org/groups/1363/P1363/draft.html (1999).

[10] Joye, M., Quisquater, J.J., and Yung, M.: On the Power of Misbehaving Adversaries and Security Analysis of EPOC, Manuscript (February 2000).

[11] Menezes, A., van Oorschot, P., and Vanstone, S., "Handbook of Applied Cryptography", CRC Press, Boca Raton, Florida 1996.

[12] Okamoto, T. and Pointcheval, D.: OCAC: an Optimal Conversion for Asymmetric Cryptosystems, manuscript (2000).

[13] Okamoto, T. and Pointcheval, D.: PSEC–3: Provably Secure Elliptic Curve Encryption Scheme – V3, submission to P1363a (2000).

[14] Bailey, D. V. and Paar, C.: Optimal Extension Fields for Fast Arithmetic in Public-Key Algorithms, Proc. of Crypto'98, LNCS 1462, Springer-Verlag, pp.472-485 (1998).

[15] FIPS 180-1 "Secure Hash Standard", Federal Information Processing Standards Publication 180-1, U.S. Department of Commerce/N.I.S.T., National Technical Information Service, Springfield, Virginia, April 17 1995.

Appendix

A Definition of Optimal Extension Field(OEF)

binomial : a polynomial of the form $t^m - \omega$.

pseudo-Mersenne prime: a positive rational integer of the form $2^n \pm c, \log_2 \leq \lfloor n/2 \rfloor$.

Optimal Extension Field(OEF): a finite field $\mathbf{F}_{p^m}$ with p a pseudo-Mersenne prime and an irreducible binomial as the field polynomial.

B Finite Field Arithmetic

An odd characteristic extension field is a finite field whose number of elements is a power of an odd prime. If $m \geq 1$, then there is a unique field $\mathbf{F}_{p^m}$ with p^m elements. For purposes of conversion, the elements of $\mathbf{F}_{p^m}$ shall be represented in polynomial basis.

This representation is determined by choosing an irreducible polynomial $p(t)$ over $\mathbf{F}_p$. Then $\mathbf{F}_{p^m}$ is isomorphic to $\mathbf{F}_p[t]/p(t)$. This interpretation shall be the bit string formed by concatenating the values of the coefficients represented as integers. Thus the polynomial

$$a_{m-1}t^{m-1} + \ldots + a_2 t^2 + a_1 t + a_0$$

is represented by the bit string

$$(a_{m-1}, \ldots, a_2, a_1, a_0)$$

where each of the a_i are positive integers less than p, padded with leading 0 bits so that each a_i is represented with $\lceil \log_2 p \rceil$ bits.