Specification of PSEC: Provably Secure Elliptic Curve Encryption Scheme

1 Introduction

We describe an elliptic curve encryption scheme, PSEC (provably secure elliptic curve encryption scheme), which has three versions: PSEC-1, PSEC-2 and PSEC-3. PSEC-1 is a public-key encryption system that uses the elliptic curve ElGamal trapdoor function and a random function (hash function). PSEC-2 and PSEC-3 are public-key encryption systems that use the elliptic curve ElGamal trapdoor function, two random functions (hash functions) and a symmetric-key encryption (e.g., one-time padding and block-ciphers).

The encryption scheme described in this contribution is obtained by using three results on conversion techniques using random functions [10, 11, 17, 18].

2 Design Policy

One of the most important properties of public-key encryption is provable security. The strongest security notion in public-key encryption is that of non-malleability or semantical security against adaptive chosen-ciphertext attacks. Bellare, Desai, Pointcheval and Rogaway [3] show that semantical security against adaptive chosen-ciphertext attacks (IND-CCA2) is equivalent to (or sufficient for) the strongest security notion (NM-CCA2).

A promising way to construct a practical public-key encryption scheme semantically secure against adaptive chosen-ciphertext attacks (IND-CCA2) is to convert a primitive trap-door one-way function (such as RSA or ElGamal) by using *random functions*. Here, an ideally random function, the "random oracle", is assumed when proving the security, and the random function is replaced by a practical random-like function such as a one-way hash function (e.g., SHA-1 and MD5, etc.) when realizing it in practice. This approach was initiated by Bellare and Rogaway, and is called the *random oracle model* [4, 5].

Although security in the random oracle model cannot be guaranteed formally when a practical random-like function is used in place of the random oracle, this paradigm often yields much more efficient schemes than those in the *standard model* and gives an informal security guarantee.

Two typical primitives of the trap-door one-way function are deterministic one-way permutation (e.g. RSA function) and probabilistic one-way function (e.g., ElGamal and Okamoto-Uchiyama functions).

Bellare and Rogaway presented a generic and efficient way to convert a trap-door one-way permutation to an IND-CCA2 secure scheme in the random oracle model. (The scheme created in this way from the RSA function is often called OAEP.) However, their method cannot be applied to probabilistic trap-door one-way functions such as ElGamal.

Recently the authors, Fujisaki and Okamoto [10, 11], and Okamoto and Pointcheval [17, 18] realized three generic and efficient measures to convert a probabilistic trap-door one-way function to an IND-CCA2 secure scheme in the random oracle model. One is conversion from a semantically secure (IND-CPA) trap-door one-way function to an IND-CCA2 secure scheme [10]. Another is from a trap-door one-way (OW-CPA) function and a symmetric-key encryption (including one-time padding) to an IND-CCA2 secure scheme [11]. The other is from a gap-trap-door one-way (OW-CPA) function and a symmetric-key encryption (including one-time padding) to an IND-CCA2 secure scheme [17, 18]. The latter two conversions can guarantee the total security of the public-key encryption system in combination with a symmetric-key encryption scheme.

PSEC has several outstanding properties as follows:

1. PSEC-1 is semantically secure or non-malleable against chosen ciphertext attacks (IND-CCA2 or NM-CCA2) in the random oracle model under the elliptic curve decision Diffie-Hellman (EC-DDH) assumption.

2. PSEC-2 with one-time padding (PSEC-2-OTP) is semantically secure or non-malleable against chosen ciphertext attacks (IND-CCA2 or NM-CCA2) in the random oracle model under the elliptic curve Diffie-Hellman (EC-DH) assumption.

3. PSEC-2 with symmetric encryption (PSEC-2-SymE) is semantically secure or non-malleable against chosen ciphertext attacks (IND-CCA2 or NM-CCA2) in the random oracle model under the elliptic curve Diffie-Hellman (EC-DH) assumption, if the underlying symmetric encryption is secure against passive attacks.

4. PSEC-3 with one-time padding (PSEC-3-OTP) is semantically secure or non-malleable against chosen ciphertext attacks (IND-CCA2 or NM-CCA2) in the random oracle model under the elliptic curve gap Diffie-Hellman (EC-Gap-DH) assumption.

5. PSEC-3 with symmetric encryption (PSEC-3-SymE) is semantically secure or non-malleable against chosen ciphertext attacks (IND-CCA2 or NM-CCA2) in the random oracle model under the elliptic curve gap Diffie-Hellman (EC-gap-DH) assumption, if the underlying symmetric encryption is secure against passive attacks.

6. If practical hash functions (e.g., SHA and MD5) are used as the underlying random functions, PSEC is almost as efficient as the elliptic curve ElGamal scheme. (Note that the elliptic curve ElGamal scheme is not secure against a chosen ciphertext attack.)

3 Notations

PSEC is specified by triplet $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, where $\mathcal{G}$ is the key generation operation, $\mathcal{E}$ the encryption operation, and $\mathcal{D}$ the decryption operation.

We have three versions of PSEC: PSEC-1, PSEC-2 and PSEC-3. PSEC-1 is designed for key-distribution and PSEC-2 and PSEC-3 are designed for both usages: the combination of key-distribution and encrypted data transfer, as well as distribution of a longer key under limited public-key size.

In this specification, we use following notations.

- $a := b$: the value of b is substituted to a, or a is defined as b.
- $\mathbf{Z}$: the set of integers.
- $\mathbf{Z}/n\mathbf{Z} := \{0, 1, \ldots, n-1\}$.
- Let A, B be sets. $A \backslash B := \{x \mid x \in A \wedge x \notin B\}$.
- $(\mathbf{Z}/n\mathbf{Z})^* := \{1, 2, \ldots, n-1\} \backslash \{x \mid \gcd(x, n) \neq 1\}$.
- Let $\mathbf{F}_q$ be a finite field with q elements, where $q = {q_0}^n$ (q_0: prime). When a minimal polynomial over $\mathbf{Z}/p\mathbf{Z}$, $f(x) = f_0 + f_1 x + \cdot f_n x^n$, and basis (normal basis or polynomial basis) are fixed, en element of $\mathbf{F}_q$ is expressed by $a = (a_{n-1}, a_{n-2}, \ldots, a_0)$ ($a_i \in \mathbf{Z}/p\mathbf{Z}$).
- $\{0,1\}^*$ is the set of finite strings. $\{0,1\}^*$ is also denoted by $\mathbf{B}$.
- $\{0,1\}^i$ is the set of i bit length bit strings. $\{0,1\}^i$ is also denoted by $\mathbf{B}_i$.
- Let $a \in \mathbf{Z}$. $\mathbf{B}_i[a]$ denotes a bit string $(a_{i-1}, a_{i-2}, \ldots, a_0) \in \mathbf{B}_i$ such that

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

  .
- When $a \in \mathbf{F}_q$ and $q = {q_0}^n$ (q_0: prime), let a be represented by $a = (a_{n-1}, a_{n-2}, \ldots, a_0)$ ($a_i \in \mathbf{Z}/p\mathbf{Z}$), where $2^{k-1} \leq p \leq 2^k - 1$. Then $\mathbf{B}_{n \cdot k}[a]$ denotes a bit string, $(\mathbf{B}_k[a_{n-1}] || \mathbf{B}_k[a_{n-2}] \cdots \mathbf{B}_k[a_0]) \in \mathbf{B}_{n \cdot k}$, where $|\mathbf{F}_q| := n \cdot k$.
- Let P be a point on an elliptic curve over $\mathbf{F}_q$. $\mathbf{B}_{8+2 \cdot qLen}[P]$ denotes a bit string $(00000UCY || \mathbf{B}_{qLen}[x_P] \; || \mathbf{B}_{qLen}[y_P]) \in \mathbf{B}_{8+2 \cdot qLen}$ ($qLen := |\mathbf{F}_q|$). Here, x_P and y_P denotes the x-coordinate and y-coordinate of P, and (UCY) follows the definition of IEEE P1363 E.2.3.2 [13].
- Let $a := (a_{i-1}, a_{i-2}, \ldots, a_0) \in \mathbf{B}_i$. $\mathbf{I}[a]$ denotes an integer $b \in \mathbf{Z}$ such that

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

  .
- If $a \in \mathbf{B}_i$, $|a| := i$.
- $a \equiv b \pmod{n}$ means $a - b$ is divided by n. $a := b \bmod n$ denotes $a \in \mathbf{Z}/n\mathbf{Z}$ and $a \equiv b \pmod{n}$.

- Let $a \in \mathbf{B}$ and $b \in \mathbf{B}$. $a||b$ denotes the concatenation of a and b. For example, $(0,1,0,0)||(1,1,0) = (0,1,0,0,1,1,0)$.
- Let $X \in \mathbf{B}$. $[X]^k$ denotes the most k significant bits of X.
- Let $X \in \mathbf{B}$. $[X]_k$ denotes the least k significant bits of X.
- Let $a \in \mathbf{B}_i$ and $b \in \mathbf{B}_i$. $a \oplus b$ means the bit-wise exclusive-or operation. (i.e., $a \oplus b \in \mathbf{B}_i$.)
- Let $a \in \mathbf{B}_i$ and $b \in \mathbf{B}_j$ $(i < j)$. When $a \oplus b$ is calculated, '0' bits are padded to the upper of a and the resulting string is j bit long, then the $\oplus$ operation is executed. For example, $(101) \oplus (10100) := (00101) \oplus (10100) = (10001)$.
- Let Q be a point on an elliptic curve, x_Q denotes the x-coordinate of Q.

4 Primitive Encryption Function

PSEC employs the elliptic curve ElGamal encryption function as a primitive encryption function.

4.1 Key Generation: $\mathcal{G}$

The input and output of $\mathcal{G}$ are as follows:

[Input] Security parameter $k \in \mathbf{Z}$.

[Output] A pair of public-key, $(\mathbf{F}_q, a, b, p, P, W, pLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^7$, and secret-key $s \in \mathbf{Z}$.

The operation of $\mathcal{G}$, on input k, is as follows:

- Choose elliptic curve (EC) domain parameters, q for a finite field $\mathbf{F}_q$; two elliptic curve coefficients a and b, elements of $\mathbf{F}_q$, that defines an elliptic curve E; a positive prime integer p dividing the number of points on E; and a curve point P with order p. (See IEEE P1363 A.12.4 – 12.7, [13]). Here the parameter (a, b) is based on the Weierstrass standard form, $2^{k-1} \leq p \leq 2^k - 1$, and P is represented by the affine coordinates (i.e., $P \in (\mathbf{F}_q)^2$).

  When $q = {q_0}^n$ (q_0: prime), $\mathbf{F}_q$ is represented by $((q_0, n), (f_n, \cdots, f_1, f_0), b) \in \mathbf{Z}^{n+4}$, where $(f_n, \cdots, f_1, f_0)$ denoted the minimum polynomial over $\mathbf{Z}/q_0\mathbf{Z}$ $f(x) = f_0 + f_1 x + \cdot f_n x^n$, and b denotes the type of basis ($b = 1$: normal basis; $b = 2$: polynomial basis).
- Choose $s \in (\mathbf{Z}/p\mathbf{Z})^*$ randomly, and calculates a point W on E, where $W = sP$.
- Set $pLen := k$, and $qLen := |q|$.

Note: The EC domain parameters are used in every EC primitive and scheme and an implicit component of every EC key. h can be fixed by the system and shared by many users.

4.2 Encryption: $\mathcal{E}$

The input and output of $\mathcal{E}$ are as follows:

[Input] Plaintext $m \in \{0,1\}^{mLen}$ along with public-key $(\mathbf{F}_q, a, b, p, P, W,\ pLen,\ qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^9$.

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

The operation of $\mathcal{E}$, on input m and $(\mathbf{F}_q, a, b, p, P, W, hID,\ pLen,\ mLen,\ hLen,\ rLen, qLen)$ is as follows:

- Select $r \in \{0,1\}^{rLen}$ uniformly.
- Compute Q and C_1, such that

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

- Compute c_2:

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

4.3 Decryption: $\mathcal{D}$

The input and output of $\mathcal{D}$ are as follows:

[Input] Ciphertext $c = (C_1, c_2) \in \mathbf{Z}^2 \times \{0,1\}^{qLen}$ along with public-key $(\mathbf{F}_q, a, b, p, P, W, pLen,\ qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^9$ and secret-key $s \in \mathbf{Z}$.

[Output] Plaintext $m \in \{0,1\}^{qLen}$.

The operation of $\mathcal{D}$, on input c along with $(q, a, b, p, P, W, pLen,\ qLen)$ and s, is as follows:

- Compute $Q' := sC_1$, a point on E, and $m' := c_2 \oplus \mathbf{B}_{qLen}[x_{Q'}]$.
- Output m' as decrypted plaintext.

5 Auxiliary Functions

In this section, we show auxiliary functions we use in this specification.

- (k bit) pseudo-random number generator.
- (k bit) prime number generator.
- Hash function.
- Symmetric encryption algorithm $SymE$.
- Elliptic curve cryptosystem generating algorithm.
- Elliptic curve group arithmetic algorithm.
- Primitive integer arithmetic algorithm.

6 Specification of PSEC-1

6.1 Key Generation: $\mathcal{G}$

The input and output of $\mathcal{G}$ are as follows:

[Input] Security parameter $k \in \mathbf{Z}$.

[Output] A pair of public-key, $(\mathbf{F}_q, a, b, p, P, W, hID, pLen, mLen, hLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{13}$, and secret-key $s \in \mathbf{Z}$.

The operation of $\mathcal{G}$, on input k, is as follows:

- Choose elliptic curve (EC) domain parameters, q for a finite field $\mathbf{F}_q$; two elliptic curve coefficients a and b, elements of $\mathbf{F}_q$, that defines an elliptic curve E; a positive prime integer p dividing the number of points on E; and a curve point P with order p. (See IEEE P1363 A.12.4 – 12.7, [13]). Here the parameter (a, b) is based on the Weierstrass standard form, $2^{k-1} \leq p \leq 2^k - 1$, and P is represented by the affine coordinates (i.e., $P \in (\mathbf{F}_q)^2$).

  When $q = {q_0}^n$ (q_0: prime), $\mathbf{F}_q$ is represented by $((q_0, n), (f_n, \cdots, f_1, f_0), b) \in \mathbf{Z}^{n+4}$, where $(f_n, \cdots, f_1, f_0)$ denoted the minimum polynomial over $\mathbf{Z}/q_0\mathbf{Z}$ $f(x) = f_0 + f_1 x + \cdot f_n x^n$, and b denotes the type of basis ($b = 1$: normal basis; $b = 2$: polynomial basis).

- Choose $s \in (\mathbf{Z}/p\mathbf{Z})^*$ randomly, and calculates a point W on E, where $W = sP$.

- Set $pLen := k$, and $qLen := |q|$. Set $mLen$ and $rLen$ such that $mLen + rLen \leq qLen$. Set $hLen \leq pLen$.

- Select a (hash) function h: $\{0,1\}^{mLen+rLen} \longrightarrow \{0,1\}^{hLen}$, and its identifier is hID.

Note: The EC domain parameters are used in every EC primitive and scheme and an implicit component of every EC key. h can be fixed by the system and shared by many users.

6.2 Encryption: $\mathcal{E}$

The input and output of $\mathcal{E}$ are as follows:

[Input] Plaintext $m \in \{0,1\}^{mLen}$ along with public-key $(\mathbf{F}_q, a, b, p, P, W, hID, pLen, mLen, hLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{13}$.

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

The operation of $\mathcal{E}$, on input m and $(\mathbf{F}_q, a, b, p, P, W, hID, pLen, mLen, hLen, rLen, qLen)$ is as follows:

- Select $r \in \{0,1\}^{rLen}$ uniformly, and compute $t := h(m||r)$.

- Set $\alpha := \mathbf{I}[t]$, and compute Q and R, points on C_1, such that

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

- Compute c_2:

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

6.3 Decryption: $\mathcal{D}$

The input and output of $\mathcal{D}$ are as follows:

[Input] Ciphertext $c = (C_1, c_2) \in \mathbf{Z}^2 \times \{0,1\}^{qLen}$ along with public-key $(\mathbf{F}_q, a, b, p, P, W, hID, pLen, mLen, hLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{13}$ and secret-key $s \in \mathbf{Z}$.

[Output] Plaintext $m \in \{0,1\}^{qLen}$ or null string.

The operation of $\mathcal{D}$, on input c along with $(\mathbf{F}_q, a, b, p, P, W, hID, pLen, mLen, hLen, rLen, qLen)$ and s, is as follows:

- Compute $Q' := sC_1$, a point on E, and $u := c_2 \oplus \mathbf{B}_{qLen}[x_{Q'}]$. Set $u' = [u]_{mLen+rLen}$.
- Check whether the following equation holds or not:

$$C_1 = \alpha' P,$$

  where $\alpha' := \mathbf{I}[h(u')]$.

- If it holds, output $[u']^{mLen}$ as decrypted plaintext. Otherwise, output null string.

7 Specification of PSEC-2

7.1 Key Generation: $\mathcal{G}$

The input and output of $\mathcal{G}$ are as follows:

[Input] Security parameter $k \in \mathbf{Z}$.

[Output] A pair of public-key, $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{15}$, and secret-key $s \in \mathbf{Z}$.

The operation of $\mathcal{G}$, on input k, is as follows:

- Choose elliptic curve (EC) domain parameters, q for a finite field $\mathbf{F}_q$; two elliptic curve coefficients a and b, elements of $\mathbf{F}_q$, that defines an elliptic curve E; a positive prime integer p dividing the number of points on E; and a curve point P with order p. (See IEEE P1363 A.12.4 – 12.7, [13]). Here the parameter (a, b) is based on the Weierstrass standard form, $2^{k-1} \leq p \leq 2^k - 1$, and P is represented by the affine coordinates (i.e., $P \in (\mathbf{F}_q)^2$).

  When $q = {q_0}^n$ (q_0: prime), $\mathbf{F}_q$ is represented by $((q_0, n), (f_n, \cdots, f_1, f_0), b) \in \mathbf{Z}^{n+4}$, where $(f_n, \cdots, f_1, f_0)$ denoted the minimum polynomial over $\mathbf{Z}/q_0\mathbf{Z}$ $f(x) = f_0 + f_1 x + \cdot f_n x^n$, and b denotes the type of basis ($b = 1$: normal basis; $b = 2$: polynomial basis).

- Choose $s \in (\mathbf{Z}/p\mathbf{Z})^*$ randomly, and calculates a point W on E, where $W = sP$.
- Set $pLen := k$, and $qLen := |q|$. Set $rLen$ such that $rLen \leq qLen$.

- Select two (hash) functions, h: $\{0,1\}^{mLen+rLen} \longrightarrow \{0,1\}^{hLen}$, g: $\{0,1\}^{rLen} \longrightarrow \{0,1\}^{gLen}$, and their identifiers are hID and gID respectively.

- Let $SymE = (SymEnc, SymDec)$ be a pair of symmetric-key encryption and decryption algorithms with symmetric-key K, where the length of K is $gLen$. The identifier of $SymE$ is SEID. Let SEID = 1 denotes that $SymE$ is the one-time-pad.

  Encryption algorithm $SymEnc$ takes key K and plaintext X, and returns ciphertext $SymEnc(K, X)$. Decryption algorithm $SymDec$ takes key K and ciphertext Y, and returns plaintext $SymDec(K, Y)$. Here we assume that for any key K, function $SymEnc(K, \cdot)$ is one-to-one and onto.

Note: The EC domain parameters are used in every EC primitive and scheme and an implicit component of every EC key. h can be fixed by the system and shared by many users.

7.2 Encryption: $\mathcal{E}$

The input and output of $\mathcal{E}$ are as follows:

[Input] Plaintext $m \in \{0,1\}^{mLen}$ along with public-key $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{15}$.

[Output] Ciphertext $c = (C_1, c_2, c_3) \in \mathbf{Z}^2 \times \{0,1\}^{qLen+mLen}$.

The operation of $\mathcal{E}$, on input m, $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen)$ is as follows:

- Select $r \in \{0,1\}^{rLen}$ uniformly, and compute $g(r)$ and $t := h(m||r)$.

- Set $\alpha := \mathbf{I}[t]$, and compute Q and C_1, points on E, such that

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

- Compute c_2 and c_3 as follows:

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

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

Remark: A typical way to realize $SymE$ is one-time padding.
That is, $SymEnc(key, ptext) := key \oplus ptext$, and $SymDec(key, ctext) := key \oplus ctext$, where $\oplus$ denotes the bit-wise exclusive-or operation.

When $mLen$ is longer than $gLen$, we use an appropriate symmetric encryption (block cipher or stream cipher) rather than one-time padding.

7.3 Decryption: $\mathcal{D}$

The input and output of $\mathcal{D}$ are as follows:

[Input] Ciphertext $c = (C_1, c_2, c_3)$ along with public-key $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{15}$ secret-key $s \in \mathbf{Z}$.

[Output] Plaintext $m \in \{0,1\}^{mLen}$ or null string.

The operation of $\mathcal{D}$, on input c along with $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen)$ and s, is as follows:

- Compute $Q' := sC_1$, a point on E, and $u := c_2 \oplus \mathbf{B}_{qLen}[x_{Q'}]$. Set $r' := [u]_{rLen}$.
- Compute $m' := SymDec(g(r'), c_3)$.
- Check whether the following equation holds or not:

$$C_1 = \alpha' P,$$

  where $\alpha' := \mathbf{I}[h(m'||r')]$.

- If it holds, output m' as the decrypted plaintext. Otherwise, output null string.

8 Specification of PSEC-3

8.1 Key Generation: $\mathcal{G}$

The input and output of $\mathcal{G}$ are as follows:

[Input] Security parameter $k \in \mathbf{Z}$.

[Output] A pair of public-key, $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{15}$, and secret-key $s \in \mathbf{Z}$.

The operation of $\mathcal{G}$, on input k, is as follows:

- Choose elliptic curve (EC) domain parameters, q for a finite field $\mathbf{F}_q$; two elliptic curve coefficients a and b, elements of $\mathbf{F}_q$, that defines an elliptic curve E; a positive prime integer p dividing the number of points on E; and a curve point P with order p. (See IEEE P1363 A.12.4 – 12.7, [13]). Here the parameter (a, b) is based on the Weierstrass standard form, $2^{k-1} \leq p \leq 2^k - 1$, and P is represented by the affine coordinates (i.e., $P \in (\mathbf{F}_q)^2$).

  When $q = {q_0}^n$ (q_0: prime), $\mathbf{F}_q$ is represented by $((q_0, n), (f_n, \cdots, f_1, f_0), b) \in \mathbf{Z}^{n+4}$, where $(f_n, \cdots, f_1, f_0)$ denoted the minimum polynomial over $\mathbf{Z}/q_0\mathbf{Z}$ $f(x) = f_0 + f_1 x + \cdot f_n x^n$, and b denotes the type of basis ($b = 1$: normal basis; $b = 2$: polynomial basis).

- Choose $s \in (\mathbf{Z}/p\mathbf{Z})^*$ randomly, and calculates a point W on E, where $W = sP$.

- Set $pLen := k$, and $qLen := |q|$. Set $rLen$ such that $rLen \leq qLen$.
- Select two (hash) functions, h: $\{0,1\}^{8+4\cdot qLen+2\cdot mLen} \longrightarrow \{0,1\}^{hLen}$, g: $\{0,1\}^{qLen} \longrightarrow \{0,1\}^{gLen}$ and their identifiers are hID and gID respectively.
- Let $SymE = (SymEnc, SymDec)$ be a pair of symmetric-key encryption and decryption algorithms with symmetric-key K, where the length of K is $gLen$. The identifier of $SymE$ is SEID. Let SEID = 1 denotes that $SymE$ is the one-time-pad.

  Encryption algorithm $SymEnc$ takes key K and plaintext X, and returns ciphertext $SymEnc(K, X)$. Decryption algorithm $SymDec$ takes key K and ciphertext Y, and returns plaintext $SymDec(K, Y)$. Here we assume that for any key K, function $SymEnc(K, \cdot)$ is one-to-one and onto.

Note: The EC domain parameters are used in every EC primitive and scheme and an implicit component of every EC key. h can be fixed by the system and shared by many users.

8.2 Encryption: $\mathcal{E}$

The input and output of $\mathcal{E}$ are as follows:

[Input] Plaintext $m \in \{0,1\}^{mLen}$ along with public-key $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{15}$.

[Output] Ciphertext $c = (C_1, c_2, c_3, c_4) \in \mathbf{Z}^2 \times \{0,1\}^{qLen+mLen+hLen}$.

The operation of $\mathcal{E}$, on input m, $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen)$ is as follows:

- Select $u \in \{0,1\}^{qLen}$ and $r \in (\mathbf{Z}/p\mathbf{Z})^*$ uniformly.
- Compute C_1 and T, points on E, such that

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

- Compute

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

  and $g(u)$.
- Compute c_3 and c_4 as follows:

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

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

Remark: A typical way to realize $SymE$ is one-time padding.
That is, $SymEnc(key, ptext) := key \oplus ptext$, and $SymDec(key, ctext) := key \oplus ctext$, where $\oplus$ denotes the bit-wise exclusive-or operation.

When $mLen$ is longer than $gLen$, we use an appropriate symmetric encryption (block cipher or stream cipher) rather than one-time padding.

8.3 Decryption: $\mathcal{D}$

The input and output of $\mathcal{D}$ are as follows:

[Input] Ciphertext $c = (C_1, c_2, c_3, c_4) \in \mathbf{Z}^2 \times \{0,1\}^{qLen+mLen+hLen}$ along with public-key $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen) \in \mathbf{Z}^{n+4} \times \mathbf{Z}^{15}$ secret-key $s \in \mathbf{Z}$.

[Output] Plaintext $m \in \{0,1\}^{mLen}$ or null string.

The operation of $\mathcal{D}$, on input c along with $(\mathbf{F}_q, a, b, p, P, W, hID, gID, SEID, pLen, hLen, gLen, rLen, qLen)$ and s, is as follows:

- Compute $T' := sC_1$, a point on E, and $u' := c_2 \oplus \mathbf{B}_{qLen}[x_{T'}]$.
- Compute $m' := SymDec(g(u'), c_3)$.
- Check whether the following equation holds or not:

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

- If it holds, output m' as the decrypted plaintext. Otherwise, output null string.

8.4 Session Like Method for PSEC-3 Encryption

We can use PSEC-3 as following session like method.

- Sender chooses uniform randomly $u \in \{0,1\}^{qLen}$ and $r \in (\mathbf{Z}/p\mathbf{Z})^*$
- Sender computes $C_1 := rP$, $T := rW$, $c_2 := u \oplus \mathbf{B}_{qLen}[x_T]$ and $K := g(u)$, and sends (C_1, c_2)
- Receiver decrypts u from (C_1, c_2), and computes $K := g(u)$. [key sending phase finished]
- For each plaintext m_i $(i = 1, 2, \ldots,)$, Sender computes $c_{3,i} := SymEnc(K, m_i)$ and $c_{4,i} := H(\mathbf{B}_{8+2\cdot qLen}[C_1]||c_2||c_{3,i}||u||m_i)$, and sends $(c_{3,i}, c_{4,i})$.
- Sender decrypts m_i by using K, and checks $c_{4,i} = H(\mathbf{B}_{8+2\cdot qLen}[C_1]||c_2||c_{3,i}||u||m_i)$ [cipher communication phase finished]

9 Recommended Parameters

For PSEC-1/2/3, the security parameter, k, should be at least 160, and $hLen$ should be at least 128.

Here we will show a typical case of parameters employed in our self-evaluation document.

The field size ($qLen$) and the size of the order of the base point ($pLen$) are 160 bits. For PSEC-1, random string length ($rLen$) is 32 bits, and hashed value length ($hLen$) is 160 bits. As for PSEC-2-OTP, random string length ($rLen$) is 160 bits, hashed value lengths ($gLen$ and $hLen$) are 128 bits and 160 bits. PSEC-3-OTP assumes that hashed value lengths ($gLen$ and $hLen$) are 128 bits.

10 Hash Function

We can use any random-like one-way functions H and G for PSEC. PSEC can be proven to be secure if H and G are ideal random functions, while no formal security is guaranteed if they are practical random-like one-way functions.) In this subsection we will show a typical construction of function H with $hLen > 160$ out of SHA (NIST Secure Hash Algorithm), which was suggested by Bellare and Rogaway [5].

We denote by $\mathrm{SHA}_\sigma(x)$ the 160-bit result of SHA applied to x, except that the 160-bit "starting value" in the algorithm description is taken to be $ABCDE = \sigma$. Let $\mathrm{SHA}^l_\sigma(x)$ denote the first l-bits of $\mathrm{SHA}_\sigma(x)$. Fix the notation $< i >$ for i encoded as a binary 32-bit word. We define the function H as:

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

where $l = \lfloor \frac{3k}{80} \rfloor$, and $L_l = hLen - 80l$.

References

[1] Abdalla, M., Bellare, M. and Rogaway, P.: DHES: An Encryption Scheme Based on the Diffie-Hellman Problem, Submission to IEEE P1363a (1998, August)

[2] Ajtai, M. and Dwork, C.: A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence, Proc. of STOC'97, pp. 284-293 (1997).

[3] 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).

[4] 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).

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

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

[7] Dolev, D., Dwork, C. and Naor, M.: Non-Malleable Cryptography, Proc. of STOC, pp.542–552 (1991).

[8] Diffie, W. and Hellman, M.: New Directions in Cryptography, IEEE Trans. on Information Theory, IT-22, 6, pp.644–654 (1976).

[9] ElGamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Trans. on Information Theory, IT-31, 4, pp.469–472 (1985).

[10] Fujisaki, E. and Okamoto, T.: How to Enhance the Security of Public-Key Encryption at Minimum Cost, Proc. of PKC'99, LNCS 1560, Springer-Verlag, pp.53–68 (1999).

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

[12] Goldwasser, S. and Micali, S.: Probabilistic Encryption, JCSS, 28, 2, pp.270-299 (1984).

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

[14] Koblitz, N.: Elliptic Curve Cryptosystems, Math. Comp., 48, 177, pp.203–209 (1987).

[15] Merkle, R.C. and Hellman, M.E.: Hiding Information and Signatures in Trapdoor Knapsacks, IEEE Trans. on Inform. Theory, 24, pp.525-530 (1978).

[16] Miller, V.S.: Use of Elliptic Curves in Cryptography, Proc. of Crypto'85, LNCS 218, Springer-Verlag, pp.417-426 (1985).

[17] Okamoto, T. and Pointcheval, D.: BEST: A Generic Coversion to Achieve Chosen-Ciphertext Security, manuscript (2000).

[18] Okamoto, T. and Pointcheval, D.: The Gap Problems: A New Class of Problems for the Security of Cryptographic Schemes, manuscript (2000).

[19] Rivest, R., Shamir, A. and Adleman,L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, Vol.21, No.2, pp.120-126 (1978).

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

$$a_{m-1}t^{m-1} + \ldots + a_2t^2 + a_1t + a_0$$

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.