

PSEC: Provably Secure Elliptic Curve Encryption Scheme (Submission to P1363a)

Tatsuaki Okamoto Eiichiro Fujisaki Hikaru Morita

NTT Laboratories
1-1 Hikarino-oka, Yokosuka-shi, 239-0847 Japan
Email: {okamoto, fujisaki, morita }@isl.ntt.co.jp

March 1999

Abstract

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

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 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 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. If the underlying random function is replaced by a practical random like function (e.g., SHA and MD5), PSEC is almost as efficient as the elliptic curve ElGamal scheme, and is almost three times as efficient as the elliptic curve Cramer-Shoup scheme.

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

Contents

1	Background: Provable Security of Public-key Encryption and Our Conversion	3
2	Description of PSEC	3
2.1	Overview	3
2.2	PSEC-1	4
2.2.1	Key Generation: \mathcal{G}	4
2.2.2	Encryption: \mathcal{E}	4
2.2.3	Decryption: \mathcal{D}	5
2.3	PSEC-2	5
2.3.1	Key Generation: \mathcal{G}	5
2.3.2	Encryption: \mathcal{E}	6
2.3.3	Decryption: \mathcal{D}	6
3	Attributes and Advantages of PSEC	7
4	Security Assessment of PSEC	8
5	Limitations	9
6	Intellectual Property Statement	9

1 Background: Provable Security of Public-key Encryption and Our Conversion

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.

Very recently the authors, Fujisaki and Okamoto [10, 11] realized two 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. The other 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. The latter conversion can guarantee the total security of the public-key encryption system in combination with a symmetric-key encryption scheme.

2 Description of PSEC

2.1 Overview

This section describes the proposed public-key encryption scheme, PSEC, which 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 two versions of PSEC: PSEC-1 and PSEC-2. PSEC-1 is designed for key-distribution and PSEC-2 is 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.

2.2 PSEC-1

2.2.1 Key Generation: \mathcal{G}

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

[Input] Security parameter $k(= pLen)$, which is a positive integer.

[Output] A pair of public-key, $(q, a, b, p, P, W, h, pLen, mLen, hLen, rLen, qLen)$, and secret-key s .

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 of order p . Here the size of p should be k (i.e., $|p| = k$).
- Choose $s \in (\mathbf{Z}/p\mathbf{Z})^*$ randomly, and calculates a point on E , where $W = sP$.
- Set $pLen := k$, and $qLen := |q|$. Set $mLen$ and $rLen$ such that $mLen + rLen \leq qLen - 1$. Set $hLen \leq pLen$.
- Select a (hash) function $h: \{0, 1\}^* \rightarrow \{0, 1\}^{hLen}$.

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.

2.2.2 Encryption: \mathcal{E}

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

[Input] Plaintext $m \in \{0, 1\}^{mLen}$ along with public-key $(q, a, b, p, P, W, h, pLen, mLen, hLen, rLen, qLen)$.

[Output] Ciphertext c .

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

- Select $r \in \{0, 1\}^{rLen}$ uniformly, and compute $t := h(m||r)$. Here $m||r$ denotes the concatenation of m and r .
- Compute Q and R , points on E , such that

$$Q := tW, \quad R := tP.$$

- Compute c :

$$c := (C_1, c_2) := (R, (m||r) \oplus x_Q),$$

where x_Q is the x -coordinate of Q .

2.2.3 Decryption: \mathcal{D}

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

[Input] Ciphertext c along with public-key $(q, a, b, p, P, W, h, pLen, mLen, hLen, rLen, qLen)$ and secret-key s .

[Output] Plaintext m or null string.

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

- Compute $D := sC_1$, a point on E , and $u := c_2 \oplus x_D$.
- Check whether the following equation holds or not:

$$C_1 = h(u)P.$$

- If it holds, output $[u]^{mLen}$ as decrypted plaintext, where $[u]^{mLen}$ denotes the most significant $mLen$ bits of u . Otherwise, output null string.

2.3 PSEC-2

2.3.1 Key Generation: \mathcal{G}

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

[Input] Security parameter $k(= pLen)$.

[Output] A pair of public-key, $(q, a, b, p, P, W, h, g, pLen, hLen, gLen, rLen, qLen)$, and secret-key, s .

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 of order p . Here the size of p should be k (i.e., $|p| = k$).
- Choose $s \in (\mathbf{Z}/p\mathbf{Z})^*$ randomly, and calculate a point on E , where $W = sP$. Set $pLen := k$. Set $rLen$ such that $rLen \leq qLen - 1$.
- Select (hash) functions $h: \{0, 1\}^* \rightarrow \{0, 1\}^{hLen}$, and $g: \{0, 1\}^* \rightarrow \{0, 1\}^{gLen}$.

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.

2.3.2 Encryption: \mathcal{E}

Let $SymE = (SymEnc, SymDec)$ be a pair of symmetric-key encryption and decryption algorithms with symmetric-key key , where the length of key is $gLen$. Encryption algorithm $SymEnc$ takes key and plaintext $ptext$, and returns ciphertext $SymEnc(key, ptext)$. Decryption algorithm $SymDec$ takes key and ciphertext $ctext$, and returns plaintext $SymDec(key, ctext)$.

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

[Input] Plaintext $m \in \{0, 1\}^{mLen}$ along with public-key $(q, a, b, p, P, W, h, g, pLen, hLen, gLen, rLen, qLen)$ and $SymEnc$.

[Output] Ciphertext c .

The operation of \mathcal{E} , on input $m, (q, a, b, p, P, W, h, g, pLen, hLen, gLen, rLen, qLen)$ is as follows:

- Select $r \in \{0, 1\}^{rLen}$ uniformly, and compute $g(r)$.
- Compute $t := h(m||r)$. Here $m||r$ denotes the concatenation of m and r .
- Compute Q and R , points on E , such that

$$Q := tW, \quad R := tP.$$

- Compute $c := (C_1, c_2, c_3)$:

$$C_1 := R,$$

$$c_2 := r \oplus x_Q,$$

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

where x_Q is the x -coordinate of Q .

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.

2.3.3 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 $(q, a, b, p, P, W, h, g, pLen, hLen, gLen, rLen, qLen)$ secret-key s and $SymDec$.

[Output] Plaintext m or null string.

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

- Compute $D := sC_1$, a point on E , and $r' := c_2 \oplus x_D$.

- Compute $m' := \text{SymDec}(G(r'), c_3)$.
- Check whether the following equation holds or not:

$$C_1 = h(m' || r')P.$$

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

3 Attributes and Advantages of PSEC

1. **[Security of PSEC-1]** If the elliptic curve decision Diffie-Hellman (EC-DDH) assumption (see the next section) is true, PSEC-1 is secure in the strongest sense under the random oracle model. Here, security in the strongest sense means to be semantically secure or non-malleable against adaptive chosen-ciphertext attacks (IND-CCA2 or NM-CCA2).
2. **[Security of PSEC-2 with one-time padding]** If the elliptic curve Diffie-Hellman (EC-DH) assumption is true, PSEC-2 with one-time padding (OTP) is secure in the strongest sense under the random oracle model if the parameters are appropriately selected.

Note that the elliptic curve version of the Cramer-Shoup (EC-CS) scheme is based on a stronger number theoretic assumption, the elliptic curve *decision* Diffie-Hellman (EC-DDH) assumption, than the elliptic curve Diffie-Hellman (EC-DH) assumption, while the EC-CS scheme is provably secure in the *standard model* (i.e., assuming a universal one-way hash function (UOWHF) not a random oracle).

Schemes	Security against CCA	Number-theoretical assumption	Random function assumption
PSEC-2(with OTP)	Secure (IND-CCA2)	EC-DH	Truly random
EC-Cramer-Shoup	Secure (IND-CCA2)	EC-DDH	UOWHF
OAEP	Secure (IND-CCA2)	RSA	Truly random

3. **[Security of PSEC-2 with symmetric-key encryption]** If the elliptic curve Diffie-Hellman (EC-DH) assumption is true and the underlying symmetric-key encryption is secure against passive attacks, PSEC-2 with the symmetric-key encryption is secure in the strongest sense under the random oracle model if the parameters are appropriately selected.

The advantage of this scheme is that security in the strongest sense is guaranteed for the total system that integrates the asymmetric and symmetric encryption schemes. Therefore, even if the underlying symmetric-key encryption is secure only against passive attacks and not against active attacks, PSEC-2, overall, guarantees security against active attacks.

An additional property of PSEC-2 is authentication without using MAC function. That is, the recipient can confirm whether the decrypted message is the same as the one the originator sent.

4. **[Efficiency]** Under the most practical environment of using public-key cryptosystems, where a public-key cryptosystem is used only for distributing a secret key (e.g., 128 bits long) of a secret-key cryptosystem (e.g., triple-DES and IDEA), a typical example of the parameters for PSEC-1 and PSEC-2 is as follows: for PSEC-1, $mLen = 128$, $rLen = 64$, and $pLen = hLen + 1 = 192$. For PSEC-2 with one-time padding, $rLen = 160$, $mLen = gLen = \text{secret-key size}$ (e.g., 128 or 256 etc.), $pLen = hLen + 1 = 160$. The encryption and decryption speeds of PSEC-1 and PSEC-2 are almost equivalent to those of the elliptic curve ElGamal scheme, and almost three times faster than those of the elliptic curve Cramer-Shoup scheme.

4 Security Assessment of PSEC

This section shows our results on the security of PSEC-1 and PSEC-2. They are easily obtained from the results presented in [10, 11].

Definition 4.1 Let \mathcal{G} be the key generator of PSEC-1, and (q, a, b, p, P) be a part of the public-key. Let s, t and u be uniformly selected in $\mathbf{Z}/p\mathbf{Z}$. $Q := sP$, $R := tP$, $V := stP$, and $W := uP$. Let $b \in \{0, 1\}$ be uniformly selected, and $X := V$ if $b = 0$, and $X := W$ if $b = 1$.

The elliptic curve decision Diffie-Hellman (EC-DDH) problem is intractable, if for any (uniform/ non-uniform) probabilistic polynomial time machine Adv , for any constant c , for sufficiently large $k(= pLen)$,

$$\Pr[Adv(q, a, b, p, P, Q, R, X) = b] < 1/2 + 1/k^c.$$

The probability is taken over the coin flips of \mathcal{G} and Adv .

The assumption that the elliptic curve Diffie-Hellman problem is intractable is called the elliptic curve Diffie-Hellman assumption.

Definition 4.2 Let \mathcal{G} be a key generator of PSEC-2, and (q, a, b, p, P) is a part of the public-key. Let s and t be uniformly selected in $\mathbf{Z}/p\mathbf{Z}$. $Q := sP$, $R := tP$, and $V := stP$.

The elliptic curve Diffie-Hellman (EC-DH) problem is intractable, if for any (uniform/non-uniform) probabilistic polynomial time machine Adv , for any constant c , for sufficiently large $k(= pLen)$,

$$\Pr[Adv(q, a, b, p, P, Q, R) = V] < 1/k^c.$$

The probability is taken over the coin flips of \mathcal{G} and Adv .

The assumption that the elliptic curve Diffie-Hellman problem is intractable is called the elliptic curve Diffie-Hellman assumption.

Definition 4.3 Let Adv be an adversary that runs in two stages. In the first stage, Adv endeavors to come up with a pair of equal-length messages, m_0 and m_1 , along with some state information s , where $|m_0| = |m_1| \leq (gLen)^a$ (a : constant). In the second stage, Adv is given a ciphertext $y := \text{SymEnc}(key, m_b)$, where $key \in \{0, 1\}^{gLen}$ and $b \in \{0, 1\}$ are randomly and uniformly chosen.

SymE is secure against passive attacks (IND-PAS), if for any (uniform/non-uniform) probabilistic polynomial time machine *Adv*, for any constant c , for sufficiently large $gLen$,

$$\Pr[Adv(gLen, m_0, m_1, s, y) = b] < 1/2 + 1/(gLen)^c.$$

The probability is taken over the coin flips of (key, b) and *Adv*.

Theorem 4.4 Let $rLen \geq c_0 pLen$ (c_0 : constant) and $hLen = pLen - 1$. PSEC-1 is semantically secure against adaptive chosen-ciphertext attacks (IND-CCA2) or non-malleable against adaptive chosen-ciphertext attacks (NM-CCA2) in the random oracle model, provided that the EC-DDH assumption is true.

Theorem 4.5 Let *SymE* for PSEC-2 be the one-time padding. Let $rLen = qLen - 1$, and $hLen = pLen - 1$. PSEC-2 is semantically secure against adaptive chosen-ciphertext attacks (IND-CCA2) or non-malleable against adaptive chosen-ciphertext attacks (NM-CCA2) in the random oracle model, provided that the EC-DH assumption is true.

Theorem 4.6 Let $rLen = qLen - 1$, and $hLen = pLen - 1$. PSEC-2 is semantically secure against adaptive chosen-ciphertext attacks (IND-CCA2) or non-malleable against adaptive chosen-ciphertext attacks (NM-CCA2) in the random oracle model, provided that the EC-DH assumption is true and that the underlying *SymE* is secure against passive attacks (IND-PAS).

Remark: We can also give the concrete efficiency analysis of the reduction for proving the security, and show that our reduction is efficient [10, 11].

5 Limitations

Recently Canetti et al. [6] have demonstrated that it is possible to devise cryptographic protocols which are provably secure in the random oracle model but for which no complexity assumption property instantiates the random-oracle-modeled hash function. However, the examples they used to make the random oracle model paradigm fail were very contrived, so the concerns induced by these examples do not appear to apply to any of the concrete practical schemes that have been proven secure in the random oracle model.

6 Intellectual Property Statement

NTT has filed patent applications on the techniques used in this contribution. NTT will license any resulting patent in a reasonable and non-discriminatory fashion. A letter to this effect will be provided.

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, to appear in Proc.of PKC'99, LNCS, Springer-Verlag.
- [11] Fujisaki, E. and Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes, manuscript (1998 November).
- [12] Goldwasser, S. and Micali, S.: Probabilistic Encryption, JCSS, 28, 2, pp.270-299 (1984).
- [13] Koblitz, N.: Elliptic Curve Cryptosystems, Math. Comp., 48, 177, pp.203–209 (1987).
- [14] Merkle, R.C. and Hellman, M.E.: Hiding Information and Signatures in Trapdoor Knapsacks, IEEE Trans. on Inform. Theory, 24, pp.525-530 (1978).
- [15] Miller, V.S.: Use of Elliptic Curves in Cryptography, Proc. of Crypto'85, LNCS 218, Springer-Verlag, pp.417-426 (1985).
- [16] 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).