EPOC: Efficient Probabilistic Public-Key Encryption (Submission to P1363a)

Tatsuaki Okamoto Shigenori Uchiyama Eiichiro Fujisaki

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

November 1998

Abstract

We describe a novel public-key cryptosystem, EPOC (Efficient Probabilistic Public-Key Encryption), which has two versions: EPOC-1 and EPOC-2. EPOC-1 is a public-key encryption system that uses a one-way trapdoor function and a random function (hash function). EPOC-2 is a public-key encryption system that uses a one-way trapdoor function, two random functions (hash functions) and a symmetric-key encryption (e.g., one-time padding and block-ciphers).

EPOC has several outstanding properties as follows:

1. EPOC-1 is semantically secure or non-malleable against chosen ciphertext attacks (IND-CCA2 or NM-CCA2) in the random oracle model under the p-subgroup assumption, which is comparable to the quadratic residue and higher degree residue assumptions.
2. EPOC-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 factoring assumption.
3. EPOC-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 factoring assumption, if the underlying symmetric encryption is secure against passive attacks.
4. The trapdoor technique with EPOC is fundamentally different from any other previous scheme including RSA-Rabin and Diffie-Hellman-ElGamal.
5. Under the most practical environment in which public-key cryptosystems would be used, the encryption and decryption speeds of EPOC are comparable (several times slower) to those of elliptic curve cryptosystems.

   Compared with OAEP (RSA) with small e (e.g.,$2^{16}+1$), although the encryption speed of EPOC is slower than that of OAEP, the decryption speed is faster than that of OAEP.

The encryption scheme described in this contribution is obtained by combining three results: one [25] on the trapdoor function technique is by Okamoto and Uchiyama, and the others [13, 14] on conversion techniques using random functions are by Fujisaki and Okamoto.

Contents

1 Background

1.1 Trapdoor One-way Functions and Our Novel Function

Diffie and Hellman proposed the concept of the public-key cryptosystem (or trapdoor one-way function) in 1976 [11]. Although extensive research has been made by numerous cryptographers and mathematicians to realize the concept of public-key cryptosystems for more than 20 years, very few concrete techniques that seem to be secure have been found.

A typical class of techniques is RSA-Rabin, which is the combination of the polynomial time algorithm of finding a root of a polynomial over a finite field and the intractability of the factoring problem. Another typical class of techniques is Diffie-Hellman-ElGamal, which is the combination of the commutative property of the logarithm in a finite Abelian group and the intractability of the discrete logarithm problem. The RSA-Rabin class includes RSA [29], Rabin [28], Williams [31, 32], LUC [30], Kurosawa-Itoh-Takeuchi [18], Cubic RSA [19] and the elliptic curve versions of RSA [17, 10]. The Diffie-Hellman-ElGamal class includes the Diffie-Hellman [11], ElGamal [12], and the elliptic/hyperelliptic curve versions of the Diffie-Hellman and ElGamal [23, 16, 6]. Several other techniques have been proposed such as the Goldwasser-Micali scheme [15] based on quadratic residuosity, the Ajtai-Dwark scheme [2] based on the lattice problem, the McEliece scheme [21] based on the error correcting code, knapsack type cryptosystems including the Merkle-Hellman, Chor-Rivest and Naccache-Stern schemes [22, 7, 24], and multivariate polynomial type cryptosystems including the Matsumoto-Imai and Patarin-Goubin schemes [20, 26, 27]. However they are not so efficient or not so secure[1]. Therefore, from the practical viewpoint, only two techniques, RSA-Rabin and Diffie-Hellman-ElGamal, have been used in many applications.

Among the RSA-Rabin and Diffie-Hellman-ElGamal techniques for realizing a trapdoor one-way function, no trapdoor function except the Rabin function and its variants such as its elliptic curve versions and Williams has been proven to be as secure as the primitive problems[2] (e.g., factoring and discrete logarithm problems).

Recently the authors, Okamoto and Uchiyama [25], proposed a novel one-way trapdoor function that is practical, provably secure, and has some other interesting properties as follows:

1. **New trick:** The trapdoor technique is fundamentally different from any other previous technique including RSA-Rabin and Diffie-Hellman-ElGamal.

2. **Probabilistic function:** It is a probabilistic trapdoor function. Let $E(m, r)$ be a ciphertext of plaintext m as randomized by r.

3. **One-wayness of the trapdoor function:** Inverting the function is proven to be as hard as factoring $n = p^2q$.

4. **Semantical security:** It is semantically secure if the following assumption, the p-subgroup assumption, is true: $E(0, r) = h^r \bmod n$ and $E(1, r') = gh^{r'} \bmod n$ are computationally in-

[1] The expression, "not so secure" includes the case where its security has not been sufficiently investigated.

[2] We say a trapdoor function is "provably secure" if inverting the function is proven to be as hard as solving the related primitive problem.

distinguishable, where r and r' are uniformly and independently selected from $\mathbf{Z}/n\mathbf{Z}$. This assumption is comparable to the quadratic residue and higher degree residue assumptions.

5. **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., 112 and 128 bits long) of a secret-key cryptosystem (e.g., triple-DES and IDEA), the encryption and decryption speeds of our trapdoor function are comparable (several times slower) to those of elliptic curve cryptosystems.

   Compared to the RSA function with small e (e.g., 3 or $2^{16}+1$), although the encryption speed of our function is slower than that of RSA, the decryption speed of our function is faster than that of RSA.

6. **Homomorphic property:** It has a homomorphic property:
   $E(m_0, r_0)E(m_1, r_1) \bmod n = E(m_0 + m_1, r_3)$, if $m_0 + m_1 < p$.

   Such a property is used for electronic voting and other cryptographic protocols.

   Note that no other encryption scheme except the higher-degree residue encryption [8] has such a homomorphic property, and the higher-degree residue encryption is extremely inefficient in decryption.

7. **Randomizability of ciphertext:** Even someone who does not know the secret key can change a ciphertext, $C = E(m, r)$, into another ciphertext, $C' = Ch^{r'} \bmod n$, while preserving plaintext m (i.e., $C' = E(m, r'')$), and the relationship between C and C' can be concealed (i.e., (C, C') and $(C, E(m', t))$ are indistinguishable).

   Such a property is useful for privacy protecting protocols.

1.2 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 of the schemes.

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 [13, 14] 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 EPOC

2.1 Overview

This section describes the proposed public-key encryption scheme, EPOC, 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 EPOC: EPOC-1 and EPOC-2. EPOC-1 is designed for key-distribution and EPOC-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 EPOC-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, $(n, g, h, H, pLen, mLen, hLen, rLen)$, and secret-key, (p, g_p).

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

- Choose two primes p, q ($|p| = |q| = k$), and compute $n := p^2 q$. Here, $p - 1 = p'u$ and $q - 1 = q'v$ such that p' and q' are primes, and $|u|$ and $|v|$ are $O(\log k)$.
- Choose $g \in (\mathbf{Z}/n\mathbf{Z})^*$ randomly such that the order of $g_p := g^{p-1} \bmod p^2$ is p. (Note that $\gcd(p, q-1) = 1$ and $\gcd(q, p-1) = 1$.)
- Choose h_0 from $(\mathbf{Z}/n\mathbf{Z})^*$ randomly and independently from g. Compute $h := h_0^n \bmod n$.
- Set $pLen := k$. Set $mLen$ and $rLen$ such that $mLen + rLen \leq pLen - 1$.

- Select a (hash) function H: $\{0,1\}^* \longrightarrow \{0,1\}^{hLen}$.

Note: g_p is a supplementary parameter that improves the efficiency of decryption, since g_p can be calculated from p and g. h can be $g^n \bmod n$ when $hLen = (2 + c_0)k$ (c_0 is a constant > 0). 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 $(n, g, h, H, pLen, mLen, hLen, rLen)$.

[**Output**] Ciphertext C.

The operation of $\mathcal{E}$, on input M and $(n, g, h, H, mLen, rLen)$, is as follows:

- Select $R \in \{0,1\}^{rLen}$ uniformly, and compute $r := H(M||R)$. Here $M||R$ denotes the concatenation of M and R.
- Compute C:
$$C := g^{(M||R)} h^r \bmod n.$$

2.2.3 Decryption: $\mathcal{D}$

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

[**Input**] Ciphertext C along with public-key $(n, g, h, H, pLen, mLen, hLen)$ and secret-key (p, g_p).

[**Output**] Plaintext M or null string.

The operation of $\mathcal{D}$, on input C along with $(n, g, h, H, pLen, mLen, hLen)$ and (p, g_p), is as follows:

- Compute $C_p := C^{p-1} \bmod p^2$, and $X := \frac{L(C_p)}{L(g_p)} \bmod p$, where $L(x) := \frac{x-1}{p}$.
- Check whether the following equation holds or not:
$$C = g^X h^{H(X)} \bmod n.$$
- If it holds, output $[X]^{mLen}$ as decrypted plaintext, where $[X]^{mLen}$ denotes the most significant $mLen$ bits of X. Otherwise, output null string.

2.3 EPOC-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, $(n, g, h, H, G, pLen, hLen, gLen, rLen)$, and secret-key, (p, g_p).

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

- Choose two primes p, q ($|p| = |q| = k$), and compute $n = p^2q$. Here, $p - 1 = p'u$ and $q - 1 = q'v$ such that p' and q' are primes, and $|u|$ and $|v|$ are $O(\log k)$.
- Choose $g \in (\mathbf{Z}/n\mathbf{Z})^*$ randomly such that the order of $g_p := g^{p-1} \bmod p^2$ is p. (Note that $\gcd(p, q - 1) = 1$ and $\gcd(q, p - 1) = 1$.)
- Choose h_0 from $(\mathbf{Z}/n\mathbf{Z})^*$ randomly and independently from g. Compute $h := h_0^n \bmod n$.
- Set $pLen := k$. Set $rLen$ such that $rLen \leq pLen - 1$.
- Select (hash) functions H: $\{0,1\}^* \longrightarrow \{0,1\}^{hLen}$, and G: $\{0,1\}^* \longrightarrow \{0,1\}^{gLen}$.

Note: g_p is a supplementary parameter that improves the efficiency of decryption, since g_p can be calculated from p and g. h can be $g^n \bmod n$ when $hLen = (2 + c_0)k$ (c_o is a constant > 0). H and G 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 K, where the length of K is $gLen$. 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 $SymDec(K, Y)$.

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

[**Input**] Plaintext $M \in \{0,1\}^{mLen}$ along with public-key $(n, g, h, H, G, pLen, hLen, gLen, rLen)$ and $SymEnc$.

[**Output**] Ciphertext $C = (C_1, C_2)$.

The operation of $\mathcal{E}$, on input M, $(n, g, h, H, G, pLen, hLen, gLen, rLen)$ and $SymEnc$, is as follows:

- Select $R \in \{0,1\}^{rLen}$ uniformly, and compute $G(R)$.
- Compute $H(M||R)$. Here $M||R$ denotes the concatenation of M and R.
- $$C_1 := g^R h^{H(M||R)} \bmod n,$$
  $$C_2 := SymEnc(G(R), M).$$

Remark: A typical way to realize $SymE$ is one-time padding.
That is, $SymEnc(K, X) := K \oplus X$, and $SymDec(K, Y) := K \oplus Y$, 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 $C = (C_1, C_2)$ along with public-key $(n, g, h, H, G, pLen, hLen, gLen, rLen)$, secret-key (p, g_p) and $SymDec$.

[**Output**] Plaintext M or null string.

The operation of $\mathcal{D}$, on input $C = (C_1, C_2)$ along with $(n, g, h, H, G, pLen, hLen, gLen)$, (p, g_p) and $SymDec$, is as follows:

- Compute $C_p := {C_1}^{p-1} \bmod p^2$, and $R' := \frac{L(C_p)}{L(g_p)} \bmod p$, where $L(x) := \frac{x-1}{p}$.
- Compute $M' := SymDec(G(R'), C_2)$.
- Check whether the following equation holds or not:

$$C_1 = g^{R'} h^{H(M'||R')} \bmod n.$$

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

2.4 Remark

We can use any random-like one-way functions H and G for EPOC. (As mentioned in subsection 1.2, EPOC 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 $L_l = hLen - 80l$.

3 Attributes and Advantages of EPOC

1. **[Security of EPOC-1]** If the p-subgroup assumption (see the next section) is true, EPOC-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 EPOC-2 with one-time padding**] If the factoring assumption for $n = p^2q$ is true, EPOC-2 with one-time padding (OTP) is secure in the strongest sense under the random oracle model, when the parameters are appropriately selected.

   Other practical and provably secure (IND-CCA2) encryption schemes such as (RSA based) OAEP and Cramer-Shoup are based on stronger number theoretic assumptions, the RSA or decision Diffie-Hellman assumption, than the factoring assumption. Here note that the Cramer-Shoup scheme is provably secure in the standard model (i.e., assuming not a random oracle but a universal one-way hash function (UOWHF)).

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

3. [**Security of EPOC-2 with symmetric-key encryption**] If the factoring assumption for $n = p^2q$ is true and the underlying symmetric-key encryption is secure against passive attacks, EPOC-2 with the symmetric-key encryption is secure in the strongest sense under the random oracle model, when 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, EPOC-2, overall, guarantees security against active attacks.

   Additional property of EPOC-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., 112 and 128 bits long) of a secret-key cryptosystem (e.g., triple-DES and IDEA), a typical example of the parameters for EPOC-1 and EPOC-2 is as follows: for EPOC-1, $mLen = 128$, $rLen = 80$, and $hLen = 208$. For EPOC-2 with one-time padding, $rLen = 80$, $gLen = 128$, $hLen = 80$. The corresponding encryption and decryption speeds of EPOC-1 and EPOC-2 are comparable (several times slower) to those of elliptic curve cryptosystems.

   Compared to OAEP (RSA) with small e (e.g.,$2^{16} + 1$), although the encryption speeds of EPOC-1 and EPOC-2 are slower than that of OAEP, the decryption speeds are faster than that of OAEP.

4 Security Assessment of EPOC

This section shows our results on the security of EPOC-1 and EPOC-2. They are easily obtained from the results presented in [25, 13, 14]. (See Appendix for [13].)

Definition 4.1 *Let $\mathcal{G}$ be a key generator of EPOC-1, and $(n, g, h, pLen, hLen)$ is the public-key. Let $b \in \{0,1\}$ and $r\{0,1\}^{hLen}$ be randomly and uniformly chosen. $C := g^b h^r \bmod n$.*

The p-subgroup problem is intractable if for any (uniform/non-uniform) probabilistic polynomial time machine Adv, for any constant c, for sufficiently large $k(= pLen)$,

$$\Pr[Adv(k, hLen, n, g, h, C) = b] < 1/2 + 1/k^c.$$

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

The assumption that the p-subgroup problem is intractable is called the p-subgroup assumption.

Definition 4.2 *Let $\mathcal{G}_0$ be an instance generator such that $\mathcal{G}_0(k) \rightarrow n$, $n = p^2 q$, $|p| = |q| = k$, $(p, q : primes)$. Here, the distribution of n is the same as that of n with EPOC-2. The factoring problem is, given (n, k), to find (p, q).*

The factoring problem is intractable, if for any (uniform/non-uniform) probabilistic polynomial time machine A, for any constant c, for sufficiently large k,

$$\Pr[A(k, n) = (p, q)] < 1/k^c.$$

The probability is taken over the coin flips of $\mathcal{G}_0$ and A.

The assumption that the factoring problem is intractable is called the factoring 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, X_0 and X_1, along with some state information s, where $|X_0| = |X_1| \leq (gLen)^a$ (a: constant). In the second stage, Adv is given a ciphertext $Y := SymEnc(K, X_b)$, where key $K \in \{0,1\}^{gLen}$ 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,

$$\Pr[Adv(gLen, X_0, X_1, s, Y) = b] < 1/2 + 1/(gLen)^c.$$

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

Theorem 4.4 *EPOC-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 p-subgroup assumption is true.*

Theorem 4.5 *Let SymE for EPOC-2 be one-time padding. Let $rLen = pLen - 1$, and $hLen = (2 + c_0)pLen$ ($c_0 > 0$: constant). EPOC-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 factoring assumption for $n = p^2 q$ is true.*

Theorem 4.6 *Let $rLen = pLen - 1$, and $hLen = (2 + c_0)pLen$ ($c_0 > 0$: constant). EPOC-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 factoring assumption for $n = p^2 q$ is true and that the underlying SymE is secure against passive attacks (IND-PAS).*

Remark: We can also give the concrete analysis of the reduction cost for proving the security, and show that our reduction is tight [13, 14]: for example, the ability to break IND-CCA2 security of EPOC-2 (with one-time padding) with a certain amount of computational resources implies the ability to factor n with almost the same computational resources.

5 Limitations

As for the limitations on the formal security proof in the random oracle model, our comments are the same as those by [1].

6 Intellectual Property Statement

NTT has filed patent applications (Japan, USA, UK, France and Germany) 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] Chao, J., Matsuda, N. and Tsujii, S.: Efficient construction of secure hyperelliptic discrete logarithm problems, Proc. of ICICS'97, LNCS 1334, Springer-Verlag, pp.292-301 (1997).

[7] Chor, B. and Rivest, R.L.: A knapsack type public key cryptosystem based on arithmetic in finite fields, Proc. of Crypto'84, LNCS 196, Springer-Verlag, pp.54-65 (1985).

[8] Cohen, J. and Fischer.: A Robust and Verifiable Cryptographically Secure Election Scheme, FOCS, pp.372-382 (1985).

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

[10] Demytko, N.: A New Elliptic Curve Based Analogue of RSA, Proc. of Eurocrypt'93, LNCS 765, Springer-Verlag, pp.40-49 (1994).

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

[12] 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).

[13] 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.

[14] Fujisaki, E. and Okamoto, T.: Provably Secure Integration of Asymmetric and Symmetric Encryption Schemes, manuscript (1998 November).

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

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

[17] Koyama, K. , Maurer, U. M. , Okamoto, T. and Vanstone, S. A.,: New Public-key Schemes based on Elliptic Curves over the Ring Zn, Proc. of Crypto'91, LNCS 576, Springer-Verlag, pp.252-266 (1992).

[18] Kurosawa, K., Ito, T. and Takeuchi, M.: Public Key Cryptosystem using a Reciprocal Number with the same Intractability as Factoring a Large Number, Cryptologia, 12, 4, pp.225-233 (1988).

[19] Loxton, J.H., Khoo, D.S.P., Bird, G.J. and Seberry, J.: A Cubic RSA Code Equivalent to Factorization, Journal of Cryptology, 5, 2, pp.139-150 (1992).

[20] Matsumoto, T. and Imai, H.: Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption, Proc. of Eurocrypt'88, LNCS 330, Springer-Verlag, pp.419-453 (1988).

[21] McEliece, R.J.: A Public-Key Cryptosystem Based on Algebraic Coding Theory, DSN progress report 42-44, Jet Propulsion Laboratories, Pasadena (1978).

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

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

[24] Naccache, D. and Stern, J.: A New Public-Key Cryptosystem, Proc. of Eurocrypt'97, LNCS 1233, Springer-Verlag, pp.27-436 (1997).

[25] Okamoto, T. and Uchiyama, S.: A New Public-Key Cryptosystem as Secure as Factoring, Proc. of Eurocrypt'98, LNCS 1403, Springer-Verlag, pp. 308–318(1998).

[26] Patarin, J. and Goubin, L.: Trapdoor one-way permutations and multivariate polynomials, Proc. of ICICS'97, LNCS 1334, Springer-Verlag, pp.356-368 (1997).

[27] Patarin, J. and Goubin, L.: Asymmetric cryptography with S-Boxes, Proc. of ICICS'97, LNCS 1334, Springer-Verlag, pp.369-380 (1997).

[28] Rabin, M.O.: Digital Signatures and Public-Key Encryptions as Intractable as Factorization, MIT, Technical Report, MIT/LCS/TR-212 (1979).

[29] 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).

[30] Smith, P. and Lennon, M.: LUC: A New Public Key System, Proc. of IFIP/SEC'93, pp. 103-117, North-Holland (1993).

[31] Williams, H.C.: A Modification of the RSA Public Key Encryption Procedure, IEEE Trans. on Inform. Theory, IT-26, 6, pp.726-729 (1980).

[32] Williams, H.C.: Some Public-Key Crypto-Functions as Intractable as Factorization, Proc. of Crypto'84, LNCS 196, Springer-Verlag, pp.66-70 (1985).

Appendix

How to Enhance the Security of Public-Key Encryption at Minimum Cost[3]

Eiichiro Fujisaki Tatsuaki Okamoto

Abstract

This paper presents a simple and efficient conversion from a semantically secure public-key encryption scheme against *passive adversaries* to a non-malleable (or semantically secure) public-key encryption scheme against *adaptive chosen-ciphertext attacks* (*active adversaries*) in the random oracle model. Since our conversion requires only one random (hash) function operation, the converted scheme is almost as efficient as the original one, when the random function is replaced by a practical hash function such as SHA-1 and MD5. We also give a concrete analysis of the reduction for proving its security, and show that our security reduction is (almost) optimally efficient. Finally this paper gives some practical examples of applying this conversion to some practical and semantically secure encryption schemes such as the ElGamal, Blum-Goldwasser and Okamoto-Uchiyama schemes [4, 7, 9].

Key words: semantical security, non-malleability, chosen-plaintext attack, adaptive chosen-ciphertext attack, random oracle model.

1 Introduction

1.1 Background

One of the most important topics in cryptography is to propose a practical and provably secure public-key encryption scheme. The strongest security notion in the public-key encryption is that of non-malleability or semantical security against adaptive chosen-ciphertext attacks. In [3], Bellare, Desai, Pointcheval and Rogaway 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 from 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* [2].

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 of the schemes.

Two typical primitives of the trap-door one-way function are RSA and ElGamal. The RSA function is a trap-door one-way permutation, and the ElGamal function is a probabilistic trap-door one-way function.

[3]The revised version of this appendix will be presented at PKC'99 in March 1–3, 1999, Kamakura, Japan, and appear in the proceedings of PKC'98, LNCS, Springer-Verlag.

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 called OAEP.)

However, their method cannot be applied to a probabilistic trap-door one-way function such as ElGamal. Therefore, a new measure to convert a probabilistic trap-door one-way function to an IND-CCA2 secure scheme (in the random oracle model) should be very valuable.

This paper will present such a generic and efficient measure. It converts a probabilistic trap-door one-way function to an IND-CCA2 secure scheme in the random oracle model provided that the trap-door one-way function is semantically secure (IND-CPA).

Since our conversion requires only one random (hash) function operation, the converted scheme is almost as efficient as the original scheme, when the random function is replaced by a practical hash function such as SHA-1 and MD5. Therefore, we can construct practical IND-CCA2 secure schemes (in the random oracle model) based on several practical IND-CPA secure schemes (under some reasonable assumptions) such as the (elliptic curve) ElGamal, Blum-Goldwasser and Okamoto-Uchiyama schemes [4, 7, 9, 11].

We begin by examining the notions of public-key encryption security.

1.2 Classification of Encryption Scheme Security

We can define the security levels of public-key encryption schemes, using the pairs of *goals* and *adversary models* (We saw this classification first in the paper of [3], which stated that the viewpoint was suggested to the authors by Naor).

The goals are one-wayness (OW), indistinguishability (IND) [8],and non-malleability (NM) [6] of encryption. One-wayness (OW) is defined by the adversary's inability, given a challenge ciphertext y, to decrypt y and get the whole plaintext x. Indistinguishability (IND) is defined by the adversary's inability, given a challenge ciphertext y, to learn any information about the plaintext x. Non-malleability (NM) is defined by the adversary's inability, given a challenge ciphertext y, to get a different ciphertext y' such that the corresponding plaintexts, x and x', are *meaningfully* related. Here a *meaningful* relation is, for instance, $x = \bar{x'}$.

The three adversary models are called chosen plaintext attack model(CPA), non-adaptive chosen-ciphertext attack model(CCA1), and adaptive chosen ciphertext attack model(CCA2). In CPA, the adversary is given only the public key. Of course, she can get the ciphertext of any plaintext chosen by her. Clearly, in public-key encryption schemes, this attack cannot be avoided. In CCA1, in addition to the public key, the adversary can access to the decryption oracle although she is only allowed to access to the oracle before given a challenge ciphertext. In CCA2, the adversary can access to the decryption oracle anytime (before or after given a challenge ciphertext). She is only prohibited from asking for the decryption of the challenge ciphertext itself.

Furthermore, we separate public-key encryption schemes into the random oracle (RO) model or the standard model. In the random oracle model, every adversary, independent of the adversary models, can be allowed to access to the random oracle anytime,

We say, for the security of public-key encryption scheme Π, that Π is secure in the sense of GOAL-ATK in the RO (or standard) model, where GOAL = {OW, IND, NM} and ATK = {CPA, CCA1,CCA2}. Here one can think of pairs of *goals* and *attacks*; OW-CPA, ..., OW-CCA2, IND-CPA, ..., NM-CCA2. According to [3], the relations among each notion of security are as follows: [4]

[4]Although one-wayness is not described in [3], the relations among OW and other goals in the diagram are

$$
\begin{array}{ccccc}
\text{NM-CPA} & \longleftarrow & \text{NM-CCA1} & \longleftarrow & \text{NM-CCA2} \\
 & & & \not\longrightarrow & \\
\downarrow & \searrow\!\!\!\!\nearrow & \downarrow & & \downarrow\!\!\uparrow \\
\text{IND-CPA} & \longleftarrow & \text{IND-CCA1} & \longleftarrow & \text{IND-CCA2} \\
\downarrow & & \downarrow & & \downarrow \\
\text{OW-CPA} & \longleftarrow & \text{OW-CCA1} & \longleftarrow & \text{OW-CCA2}
\end{array}
$$

Here, for $\mathbb{A}, \mathbb{B} \in$ GOAL-ATK "$\mathbb{A} \rightarrow \mathbb{B}$" (say, $\mathbb{A}$ implies $\mathbb{B}$) denotes that encryption scheme $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ being secure in the sense of $\mathbb{A}$ is also secure in the sense of $\mathbb{B}$, while "$\mathbb{A} \not\rightarrow \mathbb{B}$" (say, $\mathbb{A}$ doesn't imply $\mathbb{B}$) denotes Π being secure in the sense of $\mathbb{A}$ is not always secure in the sense of $\mathbb{B}$.

We will provide precise definitions of these notations in Sec.2 (Due to the space limitation, one-wayness is not discussed).

1.3 Our Results

This paper shows a simple and efficient conversion from an IND-CPA secure public-key encryption scheme to an NM-CCA2 (or IND-CCA2) secure public-key encryption scheme in the random oracle model.

Suppose $\mathcal{E}_{pk}(X, R)$ is an IND-CPA secure public-key encryption function, where pk is a public-key, X is a message with $k + k_0$ bits and R is a random string with l bits. The conversion is

$$\mathcal{E}'_{pk}(x, r) := \mathcal{E}_{pk}(x||r, H(x||r)), \tag{1}$$

where H is a random function of $\{0,1\}^{k+k_0} \longrightarrow \{0,1\}^l$, x is a message of the converted public-key encryption scheme $\Pi' := (\mathcal{K}', \mathcal{E}', \mathcal{D}')$, r is a random string with k_0 bits, and $||$ denotes concatenation.

Main Theorem (Theorem 4.3)
If there exists a (t, q_H, q_D, ϵ)-breaker A for $\Pi'(1^k)$ (the converted scheme, $\Pi' := (\mathcal{K}', \mathcal{E}', \mathcal{D}')$) in the sense of IND-CCA2 in the random oracle model, then there exist constants, c_0, c_1, and a $(t', 0, 0, \epsilon')$-breaker A' for $\Pi(1^{k+k_0})$ (the original scheme, $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$) in the sense of IND-CPA where

$$
\begin{aligned}
t' &= t + c_1 \cdot q_H \cdot k + q_D \cdot q_H \cdot (T_{\mathcal{E}}(k) + c_0 \cdot k), \text{ and} \\
\epsilon' &= (\epsilon - \frac{q_H}{2^{k_0 - 1}}) \cdot (1 - \frac{1}{2^l})^{q_D}.
\end{aligned}
$$

Here, (t, q_H, q_D, ϵ)-breaker A (informally) means that A stops within t steps, succeeds with probability $\geq \epsilon$, makes at most q_H queries to random oracle H, and makes at most $q_{\mathcal{D}}$ queries to decryption oracle $\mathcal{D}_{sk}$ (see Sec. 2 for the formal definition). $T_{\mathcal{E}}(k)$ denotes the computational time of the encryption algorithm $\mathcal{E}_{pk}(\cdot)$, and c_0 and c_1 depend on details of the underlying model of computation.

This theorem implies that if the original scheme Π is IND-CPA secure, the converted scheme Π' is IND-CCA2 secure (and NM-CCA2 secure as well) in the random oracle model, provided that k, k_0 and l are in proportion to system size.

---

clear.

1.4 Merits and Related Works

As mentioned above, Bellare-Rogaway conversion [2] can be applied to a trap-door one-way permutation (such as RSA) and our conversion can be applied to a probabilistic trap-door one-way function (such as ElGamal).

Since our conversion starts from a more secure scheme, an IND-CPA secure scheme, than that of Bellare-Rogaway conversion, our conversion is simpler and more efficient than theirs, i.e., our conversion requires only one random function operation, while Bellare-Rogaway conversion requires two random function operations.

In addition, the security reduction in our conversion is more efficient (tight) than that of Bellare-Rogaway's, since we need no additional reduction for semantical security.

Recently, Cramer and Shoup presented a new public-key encryption scheme which is IND-CCA2 secure in the standard model [5]. Although their scheme is still practical, our approach has some advantages over their scheme as follows:

Our converted version of the ElGamal scheme (the enhanced ElGamal scheme) is at least twice as efficient as the Cramer-Shoup scheme. Here, the enhanced ElGamal is IND-CCA2 secure in the random oracle model and under the decision Diffie-Hellman assumption, while the Cramer-Shoup is IND-CCA2 secure under the universal one-way hash assumption and the decision Diffie-Hellman assumption.

In [11], Tsiounis and Yung proposed a converted ElGamal scheme which is secure in the NM-CCA2 sense in the random oracle model as well. However our conversion is much more efficient than theirs under the same assumption (the decision Diffie-Hellman assumption).

2 Definitions and Security Models

In this section, we give some definitions about encryption scheme security. Basically, we follow the terminology in [2, 3].

Definition 2.1 *Let A be a probabilistic algorithm and let $A(x_1, \ldots, x_n; r)$ be the result of A on input $(x_1, \ldots, x_n)$ and coins r. We define by $y \leftarrow A(x_1, \ldots, x_n)$ the experiment of picking r at random and letting y be $A(x_1, \ldots, x_n; r)$. If S is a finite set, let $y \leftarrow_R S$ be the operation of picking y at random and uniformly from finite set S. ε denote the null symbol and, for list τ, $\tau \leftarrow \varepsilon$ denote the operation of letting list τ be empty. Moreover, let $||$ denote the concatenation operator and, for n-bit string x, $[x]^k$ and $[x]_k$ denote the first and last k-bit strings of x respectively ($k \leq n$).*

Definition 2.2 **[Random Oracle Model]** *We define by Ω the set of all maps from the set $\{0,1\}^*$ of finite strings to the set $\{0,1\}^\infty$ of infinite strings. $H \leftarrow \Omega$ means that we chose map H from a set of an appropriate finite length (say : $\{0,1\}^a$) to a set of an appropriate finite length (say $\{0,1\}^b$), from Ω at random and uniformly, restricting the domain to $\{0,1\}^a$ and the range to the first b bits of output.*

Definition 2.3 **[Public-Key Encryption]** *We say that a triple of algorithm $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is a public-key encryption scheme if*

- *$\mathcal{K}$, the key-generation algorithm, is a probabilistic algorithm which on input 1^k ($k \in \mathbb{N}$) outputs, in polynomial-time in k, a pair (pk, sk) of matching public and secret keys.*
- *$\mathcal{E}$, the encryption algorithm, is a probabilistic algorithm which on input public-key pk and message $x \in \{0,1\}^*$ outputs ciphertext y in polynomial-time in k. We denote by $\mathcal{E}_{pk} : \{0,1\}^k \times \{0,1\}^{l(k)} \to \{0,1\}^{n(k)}$ the map from the product of k-bit message and*

$l(k)$-bit coin-flipping spaces to $n(k)$-bit cipher space, where functions, $l(\cdot)$ and $n(\cdot)$, are bounded in some polynomial, namely $l(k), n(k) < \exists p(k)$ for enough large k.

- $\mathcal{D}$, the decryption algorithm, is a probabilistic algorithm which on input secret-key sk and ciphertext y outputs $\mathcal{D}_{sk}(y)$ such that

$$\mathcal{D}_{sk}(y) := \begin{cases} x \in \{0,1\}^* & \text{if there exists } x \text{ such that } y = \mathcal{E}_{pk}(x) \\ \varepsilon \text{ (null)} & \text{otherwise.} \end{cases}$$

We say that ciphertext y is valid if if there exists a message x such that $y = \mathcal{E}_{pk}(x)$.

When $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is defined in the random oracle model and we insist on the fact, we will denote $\Pi := (\mathcal{K}, \mathcal{E}^H, \mathcal{D}^H)$.

Below, we give the precise definitions of GOAL-ATK described in Sec.1.2. Due to the space limitations, one-wayness is not described.

Definition 2.4 **[IND-ATK]** *Let $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme and let $A := (A_1, A_2)$ be a pair of probabilistic algorithms (say Adversary). For atk $\in$ {cpa,cca1,cca2} and $k \in \mathbb{N}$, let define $Adv^{\text{ind-atk}}_{A,\Pi}(k) :=$*

$$2\Pr[H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); (x_0, x_1, s) \leftarrow A_1^{O_1,H}(pk); b \leftarrow_R \{0,1\}; \\ y \leftarrow \mathcal{E}_{pk}(x_b) : A_2^{O_2,H}(x_0, x_1, s, y) = b] - 1.$$

Here, $O_1(\cdot)$, $O_2(\cdot)$ are defined as follows:

- *If atk=cpa then $O_1(\cdot) = \varepsilon$ and $O_2(\cdot) = \varepsilon$*
- *If atk=cca1 then $O_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ and $O_2(\cdot) = \varepsilon$*
- *If atk=cca2 then $O_1(\cdot) = \mathcal{D}_{sk}(\cdot)$ and $O_2(\cdot) = \mathcal{D}_{sk}(\cdot)$*

In addition we define that A_1 outputs x_0, x_1 with $|x_0| = |x_1|$ and, in the case of IND-CCA2, A_2 does not ask its oracle to decrypt y.

We say that Π is secure in the sense of IND-ATK if for any adversary A being polynomial-time in k $Adv^{\text{ind-atk}}_{A,\Pi}(k)$ is negligible in k.

We insist that $A := (A_1, A_2)$ is not allowed to access to H in the standard model. When we insist on that, we write $A_1^{O_1}$ and $A_2^{O_2}$ instead of $A_1^{O_1,H}$ and $A_2^{O_2,H}$, respectively. On the other hand, when we insist on the random oracle model, we write $\mathcal{E}^H_{pk}(\cdot)$ and $\mathcal{D}^H_{sk}(\cdot)$ instead of $\mathcal{E}_{pk}(\cdot)$ and $\mathcal{D}_{sk}(\cdot)$, respectively.

Definition 2.5 **[NM-ATK]** *Let $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme and let $A := (A_1, A_2)$ be a pair of probabilistic algorithms (say Adversary). For atk $\in$ {cpa,cca1,cca2} and $k \in \mathbb{N}$, let define*

$$Adv^{\text{nm-atk}}_{A,\Pi}(k) := |Succ^{\text{nm-atk}}_{A,\Pi}(k) - Succ^{\text{nm-atk}}_{A,\Pi,\$}(k)|$$

where $Succ^{\text{nm-atk}}_{A,\Pi}(k) :=$

$$\Pr[H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); (M, s) \leftarrow A_1^{O_1,H}(pk); x, x' \leftarrow M; y \leftarrow \mathcal{E}_{pk}(x); \\ (R, \vec{y}) \leftarrow A_2^{O_2,H}(M, s, y); \vec{x} \leftarrow \mathcal{D}_{sk}(\vec{y}) : (y \notin \vec{y}) \wedge (\varepsilon(\text{null}) \notin \vec{x}) \wedge R(x, \vec{x})]$$

and $Succ^{\text{nm-atk}}_{A,\Pi,\$}(k) :=$

$$\Pr[H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); (M, s) \leftarrow A_1^{O_1,H}(pk); x, x' \leftarrow M; y \leftarrow \mathcal{E}_{pk}(x); \\ (R, \vec{y}) \leftarrow A_2^{O_2,H}(M, s, y); \vec{x} \leftarrow \mathcal{D}_{sk}(\vec{y}) : (y \notin \vec{y}) \wedge (\varepsilon(\text{null}) \notin \vec{x}) \wedge R(x', \vec{x})]$$

Here, $O_1(\cdot)$, $O_2(\cdot)$ are defined as before. In the case of IND-CCA2, A_2 does not ask its oracle to decrypt y.

We say that M is valid if $|x| = |x'|$ for any x, x' that are given non-zero probability in the message space M.

We say that Π is secure in the sense of NM-ATK if any adversary A being polynomial-time in k outputs a valid message space M samplable in polynomial in k and a relation R computable in polynomial in k, then $Adv^{\text{nm-atk}}_{A,\Pi}(k)$ is negligible in k.

We insist that $A := (A_1, A_2)$ is not allowed to access to H in the standard model. When we insist on that, we write $A_1^{O_1}$ and $A_2^{O_2}$ instead of $A_1^{O_1,H}$ and $A_2^{O_2,H}$, respectively. On the other hand, when we insist on the random oracle model, we write $\mathcal{E}^H_{pk}(\cdot)$ and $\mathcal{D}^H_{sk}(\cdot)$ instead of $\mathcal{E}_{pk}(\cdot)$ and $\mathcal{D}_{sk}(\cdot)$, respectively.

We review some important results proven in [3] below. Here, as mentioned above, for $\mathbb{A}, \mathbb{B} \in$ GOAL-ATK "$\mathbb{A} \to \mathbb{B}$" (say, $\mathbb{A}$ implies $\mathbb{B}$) denotes that encryption scheme $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ being secure in the sense of $\mathbb{A}$ is also secure in the sense of $\mathbb{B}$, while "$\mathbb{A} \not\to \mathbb{B}$" (say, $\mathbb{A}$ doesn't imply $\mathbb{B}$) denotes Π being secure in the sense of $\mathbb{A}$ is not always secure in the sense of $\mathbb{B}$.

Proposition 2.6 *IND-CCA2 $\to$ NM-CCA2.*

From this proposition, it is clear that

Corollary 2.7 *IND-CCA2 $\longleftrightarrow$ NM-CCA2.*

Proposition 2.8 *IND-CCA1 $\not\to$ NM-CCA2.*

The following definition is utilized to discuss security more exactly (exact security).

Definition 2.9 [**Breaking Algorithm**] *Let $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme. We say that an adversary A is a $(t, q_H, q_{\mathcal{D}}, \epsilon)$-breaker for $\Pi(1^k)$ in GOAL-ATK if $Adv^{\text{goal-atk}}_{A,\Pi}(k) \geq \epsilon$ and, moreover, A runs within at most running time t, asking at most q_H queries to $H(\cdot)$ and at most $q_{\mathcal{D}}$ queries to $\mathcal{D}_{sk}(\cdot)$. In addition, q_H denotes the number of queries A asks to random function $H(\cdot)$, and similarly, $q_{\mathcal{D}}$ denotes the number of queries A asks to decryption oracle $\mathcal{D}_{sk}(\cdot)$. In the case of atk = cpa, then $q_{\mathcal{D}} = 0$. In the case of the standard model, then $q_H = 0$.*

In the following, we will recall the notion of Plaintext Awareness and the main results.

Definition 2.10 [**Plaintext Awareness (PA)**] *Let $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a public-key encryption scheme, let B be an adversary, and let K be an polynomial-time algorithm (say knowledge extractor) . For any $k \in \mathbb{N}$ let $Succ^{pa}_{K,B,\Pi}(k) :=$*

$$\Pr[H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^k); (\tau, \eta, y) \leftarrow runB^{H, \mathcal{E}_{pk}}(pk) : K(\tau, \eta, y, pk) = \mathcal{D}_{sk}(y)].$$

where $\tau := \{(h_1, H_1), \ldots, (h_{q_H}, H_{q_H})\}$, $\eta := \{y_1, \ldots, y_{q_E}\}$, and $y \notin \eta$. We describe a supplementary explanation: By $(\tau, \eta, y) \leftarrow runB^{H, \mathcal{E}_{pk}}(pk)$ we mean the following. Run B on input pk and oracles $H(\cdot)$ and $\mathcal{E}_{pk}(\cdot)$ and record (τ, η, y) from B's interaction with its oracles. τ denotes the set of all B's queries and the corresponding answers of $H(\cdot)$. η denotes the set of all the answers (ciphertexts) received as the result of $\mathcal{E}_{pk}$. Here we insist that η doesn't include the corresponding queries (plaintexts) from B. y denotes the output of B.

We say that K is a $(t, \lambda(k))$-knowledge extractor if $Succ^{pa}_{K,B,\Pi}(k) \geq \lambda(k)$ and K runs within at most running time t (or t steps).

We say that Π *is secure in the sense of PA if* Π *is secure in the sense of IND-CPA and there exists a* $(t, \lambda(k))$*-knowledge extractor* K *where* t *is polynomial in* k *and* $(1 - \lambda(k))$ *is negligible in* k*.*

The following results proven in [3] is important.

Proposition 2.11 *PA* $\rightarrow$ *IND-CCA2 in the random oracle model.*

Corollary 2.12 *PA* $\rightarrow$ *NM-CCA2 in the random oracle model.*

3 Basic Scheme

Suppose a public-key encryption scheme, $\Pi := (\mathcal{K}, \mathcal{E}, \mathcal{D})$, exists which is semantically secure against every chosen-plaintext (passive) attack. We denote by $\mathcal{E} := \mathcal{E}_{pk}(x, r)$ the encryption function of the encryption scheme. Here x denotes a message, r denotes a random value, and $\mathcal{E}_{pk} : \{0,1\}^k \times \{0,1\}^{l(k)} \rightarrow \{0,1\}^{n(k)}$. Let $H : \{0,1\}^{k+k_0} \rightarrow \{0,1\}^l$ be an ideal hash function, where $l := l(k + k_0)$.

We introduce a new public-key encryption scheme, $\Pi' := (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ which is derived from Π and hash function H as follows:

Basic Scheme $\Pi' := (\mathcal{K}', \mathcal{E}', \mathcal{D}')$

- $\mathcal{K}'(1^k) := \mathcal{K}(1^{k+k_0})$ where $k_0 := k_0(k)$ is associated to k, for instance $k_0 \leq k$.
- $\mathcal{E}'_{pk} : \{0,1\}^k \times \{0,1\}^{k_0} \rightarrow \{0,1\}^n$ is defined by

$$\mathcal{E}'_{pk}(x, r) := \mathcal{E}_{pk}(x||r, H(x||r)),$$

  where $|x| = k$, $|r| = k_0$, and $n := n(k + k_0)$.
- $\mathcal{D}'_{sk}(y) : \{0,1\}^n \rightarrow \{0,1\}^k$ is defined by

$$\mathcal{D}'_{sk}(y) := \begin{cases} [\mathcal{D}_{sk}(y)]^k & \text{if } y = \mathcal{E}_{pk}(\mathcal{D}_{sk}(y), H(\mathcal{D}_{sk}(y))) \\ \varepsilon \text{ (null)} & \text{otherwise} \end{cases}$$

  where $[\mathcal{D}_{sk}(y)]^k$ denotes the first k-bit of $\mathcal{D}_{sk}(y)$.

Hereafter we will show that Π' is semantically secure against every adaptive chosen-ciphertext attack, namely, non-malleable against every adaptive chosen-ciphertext attack as well.

4 Security

Theorem 4.1 [Knowledge extractor K **of** Π'**]** *If there exists a* (t, q_H)*-adversary* B*, then there exist a constant* c_0 *and a* $(t', \lambda(k))$*-knowledge extractor* K *such that*

$$\begin{aligned} t' &= t + q_H(T_{\mathcal{E}}(k) + c_0 \cdot k) \text{ and} \\ \lambda(k) &= 1 - 2^{-l}. \end{aligned}$$

Here $T_{\mathcal{E}}(k)$ *denotes the computational running time of the encryption algorithm* $\mathcal{E}_{pk}(\cdot)$*, and* c_0 *depends on details of the underlying model of computation.*

Proof:

The specification of knowledge extractor K is as follows:

Extractor: $K(\tau, \eta, y, pk)$
for q_H times do
if $y == \mathcal{E}_{pk}(h_i, H_i)$;
then $x \leftarrow [h_i]^k$ and break
else $x \leftarrow \varepsilon$ (null)
return x
End.

Now we define c_0 as corresponding to the computation time of comparing a bit to a bit plus some overhead. Then, from the specification, K runs within $t + q_H(T_{\mathcal{E}}(k) + c_0 \cdot k)$ time.

Next we think of the probability that K outputs the plaintext x correctly, namely $x = \mathcal{D}_{sk}(y)$. Recall that $\tau := \{(h_1, H_1), \ldots, (h_{q_H}, H_{q_H})\}$, $\eta := \{y_1, \ldots, y_{q_E}\}$, and $y \notin \eta$. Here let $Fail$ be an event (or a propositional variable) assigned to be true iff $x \neq \mathcal{D}'_{sk}(y)$ and let $AskH$ be an event assigned to be true iff there exists (h_i, H_i) in the list τ such that $y = \mathcal{E}_{pk}(h_i, H_i)$. Then it follows that

$$\begin{aligned} \Pr[Fail] &= \Pr[Fail|AskH] \cdot \Pr[AskH] + \Pr[Fail|\neg AskH] \cdot \Pr[\neg AskH] \\ &\leq \Pr[Fail|AskH] + \Pr[Fail|\neg AskH] \leq 0 + 2^{-l} = 2^{-l}. \end{aligned}$$

We explain that $\Pr[Fail|\neg AskH]$ is at most 2^{-l}. For *valid* y, there exists h such that $y = \mathcal{E}_{pk}(h, H(h))$. As $y \notin \eta$, it follows that $h \neq \mathcal{D}_{sk}(y_i)$ for every $y_i \in \eta$. Therefore, if B doesn't ask query h to oracle $H(\cdot)$, she can only guess a pair $(h, H(h))$ at most with probability 2^{-l}. This means that $\Pr[y \text{ is } valid|\neg AskH] \leq 2^{-l}$. On the other hand, if $\neg AskH$ is true, from the specification, the extractor K always outputs ε, namely y is *invalid*. This means that $\Pr[Fail|\neg AskH] = \Pr[\varepsilon \neq \mathcal{D}_{sk}(y)|\neg AskH] \leq 2^{-l}$.

Hence, $\lambda(k) = 1 - \Pr[Fail] = 1 - 2^{-l}$. ¶

Theorem 4.2 [Π'**: IND-CPA secure**] *If there exists a* $(t, q_H, 0, \epsilon)$*-breaker* $A := (A_1, A_2)$ *for* $\Pi'(1^k)$ *in the sense of IND-CPA in the RO model, then there exist a constant* c_1 *and a* $(t', 0, 0, \epsilon')$*-breaker* $A' := (A'_1, A'_2)$ *for* $\Pi(1^{k+k_0})$ *in the sense of IND-CPA (in the standard model) where*

$$t' = t + c_1 \cdot q_H \cdot k, \text{ and } \epsilon' = \epsilon - \frac{q_H}{2^{k_0 - 1}}.$$

Here c_1 *depends on details of the underlying model of computation of* A'*.*

Proof:

We run $A' := (A'_1, A'_2)$ in the IND-CPA and standard model setting, using $A := (A_1, A_2)$ as oracles respectively.

Basically, when A_i asks query h, A'_i works as follows: If h has not been entered in list τ, A'_i, choosing l-bit random string H, makes an entry of (h, H) in τ and answers A_i with H. If (h, H) is already in list τ, A'_i answers A_i with the corresponding H. The list τ is empty at first. When A_1 outputs (x_0, x_1, s), A'_1 outputs $(x_0||r_0, x_1||r_1, s)$ where r_0, r_1 are k_0-bit random strings generated by A'_1. Then, outside A', $y := \mathcal{E}_{pk}(X_b, R)$ is computed using a random bit $b \in \{0, 1\}$ and l-bit random string R, where $X_0 := (x_0||r_0)$ and $X_1 := (x_1||r_1)$. y is inputted on A'_2 as well as (X_0, X_1, s).

If A_2 asks either X_0 or X_1 as a query, A'_2 makes A_2 stop and outputs the corresponding $b \in \{0, 1\}$ as an answer, otherwise A_2 follows the basic rule mentioned above. When A_2 asks neither of them, A'_2 outputs b that A_2 output as an answer.

The argument behind the proof is as follows: If A_2 asks a query to A'_2, which coincides with either $(x_0||r_0)$ or $(x_1||r_1)$, it is almost equivalent to $\mathcal{D}_{sk}(y)$, because (even unbounded

powerful) A_2 has no clue to k_0-bit random string $r_{\bar{b}}$, where $\bar{b}$ is the complement of bit b. Therefore, if A_2 asks either of them, the corresponding b is expected to be *valid*. On the other hand, if A_2 asks neither of them, A_2 is expected to output *valid* b because A_2 cannot distinguish y from a correct ciphertext for A_2.

The specification of adversary $A' := (A'_1, A'_2)$ is as follows:

Adversary: $A'_1(pk)$
$\tau \leftarrow \varepsilon$;
run $A_1(pk)$
do while A_1 does not ask query h to $H(\cdot)$
if $h \notin \tau_h$, where τ_h is the list of h's in τ
$H \leftarrow_R \{0,1\}^l$;
put (h, H) on the list τ;
answer A_1 with H;
else $h \in \tau_h$
answer A_1 with H such that $(h, H) \in \tau$
A_1 outputs (x_0, x_1, s)
$r_0, r_1 \leftarrow_R \{0,1\}^{k_0}$;
return $(x_0||r_0, x_1||r_1, s)$
End.

Adversary: $A'_2(x_0||r_0, x_1||r_1, s, y)$
run $A_2(x_0, x_1, s, y)$
do while A_1 does not ask query h to $H(\cdot)$
if $h == (x_b||r_b)$ for $b \in \{0,1\}$
stop A_2 and output b
else if $h \notin \tau_h$, where τ_h is the list of h's in τ
$H \leftarrow_R \{0,1\}^l$;
put (h, H) on the list τ;
answer A_1 with H;
else $h \in \tau_h$
answer A_1 with H such that $(h, H) \in \tau$
A_2 outputs b
return b
End.

Here, from Definition 2.4, b is chosen from $\{0,1\}$ with probability $1/2$, R is an l-bit random string, and $y = \mathcal{E}_{pk}(x_b||r_b, R)$.

We consider that c_1 corresponds to the computational time of comparing a bit to a bit, coin-flipping, plus some overhead. Then, from the specification of A', it runs within at most running time $(t + c_1 \cdot q_H \cdot k)$.

We now analyze the success probability of adversary $A' := (A'_1, A'_2)$. First we define the following events:

$$
\begin{aligned}
SuccA &:= [H \leftarrow \Omega; (pk, sk) \leftarrow \mathcal{K}(1^{k+k_0}); (x_0, x_1, s) \leftarrow A_1^H(pk); b \leftarrow_R \{0,1\}; \\
&\quad r_b, r_{\bar{b}} \leftarrow_R \{0,1\}^{k_0}; y \leftarrow \mathcal{E}_{pk}((x_b||r_b), H(x_b||r_b)) : A_2^H(x_0, x_1, s, y) = b], \text{ and} \\
SuccA' &:= [(pk, sk) \leftarrow \mathcal{K}(1^{k+k_0}); (X_0, X_1, s) \leftarrow A'_1(pk'); b \leftarrow_R \{0,1\}; \\
&\quad R_b, R_{\bar{b}} \leftarrow_R \{0,1\}^{k_0}; y \leftarrow \mathcal{E}_{pk}(X_b, R_b) : A'_2(X_0, X_1, s, y) = b],
\end{aligned}
$$

where $\bar{b}$ denotes the complement of b.

We can define the advantages of A and A', without loss of generality, as $Adv^{\text{ind-atk}}_{A,\Pi'}(k + k_0) := 2 \cdot \Pr[SuccA] - 1$, and $Adv^{\text{ind-atk}}_{A',\Pi}(k) := 2 \cdot \Pr[SuccA'] - 1$.

Next, let us define by $Ask0$ an event assigned to be true iff a query of A_2 coincides with $(x_b||r_b)$ and by $Ask1$ an event assigned to be true iff a query of A_2 coincides with $(x_{\bar{b}}||r_{\bar{b}})$. Then,

$$\begin{aligned}
\Pr[SuccA] &= \Pr[SuccA|Ask0] \cdot \Pr[Ask0] + \Pr[SuccA|(\neg Ask0) \wedge Ask1] \cdot \Pr[(\neg Ask0) \wedge Ask1] \\
&\quad + \Pr[SuccA|(\neg Ask0) \wedge (\neg Ask1)] \cdot \Pr[(\neg Ask0) \wedge (\neg Ask1)], \text{ and} \\
\Pr[SuccA'] &= \Pr[SuccA'|Ask0] \cdot \Pr[Ask0] + \Pr[SuccA'|(\neg Ask0) \wedge Ask1] \cdot \Pr[(\neg Ask0) \wedge Ask1] \\
&\quad + \Pr[SuccA'|(\neg Ask0) \wedge (\neg Ask1)] \cdot \Pr[(\neg Ask0) \wedge (\neg Ask1)].
\end{aligned}$$

From the specification of A' above, it is clear that $\Pr[SuccA'|Ask0] = 1$, $\Pr[SuccA'|(\neg Ask0) \wedge Ask1] = 0$ and $\Pr[SuccA|(\neg Ask0) \wedge (\neg Ask1)] = \Pr[SuccA'|(\neg Ask0) \wedge (\neg Ask1)]$. Hence, $\Pr[SuccA']$ is at most $\Pr[(\neg Ask0) \wedge Ask1])$ less than $\Pr[SuccA]$ because

$$\begin{aligned}
\Pr[SuccA'] - \Pr[SuccA] &= (1 - \Pr[SuccA|Ask0]) \cdot \Pr[Ask0] - \Pr[SuccA|(\neg Ask0) \wedge Ask1] \\
&\quad \cdot \Pr[(\neg Ask0) \wedge Ask1] \geq -\Pr[(\neg Ask0) \wedge Ask1].
\end{aligned}$$

Finally, we have

$$\Pr[SuccA'] \geq \frac{\epsilon + 1}{2} - \frac{q_H}{2^{k_0}},$$

since we infer that $\Pr[(\neg Ask0) \wedge Ask1] \leq \frac{q_H}{2^{k_0}}$,

Therefore, we have that $\epsilon' = \epsilon - \frac{q_H}{2^{k_0 - 1}}$. ¶

From Definition 2.10 and Theorems, 4.1 and 4.2, Π is secure in the sense of PA, and hence, by Proposition 2.11, secure in the sense of IND-CCA2. Thus, our interest in the following theorem is focused on the efficiency of the reduction.

Theorem 4.3 [Π'**: IND-CCA2 secure**] *If there exists a (t, q_H, q_D, ϵ)-breaker $A := (A_1, A_2)$ for $\Pi'(1^k)$ in the sense of IND-CCA2 in the RO model, then there exist constants, c_0, c_1, and a $(t', 0, 0, \epsilon')$-breaker $A' := (A'_1, A'_2)$ for $\Pi(1^{k+k_0})$ in the sense of IND-CPA (in the standard model) where*

$$\begin{aligned}
t' &= t + c_1 \cdot q_H \cdot k + q_D \cdot q_H \cdot (T_{\mathcal{E}}(k) + c_0 \cdot k), \text{ and} \\
\epsilon' &= (\epsilon - \frac{q_H}{2^{k_0 - 1}}) \cdot (1 - \frac{1}{2^l})^{q_D}.
\end{aligned}$$

Here $T_{\mathcal{E}}(k)$ is defined as before, and c_0 and c_1 depend on details of the underlying model of computation.

The specification of adversary A' is as follows:

Adversary: $A'_1(pk)$
$\tau \leftarrow \varepsilon$;
$\eta \leftarrow \varepsilon$;
run $A_1^{\mathcal{D}_{sk}, H}(pk)$
do while A_1 does not ask query h to $H(\cdot)$ nor ask query y' to $\mathcal{D}_{sk}(\cdot)$
if A_1 asks query h to $H(\cdot)$
if $h \notin \tau_h$
$H \leftarrow_R \{0,1\}^l$;

put (h, H) on the list τ;
answer A_1 with H;
else $h \in \tau_h$
answer A_1 with H such that $(h, H) \in \tau$
else if A_1 asks query y' to $\mathcal{D}_{sk}(\cdot)$
run $K(\tau, \eta, y', pk)$
K outputs x'
answer A_1 with x'
A_1 outputs (x_0, x_1, s)
$r_0, r_1 \leftarrow_R \{0,1\}^{k_0}$;
return $(x_0||r_0, x_1||r_1, s)$

End.

Adversary: $A_2'(x_0||r_0, x_1||r_1, s, y)$
$\eta \leftarrow y$;
run $A_2^{\mathcal{D}_{sk}, H}(x_0, x_1, s, y)$
do while A_1 does not ask query h to $H(\cdot)$ nor ask query y' to $\mathcal{D}_{sk}(\cdot)$
if A_1 asks query h to $H(\cdot)$
if $[h]_{k_0} == r_b$ where $[h]_{k_0}$ denotes the last k_0-bit of h.
stop A_1 and output b
else if $h \notin \tau_h$
$H \leftarrow_R \{0,1\}^l$;
put (h, H) on the list τ;
answer A_1 with H;
else $h \in \tau_h$
answer A_1 with H such that $(h, H) \in \tau$
else if A_1 asks query y' to $\mathcal{D}_{sk}(\cdot)$
run $K(\tau, \eta, y', pk)$
K outputs x'
answer A_1 with x'
A_1 outputs b
return b

End.

5 Examples: Enhanced Probabilistic Encryptions

In this section, we convert IND-CPA secure ones to IND-CCA2 (or NM-CCA2) secure ones. The ElGamal, Okamoto-Uchiyama, and Blum-Goldwasser encryption schemes [4, 7, 9] are candidates, since they are practical and secure in the IND-CPA sense under some reasonable assumptions; the decision Diffie-Hellman [5], p-subgroup, and factoring assumptions, respectively.

[**Enhanced ElGamal scheme**]

- Key-generator $\mathcal{K}$: $(pk, sk) \leftarrow \mathcal{K}(1^{k+k_0})$
- $pk := (p, q, g, y)$ and $sk := (p, q, g, s)$ where $y = g^s \bmod p$, $|p| = k + k_0$, $s \in \mathbb{Z}/q\mathbb{Z}$, $q|p-1$, and $\# < g >= q$.

[5] To our knowledge, Tsiounis and Yung first proved in [11] that the ElGamal encryption scheme is as secure as the decision Diffie-Hellman problem. In addition, they also presented a converted ElGamal scheme which is NM-CCA2 secure in the random oracle model. However, our conversion is far more efficient than theirs.

- Hash function H: $\{0,1\}^{k+k_0} \longrightarrow \mathbb{Z}/q\mathbb{Z}$.
- Encryption $\mathcal{E}$:

$$(y_1, y_2) := \mathcal{E}'_{pk}(x, r) := (g^{H(x||r)} \bmod p, (x||r) \oplus (y^{H(x||r)} \bmod p)),$$

where message $x \in \{0,1\}^k$ and $r \leftarrow_R \{0,1\}^{k_0}$.

- Decryption $\mathcal{D}$:

$$\mathcal{D}_{sk}(y_1, y_2) := \begin{cases} [y_2 \oplus (y_1^s \bmod p)]^k & \text{if } y_1 = g^{H(y_2 \oplus (y_1^s \bmod p))} \bmod p \\ \varepsilon \text{ (null)} & \text{otherwise} \end{cases}$$

where $[y_2 \oplus (y_1^s \bmod p)]^k$ denotes the first k-bit of $y_2 \oplus (y_1^s \bmod p)$.

Lemma 5.1 *In the random oracle model, the Enhanced ElGamal encryption scheme is secure in the sense of NM-CCA2 (or IND-CCA2) if the decision Diffie-Hellman problem is intractable.*

[Enhanced Okamoto-Uchiyama scheme]

- Key-generator $\mathcal{K}$: $(pk, sk) \leftarrow \mathcal{K}(1^{k+k_0})$
- $pk := (n, g, h, k)$ and $sk := (p, q)$ where $n = p^2 q$, $|p| = |q| = k + k_0$, $g \in (\mathbb{Z}/n\mathbb{Z})^*$ such that the order of $g_p := g^{p-1} \bmod p^2$ is p, and $h = g^n \bmod n$.
- Hash function H: $\{0,1\}^{k+k_0-1} \longrightarrow \mathbb{Z}/n\mathbb{Z}$.
- Encryption $\mathcal{E}$:

$$y := \mathcal{E}_{pk}(x, r) := g^{(x||r)} h^{H(x||r)} \bmod n,$$

where message $x \in \{0,1\}^k$ and $r \leftarrow_R \{0,1\}^{k_0-1}$.

- Decryption $\mathcal{D}$:

$$\mathcal{D}_{sk}(y) := \begin{cases} [\frac{L(y_p)}{L(g_p)} \bmod p]^k & \text{if } y = g^X h^{H(X)} \bmod n \\ \varepsilon \text{ (null)} & \text{otherwise} \end{cases}$$

where $y_p := y^{p-1} \bmod p^2$, $L(x) := \frac{x-1}{p}$, and $X := \frac{L(y_p)}{L(g_p)} \bmod p$.

Lemma 5.2 *In the random oracle model, the Enhanced Okamoto-Uchiyama encryption scheme is secure in the sense of NM-CCA2 (or IND-CCA2) if the p-subgroup problem (see [9]) is intractable.*

[Enhanced Blum-Goldwasser scheme]

- Key-generator $\mathcal{K}$: $(pk, sk) \leftarrow \mathcal{K}(1^{k+k_0})$
- $pk := (n)$ and $sk := (n, p, q)$ where $n = pq$, $|p| = |q| = k/2$, and p, q are William integers (i.e. $p, q \equiv 7 \pmod 8$ and primes).
- Hash function H: $\{0,1\}^{k+k_0} \longrightarrow \mathbb{Z}/n\mathbb{Z}$.
- Encryption $\mathcal{E}$:

$$(y_1, y_2) := \mathcal{E}_{pk}(x, r) := (H(x||r)^{2^{k+1}} \bmod n,\ x \oplus R).$$

where message $x \in \{0,1\}^k$, $r \leftarrow_R \{0,1\}^{k_0}$, and $R := LSB[H(x||r)^2] \,||LSB[H(x||r)^{2^2}]|| \cdots ||$ $LSB[H(x||r)^{2^k}]$.

- Decryption $\mathcal{D}$:

$$\mathcal{D}_{sk}(y_1, y_2) := \begin{cases} [y_2 \oplus \bar{R}]^k & \text{if } y_1 = H(y_2 \oplus \bar{R})^{2^{k+1}} \bmod n \\ \varepsilon \text{ (null)} & \text{otherwise} \end{cases}$$

where $\bar{R} := LSB[y_1^{2^{-k}}]||| \cdots ||LSB[y_1^{2^{-1}}]$.

Lemma 5.3 *In the random oracle model, the Enhanced Blum-Goldwasser encryption scheme is secure in the sense of NM-CCA2 (or IND-CCA2) if the factoring problem is intractable.*

6 Conclusion

This paper presented a simple and efficient conversion from a semantically secure public-key encryption scheme against *passive adversaries* to a non-malleable (or semantically secure) public-key encryption scheme against *chosen-ciphertext attacks* (*active adversaries*) in the random oracle model. Our conversion incurs minimum cost, i.e., only one random (hash) function operation. We also showed that our security reduction is (almost) optimally efficient, or exact security. Finally this paper presented some practical examples, the enhanced ElGamal, Blum-Goldwasser and Okamoto-Uchiyama schemes.

Acknowledgment

The second author would like to thank Phillip Rogaway for useful discussions.

References

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

[2] M. Bellare and P. Rogaway, "Optimal Asymmetric Encryption—How to encrypt with RSA" Advances in Cryptology –EUROCRYPT'94.

[3] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations Among Notions of Security for Public-Key Encryption Schemes" Advances in Cryptology –CRYPTO'98.

[4] M. Blum, and S. Goldwasser, "An efficient probabilistic public-key encryption scheme which hides all partial information", Proceeding of Crypto'84, LNCS 196, Springer-Verlag, pp.289-299 (1985).

[5] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen message attack", Advances in Cryptology –CRYPTO'98, Springer-Verlag, 1998.

[6] D. Dolev and C. Dwork and M. Naor, "Non-malleable cryptography", Proceeding of STOC91, pp 542–552.

[7] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," IEEE Transactions on Information Theory, IT-31, 4, pp.469–472, 1985.

[8] S. Goldwasser, and S. Micali, "Probabilistic Encryption", JCSS, vol.28, pp.270–299, 1984.

[9] T. Okamoto, and S. Uchiyama, "A New Public-Key Cryptosystem as Secure as Factoring", Advances in Cryptology –EUROCRYPT'98, Springer-Verlag, 1998.

[10] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems", Communications of ACM, 21, 2, pp.120-126, 1978.

[11] Y. Tsiounis and M. Yung, "On the Security of ElGamal based Encryption", PKC'98, January, 1998.