EPOC–3: Efficient Probabilistic Public-Key Encryption – V3 (Submission to P1363a)

Tatsuaki Okamoto[1] and David Pointcheval[2]

[1] NTT Labs, 1-1 Hikarinooka, Yokosuka-shi 239-0847 Japan.
E-mail: okamoto@isl.ntt.co.jp.

[2] Dépt d'Informatique, ENS – CNRS, 45 rue d'Ulm, 75230 Paris Cedex 05, France.
E-mail: David.Pointcheval@ens.fr – URL: http://www.di.ens.fr/~pointche.

May 2000

Abstract.

We describe a novel version of the public-key cryptosystems EPOC (Efficient Probabilistic Public-Key Encryption). EPOC–3 is a public-key encryption system that uses the Okamoto-Uchiyama one-way trapdoor function and two random functions (hash functions) as well as any symmetric encryption scheme such as the one-time pad, or any classical block-cipher.
We furthermore define a new variant of the High-Residuosity problem, named the Gap–High-Residuosity problem which is likely stronger than the classical High-Residuosity problem.

EPOC–3 therefore has several outstanding properties as follows:

1. with the one-time pad, EPOC–3 is semantically secure or non-malleable against chosen-ciphertext attacks (IND-CCA2 or NM-CCA2), in the random oracle model, under the Gap–High-Residuosity assumption.
2. with any symmetric encryption, EPOC–3 is semantically secure or non-malleable against chosen-ciphertext attacks (IND-CCA2 or NM-CCA2), in the random oracle model, under the Gap–High-Residuosity assumption, if the underlying symmetric encryption is simply semantically secure against passive attacks.
3. under the most practical environment in which public-key cryptosystems would be used, the efficiency of EPOC–3 is comparable to OAEP–RSA with a small exponent e (e.g., $2^{16}+1$). Although the encryption speed of EPOC–3 is slower than that of OAEP, the decryption speed is faster than that of OAEP. And this fast decryption is the most interesting property for an asymmetric cryptosystem since it is usually proceeded by a limited-power device, such as a smart-card.

The encryption scheme described in this contribution is obtained by combining two results: one on the trapdoor function technique is by Okamoto and Uchiyama, and the other on conversion techniques using random functions is by the authors.

Table of Contents

1 Background **3**
1.1 The Trapdoor One-Way Function . . . 3
1.1.1 Previous Work . . . 3
1.1.2 Definition of the Trapdoor One-Way Function . . . 3
1.1.3 Properties . . . 3
1.2 Provable Security . . . 4

2 Description of EPOC–3 **5**
2.1 Overview . . . 5
2.2 Key Generation: $\mathcal{K}$ . . . 5
2.3 Encryption: $\mathcal{E}$ . . . 6
2.4 Decryption: $\mathcal{D}$ . . . 7
2.5 Remark . . . 7

3 Security Assessment of EPOC–3 **7**

4 Attributes and Advantages of EPOC–3 **9**
4.1 Security of OTP—EPOC–3 . . . 9
4.2 Security of EPOC–3 with any Symmetric Encryption . . . 9
4.3 Efficiency . . . 10

5 Limitations **10**

6 Intellectual Property Statement **11**

1 Background

1.1 The Trapdoor One-Way Function

1.1.1 Previous Work

Since the seminal paper of Diffie and Hellman [11] which proposed the concept of the public-key cryptosystems (or trapdoor one-way functions), an extensive research has been driven by numerous cryptographers and mathematicians to realize it. Unfortunately, very few concrete techniques that seem to be secure have been found.

The first class of techniques is based of the Diffie-Hellman problem [11] which is the combination of the commutative property of the logarithm in a finite Abelian group and the intractability of the discrete logarithm problem. It has been used few years later by El Gamal [13]. Further applications appeared, using the elliptic/hyperelliptic curve-based Abelian groups [25, 20, 21, 22].

Another typical class of techniques is RSA-Rabin [39, 35, 36], 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. It includes many variants, even an elliptic curve version [23].

Several other techniques have been proposed such as the Goldwasser-Micali scheme [17] based on the Quadratic Residuosity problem, and many other sub-problems of $\mathcal{NP}$-complete problems (short vectors in lattices [2], the knapsack problem [9], the decoding problem [24], etc.). However they are not so efficient or not so secure, since some have been broken, or other have not been sufficiently investigated.

Among the RSA-Rabin and Diffie-Hellman-El Gamal techniques for realizing a trapdoor one-way function, no trapdoor function has been proven to be as secure as the underlying problems (providing a polynomial reduction).

Recently, the author Okamoto and Uchiyama [30] proposed a novel one-way trapdoor function that is practical, provably secure, and has some other interesting properties.

1.1.2 Definition of the Trapdoor One-Way Function

Let $n = p^2q$ be a large integer where p and q are two large primes, such that both $\gcd(p, q-1) = 1$ and $\gcd(q, p-1) = 1$. Let $g \in \mathbb{Z}_n^\star$, such that the order of the element $g_p = g^{p-1} \bmod p^2$ is p. Let h_0 be any element of $\mathbb{Z}_n^\star$ and set $h = h_0^n \bmod n$. Let $k = |p| = |q|$.

$$
\begin{array}{rccl}
f_{(n,g,h)} : & \{0,1\}^{k-1} \times \{0,1\}^{2k+c_0} & \longrightarrow & \mathbb{Z}_n^\star \\
 & (x \ , \ r) & \longmapsto & g^x h^r \bmod n \\
f_p^{-1} : & \mathbb{Z}_n^\star & \longrightarrow & \{0,1\}^{k-1} \\
 & y & \longmapsto & L(y^{p-1})/L(g^{p-1}) \bmod p
\end{array}
$$

$$\text{where } L(u) = (u-1)/p \text{ for any } u \equiv 1 \pmod{p}$$

Here c_0 is a positive constant.

1.1.3 Properties

1. **Probabilistic function:** The function $f_{(n,g,h)}(\cdot, r)$ is a probabilistic trapdoor function. For a given input x, the output depends on the random exponent r.

2. **One-wayness of the trapdoor function:** Inverting the function $f_{(n,g,h)}(\cdot, r)$ is proven to be as hard as factoring $n = p^2 q$.

3. **Indistinguishability of inputs:** To distinguish whether a given y comes from x_0 or x_1 with some random r, one has to break the High-Residuosity problem: does $c/g^{x_0} \in \mathbb{Z}_n^\star$ admits a p-th root in $\mathbb{Z}_n^\star$. This problem is comparable to the Quadratic Residuosity problem.

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 of a symmetric encryption scheme (e.g., triple-DES, IDEA, or any candidate of the AES), the computational load for this trapdoor function is comparable to the RSA function with a small exponent e (e.g., 3 or $2^{16} + 1$).

5. **Homomorphic property:** It has a homomorphic property

$$f(x_0, r_0) f(x_1, r_1) \bmod n = f(x_0 + x_1, r_2), \text{for some } r_2, \text{ if } m_0 + m_1 < p.$$

6. **Randomizability of ciphertext:** Even someone who does not know the secret key can change a ciphertext, $y = E(x, r)$, into another ciphertext, $y' = yh^{r'} \bmod n$, while preserving plaintext x (i.e., $y' = f(x, r'')$). Furthermore, the relationship between y and y' can be concealed, i.e., (y, y') and $(y, f(x', t))$, for any pair (x', t), are indistinguishable, under the High-Residuosity problem.
   Such a property is useful for privacy protecting protocols.

1.2 Provable Security

During a long time, heuristic security has been accepted by all the people and even the standard organizations. After many recent attacks against such "heuristically secure schemes" [8, 10, 18, 16], everybody realized the importance of provable security.

For public-key encryption, the strongest security notion, among all those that have been defined to capture the standard adversary scenarios, is by now called the *chosen-ciphertext security*. Indeed, it prevents [3] both the distinction of encrypted messages (semantic security [17]) and the malleability of ciphertexts [12] for an adversary who can ask the decryption of any ciphertext of her choice (the adaptive chosen-ciphertext attacks [37]).

A promising way to construct a practical public-key encryption scheme that reaches the chosen-ciphertext security is to convert a primitive trapdoor one-way function (such as RSA [39] or El Gamal [13]) by using *random functions*. Here, some hash functions, such as MD5 [38] or SHA-1 [26], are assumed to behave like ideally random functions. This so-called *random oracle model* [5] has already been widely used to provide efficient and provably secure schemes, for both signature [7, 34, 27, 4] and public-key encryption [6].

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 strong security arguments.

Two typical primitives of the trapdoor one-way functions are deterministic one-way permutations (e.g., the RSA function [39]) and probabilistic one-way functions (e.g., El Gamal [11, 13], Okamoto-Uchiyama [30] and Paillier [32] functions).

Bellare and Rogaway [6] presented a generic and efficient way to convert a trapdoor one-way permutation into a chosen-ciphertext secure scheme, in the random oracle model. The scheme created this way from the RSA function is often called OAEP. However, their method cannot be applied to probabilistic trapdoor one-way functions such as El Gamal, because it requires the permutation property.

Very recently the authors, together with other people [14, 15, 33, 29] proposed some generic conversions from any probabilistic trapdoor one-way function into a chosen-ciphertext secure encryption scheme. The first two conversions led to the EPOC [31] and PSEC [28] IEEE P1363a proposals.

The most recent conversion can apply to any (partially) trapdoor one-way function into a chosen-ciphertext secure encryption scheme, in an optimal way, from the computational point of view. Indeed, all the previous conversions required a re-encryption in the decryption phase to check the validity of the ciphertext. This new conversion just needs the basic decryption, without re-encryption. Furthermore, this conversion can be combined with a symmetric encryption scheme to reach high-speed rates.

2 Description of EPOC–3

2.1 Overview

This section describes the proposed third version of the public-key encryption scheme, EPOC, which is specified by a triplet $(\mathcal{K}, \mathcal{E}, \mathcal{D})$, where $\mathcal{K}$ is the key generation operation, $\mathcal{E}$ the encryption algorithm, and $\mathcal{D}$ the decryption one.

In that description, we assume $\mathsf{SymE} = (\mathsf{E}, \mathsf{D})$ to be a pair of symmetric encryption and decryption algorithms with symmetric key K, where the length of K is $kLen$.

The encryption algorithm E takes a key K and a plaintext m, and returns a ciphertext $\mathsf{E}_K(m)$. While the decryption algorithm D takes a key K and a ciphertext c, and returns the plaintext $\mathsf{D}_K(c)$.

Remark 1. A typical way to realize SymE is the one-time pad:

$$\mathsf{E}_K(m) = K \oplus m \qquad \mathsf{D}_K(c) = K \oplus c,$$

where $\oplus$ denotes the bit-wise exclusive-or operation. Therefore, $mLen = kLen$, where $mLen$ denotes the maximal message-size which can be securely encrypted.

2.2 Key Generation: $\mathcal{K}$

The input and output of the key generation algorithm $\mathcal{K}$ are as follows:

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

[Output] A pair $(\mathsf{pk}, \mathsf{sk})$ of matching public and secret keys. The public key consists of a tuple $(n, g, h, G, H, pLen, hLen, RLen, rLen)$, where $n = p^2q$, g and h are two elements

from $\mathbb{Z}_n^\star$, whereas G and H are two hash functions. The number $hLen$ denotes the output-size of the function H, whereas the output-size of the function G is $kLen$ (the key-length of the symmetric encryption scheme). The value $pLen$ is equal to the security parameter k and is the length of both p and q. Finally $rLen$ denotes the size of the random elements, while $RLen$ denotes the size of the *session keys*. The secret key sk consists of the pair (p, g_p), where $g_p = g^{p-1} \bmod p^2$.

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

- Choose two k-bit primes p, q ($|p| = |q| = k$), and compute $n \leftarrow p^2 q$. Here, $p-1 = p'u$ and $q-1 = q'v$ such that p' and q' are also prime integers, and $|u|$ and $|v|$ are $\mathcal{O}(\log k)$.
- Choose $g \in \mathbb{Z}_n^\star$ randomly such that the order of $g_p \leftarrow g^{p-1} \bmod p^2$ is p. (Note that $\gcd(p, q-1) = 1$ and $\gcd(q, p-1) = 1$.)
- Choose $h_0 \in \mathbb{Z}_n^\star$ randomly and compute $h \leftarrow h_0^n \bmod n$.
- Set $pLen \leftarrow k$ and $rLen \leftarrow 2k + c_0$ ($c_0 > 0$: constant). Set $RLen$ such that $RLen \leq k - 1$.
- Select two hash functions G: $\{0,1\}^{RLen} \longrightarrow \{0,1\}^{kLen}$ and H: $\{0,1\}^{3k+RLen+mLen} \longrightarrow \{0,1\}^{hLen}$, where $hLen$ is linearly dependent in k.

Note 1.
- The element g_p is a supplementary parameter that improves the efficiency of decryption, however g_p is a secret element which can be calculated from p and g.
- Furthermore, h can simply be $g^n \bmod n$ (still with $rLen = 2(k+1)$).
- G and H can be fixed by the system and shared by many/all the users.

2.3 Encryption: $\mathcal{E}$

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

[**Input**] Plaintext $m \in \{0,1\}^{mLen}$ along with the public-key pk and the symmetric encryption algorithm E.

[**Output**] Ciphertext $C = (c_1, c_2, c_3)$.

The operation of $\mathcal{E}$, on input m, $\mathsf{pk} = (n, g, h, G, H, pLen, hLen, RLen, rLen)$ and E, is as follows:

- Select $r \in \{0,1\}^{rLen}$ as well as $R \in \{0,1\}^{RLen}$ both uniformly, and compute $K \leftarrow G(R)$.
- Compute $c_1 \leftarrow g^R h^r \bmod n$.
- Then $c_2 \leftarrow \mathsf{E}_K(m)$ and $c_3 \leftarrow H(c_1, R, m)$.

2.4 Decryption: $\mathcal{D}$

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

[**Input**] Ciphertext $C = (c_1, c_2, c_3)$ along with the public-key pk, the secret key sk and D.

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

The operation of $\mathcal{D}$, on input a ciphertext $C = (c_1, c_2, c_3)$ along with the public-key $\mathsf{pk} = (n, g, h, G, H, pLen, hLen, RLen, rLen)$, the secret key $\mathsf{sk} = (p, g_p)$ and D, is as follows:

- Compute $c_p \leftarrow {c_1}^{p-1} \bmod p^2$, and $R' \leftarrow \frac{L(c_p)}{L(g_p)} \bmod p$, where $L : x \mapsto (x-1)/p$ for any $x = 1 \bmod p$.
- Compute $K' \leftarrow G(R')$ and $m' \leftarrow \mathsf{D}_{K'}(c_2)$.
- Check whether the following equation holds or not:

$$c_3 \stackrel{?}{=} H(c_1, R', m').$$

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

2.5 Remark

Since the domains of G and H are fixed by the parameters of $RLen$ and others, only R' with $R' \in \{0,1\}^{RLen}$ is accepted by the decryption procedure, $\mathcal{D}$. (Note that the domains of G and H in the conversion of [29] are fixed by the domain of the underlying encryption function and other parameters.) More explicitly, in $\mathcal{D}$, check if $R' \in \{0,1\}^{RLen}$ (see a recent remark [19]).

We can use any random-like hash functions G and H for EPOC. However, EPOC–3 can be proven to be secure if G and H are assumed to behave like random functions, while no formal security is guaranteed if they are practical random-like hash functions. Nevertheless, some candidates have been suggested in the past, namely by Bellare and Rogaway [6].

3 Security Assessment of EPOC–3

This section reviews our results on the security of EPOC–3. They are easily obtained from [29].

Definition 1 (Factoring Assumption). Let $\mathcal{K}$ be the key generator algorithm of above EPOC–3 which, on input k, outputs $n = p^2 q$ for $|p| = |q| = k$ and p, q prime integers. The *factoring (FACT) problem* is, given (k, n), to find (p, q).

The *factoring problem* is *intractable*, if for any probabilistic polynomial time machine $\mathcal{A}$, for any constant c, for sufficiently large k,

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

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

The assumption that the *factoring problem* is *intractable* is called the *factoring assumption.*

Definition 2 (High-Residuosity Assumption). Let $\mathcal{K}$ be the key generator algorithm of EPOC–3, and $(n, g, h, pLen, rLen)$ be a part of the public-key. Let $b \in \{0,1\}$ and $r \in \{0,1\}^{rLen}$ be randomly and uniformly chosen. Set $C \leftarrow g^b h^r \bmod n$.

The *High-Residuosity (HR) problem* (a.k.a. the *p-subgroup problem* in that specific situation) is *intractable* if for any probabilistic polynomial time machine $\mathcal{A}$, for any constant c, for sufficiently large k $(= pLen)$,

$$\Pr[\mathcal{A}(n, g, h, pLen, rLen, C) = b] < 1/2 + 1/k^c.$$

The probability is taken over the coin flips of $\mathcal{K}$ and $\mathcal{A}$, as well as the random choice of b and r.

The assumption that the *High-Residuosity problem* is *intractable* is called the *High-Residuosity assumption.*

Definition 3 (Gap–High-Residuosity Assumption). Let $\mathcal{K}$ be the key generator algorithm of EPOC–3, and $(n, g, h, pLen, rLen)$ be a part of the public-key.

The *Gap–High-Residuosity (GHR) problem* is *intractable*, if for any probabilistic polynomial time machine $\mathcal{A}^{HR}$, with a full access to an oracle that perfectly answers the HR problem, for any constant c, for sufficiently large k $(= pLen)$,

$$\Pr[\mathcal{A}^{HR}(n, g, h, pLen, rLen) = (p, q)] < 1/k^c.$$

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

The assumption that the *Gap–High-Residuosity problem* is *intractable* is called the *Gap–High-Residuosity assumption.*

Definition 4 (Security of Symmetric Encryption). Let $\mathcal{A}$ be an adversary that runs in two stages. In the first stage, $\mathcal{A}$ endeavors to come up with a pair of equal-length messages, m_0 and m_1, along with some state information s, where $|m_0| = |m_1| \leq kLen^a$, for some constant a. In the second stage, $\mathcal{A}$ is given a ciphertext $c \leftarrow \mathsf{E}_K(m_b)$, where $K \in \{0,1\}^{kLen}$ and $b \in \{0,1\}$ are randomly and uniformly chosen.

SymE is *secure against passive attacks* if for any probabilistic polynomial time machine $\mathcal{A}$, for any constant d, for sufficiently large $kLen$,

$$\Pr[\mathcal{A}(kLen, m_0, m_1, s, c) = b] < 1/2 + 1/kLen^d.$$

The probability is taken over the coin flips of $\mathcal{A}$ as well as the random choice of K and b.

Theorem 1 (OTP—EPOC–3). *Let SymE be the one-time pad, and thus $mLen = kLen$. Let $hLen = pLen/a$ for some constant a. OTP—EPOC–3 is chosen-ciphertext secure in the random oracle model, provided that the GHR assumption holds.*

Theorem 2 (EPOC–3). *Let $hLen = pLen/a$ for some constant a. EPOC–3 is chosen-ciphertext secure in the random oracle model, provided that the GHR assumption holds and that the underlying SymE is secure against passive attacks, for suitable $kLen$ and $mLen$.*

Remark 2. We can also give the concrete efficiency analysis of the reduction for proving the security, and show that our reduction is tight [29], and even optimal since the probability of breaking the GHR problem is almost the same as the advantage of an adaptive adversary in breaking the chosen-ciphertext security.

4 Attributes and Advantages of EPOC–3

4.1 Security of OTP—EPOC–3

If the Gap–High-Residuosity (GHR) assumption holds, EPOC–3 with one-time pad is secure in the strongest sense, in the random oracle model, if the parameters are appropriately selected.

Note that this assumption is quite new. But one can easily show that if the GHR problem is not intractable, then the FACT and HR problems are equivalent. However this latter equivalence is very unlikely, and certainly more unlikely than the tractability of the HR problem.

Scheme	Security	Number Theoretical Assumption	Hash Function Assumption
OTP—EPOC–3	IND-CCA2	GHR	Random Oracle
OTP—EPOC–2	IND-CCA2	FACT	Random Oracle
OTP—EPOC–1	IND-CCA2	HR	Random Oracle
OAEP–RSA	IND-CCA2	RSA	Random Oracle
RSA	INV-CPA	RSA	None

4.2 Security of EPOC–3 with any Symmetric Encryption

If the Gap–High-Residuosity (GHR) assumption holds, and the underlying symmetric encryption is secure against passive attacks, EPOC–3 with the symmetric encryption is secure in the strongest sense, in 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 encryption is secure only against passive attacks (we do not care about active attacks), EPOC–3 guarantees security against adaptive chosen-ciphertexts attacks (IND-CCA2).

An additional property of EPOC–3 (as other EPOC versions) is authentication and integrity without using any MAC function. That is, the recipient can confirm whether the decrypted message is the same as the one the originator sent.

Finally, it also provides a key distribution with session key encryption and then a symmetric multi-message encryption which achieves chosen-ciphertext security. Indeed,

the ciphertext can be split, the first part c_1 is a constant overhead, and then the second part (c_2, c_3) can be reiterated with many plaintexts:

- Select $r \in \{0,1\}^{rLen}$ and $R \in \{0,1\}^{RLen}$ uniformly.
- Compute $c_1 \leftarrow g^R h^r \bmod n$ and $K \leftarrow G(R)$.
- Then, for any message m_i, compute $c_{2,i} \leftarrow \mathsf{E}_K(m_i)$ and $c_{3,i} \leftarrow H(c_1, R, m_i)$, and send the tuple $(c_1, c_{2,i}, c_{3,i})$.

4.3 Efficiency

The OTP—EPOC–3 is a very efficient scheme among all the FACT/RSA-based encryption schemes (see Figure 1). Since it furthermore allows symmetric integration with multi-message encryption, it achieves an unbeatable efficiency, namely from the decryption point of view, which is the most important one, since it is often proceeded by a limited-power device (smart card).

Scheme		Security parameter k $\lvert p\rvert = \lvert q\rvert = k$		Security level ℓ $\lvert n\rvert = 2\ell$	
		Encryption Cost	Decryption Cost	Encryption Cost	Decryption Cost
OTP—EPOC–3	$n = p^2q$	$41k[k]$+2H	$6k[k]$+2H	$12\ell[\ell]$+2H	$2\ell[\ell]$+2H
OTP—EPOC–2		$41k[k]$+2H	$29k[k]$+2H	$12\ell[\ell]$+2H	$9\ell[\ell]$+2H
OTP—EPOC–1		$41k[k]$+1H	$29k[k]$+1H	$12\ell[\ell]$+2H	$9\ell[\ell]$+2H
OAEP–RSA	$n = pq$	$68[k]$+2H	$3k[k]$+2H	$68[\ell]$+2H	$3\ell[\ell]$+2H
RSA	$e = 2^{16} + 1$	$68[k]$	$3k[k]$	$68[\ell]$	$3\ell[\ell]$

Fig. 1: Efficiency comparison of the different FACT/RSA-based encryption schemes, where "H" denotes the cost of a hashing and "$u[k]$" represents u multiplications modulo a k-bit integer

Under the most practical environment of using public-key cryptosystems, where a public-key cryptosystem is used associated with symmetric encryption (e.g., triple-DES, IDEA or any candidate of the AES), a typical example of the parameters is as follows: first, any message and any list of messages can be encrypted (i.e., $mLen = \star$), then $gLen = kLen = 128$ (or 256, according to the block-cipher), $hLen = 64$, and $pLen = RLen = k = 384$ (to provide a similar security level than RSA–1024).

Compared to OAEP-RSA with a small exponent e (e.g.,$2^{16} + 1$), although the encryption speed of EPOC–3 is slower than that of OAEP, the decryption speed is much faster.

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] M. Abdalla, M. Bellare, and P. Rogaway. DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem. Submission to IEEE P1363a. September 1998.
Available from `http://grouper.ieee.org/groups/1363/`.

[2] M. Ajtai and C. Dwork. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. In *Proc. of the 29th STOC*, pages 284–293. ACM Press, New York, 1997.

[3] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among Notions of Security for Public-Key Encryption Schemes. In *Crypto '98*, LNCS 1462, pages 26–45. Springer-Verlag, Berlin, 1998.

[4] M. Bellare and P. Rogaway. PSS: Provably Secure Encoding Method for Digital Signatures. Submission to IEEE P1363a. August 1998. Available from `http://grouper.ieee.org/groups/1363/`.

[5] M. Bellare and P. Rogaway. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In *Proc. of the 1st CCS*, pages 62–73. ACM Press, New York, 1993.

[6] M. Bellare and P. Rogaway. Optimal Asymmetric Encryption – How to Encrypt with RSA. In *Eurocrypt '94*, LNCS 950, pages 92–111. Springer-Verlag, Berlin, 1995.

[7] M. Bellare and P. Rogaway. The Exact Security of Digital Signatures – How to Sign with RSA and Rabin. In *Eurocrypt '96*, LNCS 1070, pages 399–416. Springer-Verlag, Berlin, 1996.

[8] D. Bleichenbacher. A Chosen Ciphertext Attack against Protocols based on the RSA Encryption Standard PKCS #1. In *Crypto '98*, LNCS 1462, pages 1–12. Springer-Verlag, Berlin, 1998.

[9] B. Chor and R. L. Rivest. A Knapsack Type Public Key Cryptosystem based on Arithmetic in Finite Fields. In *Crypto '84*, LNCS 196, pages 54–65. Springer-Verlag, Berlin, 1985.

[10] J. S. Coron, D. Naccache, and J. P. Stern. On the Security of RSA Padding. In *Crypto '99*, LNCS 1666, pages 1–18. Springer-Verlag, Berlin, 1999.

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

[12] D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *Proc. of the 23rd STOC*. ACM Press, New York, 1991.

[13] T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Transactions on Information Theory*, IT–31(4):469–472, July 1985.

[14] E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In *PKC '99*, LNCS 1560, pages 53–68. Springer-Verlag, Berlin, 1999.

[15] E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *Crypto '99*, LNCS 1666, pages 537–554. Springer-Verlag, Berlin, 1999.

[16] M. Girault and J. F. Misarsky. Cryptanalysis of Countermeasures Proposed for Repairing IOS/IEC 9796-1. In *Eurocrypt '2000*, LNCS. Springer-Verlag, Berlin, 2000.

[17] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.

[18] F. Grieu. A Chosen Message Attack on ISO/IEC 9796-1 Signature Scheme. In *Eurocrypt '2000*, LNCS. Springer-Verlag, Berlin, 2000.

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

[20] N. Koblitz. Elliptic Curve Cryptosystems. *Mathematics of Computation*, 48(177):203–209, January 1987.

[21] N. Koblitz. A Family of Jacobians Suitable for Discrete Log Cryptosystems. In *Crypto '88*, LNCS 403, pages 94–99. Springer-Verlag, Berlin, 1989.

[22] N. Koblitz. Hyperelliptic Cryptosystems. *Journal of Cryptology*, 1:139–150, 1989.

[23] K. Koyama, U. Maurer, T. Okamoto, and S. Vanstone. New Public-Key Schemes Based on Elliptic Curves over the Ring Z_n. In *Crypto '91*, LNCS 576, pages 252–266. Springer-Verlag, Berlin, 1992.

[24] R. J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. *DSN progress report*, 42-44:114–116, 1978. Jet Propulsion Laboratories, CALTECH.

[25] V. Miller. Uses of Elliptic Curves in Cryptography. In *Crypto '85*, LNCS 218, pages 417–426. Springer-Verlag, Berlin, 1986.

[26] NIST. *Secure Hash Standard* (SHS). Federal Information Processing Standards PUBlication 180–1, April 1995.

[27] K. Ohta and T. Okamoto. On Concrete Security Treatment of Signatures Derived from Identification. In *Crypto '98*, LNCS 1462, pages 354–369. Springer-Verlag, Berlin, 1998.

[28] T. Okamoto, E. Fujisaki, and H. Morita. PSEC: Provably Secure Elliptic Curve Encryption Scheme. Submission to IEEE P1363a. March 1999. Available from `http://grouper.ieee.org/groups/1363/`.

[29] T. Okamoto and D. Pointcheval. OCAC: an Optimal Conversion for Asymmetric Cryptosystems, 2000. Manuscript.

[30] T. Okamoto and S. Uchiyama. A New Public Key Cryptosystem as Secure as Factoring. In *Eurocrypt '98*, LNCS 1403, pages 308–318. Springer-Verlag, Berlin, 1998.

[31] T. Okamoto, S. Uchiyama, and E. Fujisaki. EPOC: Efficient Probabilistic Public-Key Encryption. Submission to IEEE P1363a. November 1998. Available from `http://grouper.ieee.org/groups/1363/`.

[32] P. Paillier. Public-Key Cryptosystems Based on Discrete Logarithms Residues. In *Eurocrypt '99*, LNCS 1592, pages 223–238. Springer-Verlag, Berlin, 1999.

[33] D. Pointcheval. The Composite Discrete Logarithm and Secure Authentication. In *PKC '2000*, LNCS 1751, pages 113–128. Springer-Verlag, Berlin, 2000.

[34] D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. In *Eurocrypt '96*, LNCS 1070, pages 387–398. Springer-Verlag, Berlin, 1996.

[35] M. O. Rabin. Digitalized Signatures. In R. Lipton and R. De Millo, editors, *Foundations of Secure Computation*, pages 155–166. Academic Press, New York, 1978.

[36] M. O. Rabin. Digitalized Signatures and Public Key Functions as Intractible as Factorization. Technical Report MIT/LCS/TR-212, Massachusetts Institute of Technology – Laboratory for Computer Science, January 1979.

[37] C. Rackoff and D. R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Crypto '91*, LNCS 576, pages 433–444. Springer-Verlag, Berlin, 1992.

[38] R. Rivest. The MD5 Message-Digest Algorithm. RFC 1321, The Internet Engineering Task Force, April 1992.

[39] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.