Nonce-aware Encryption

Ming-Luen Wu
National Kaohsiung Normal University
Department of Mathematics
No.116, Heping 1st Rd., Lingya District, Kaohsiung City 802
Taiwan
mlwu@nknucc.nknu.edu.tw

Abstract: As an alternative perspective on designing IND-CCA2 encryption, we introduce a new security notion, nonce-awareness, for encryption. An encryption scheme is nonce-aware if it is computationally infeasible to produce a valid ciphertext without knowing the associated nonce. We also show that two remarkable IND-CCA2 encryption schemes are nonce-aware.

Key–Words: Encryption, Security, Nonce-awareness, Indistinguishability, Chosen-ciphertext attack

1 Introduction

Two different goals for encryptions have been considered: indistinguishability (IND) [11, 15] and non-malleability (NM) [10]. IND requires that it be infeasible for an adversary to distinguish between the ciphertexts of any two messages, even if the two original messages are given. NM requires that an adversary given a challenge ciphertext be unable to obtain a different ciphertext such that the plaintexts underlying these two ciphertexts are meaningfully related.

These goals are often considered under three different active attacks: chosen-plaintext attack (CPA), non-adaptive chosen-ciphertext attack (CCA1) [17], and adaptive chosen-ciphertext attack (CCA2) [18]. Under CPA the adversary can obtain ciphertext of any plaintext. Public-key encryption schemes have to be safe against CPA. Under CCA1 the adversary can gain access to an oracle for the decryption function only for the period of time preceding his being given the challenge ciphertext. In other words, adversary's queries to the decryption oracle cannot depend on the challenge ciphertext. However, under CCA2 the adversary can continue to have access to a decryption oracle even after obtaining the challenge ciphertext. The only restriction is that the adversary cannot make the decryption oracle decrypt the challenge ciphertext.

One can combine the goals with the attacks to gain various security notions: IND-CPA, IND-CCA2, NM-CPA, and NM-CCA2. Their relations have been studied in [1, 5]. In particular, it is proved that IND-CCA2 is equivalent to NM-CCA2 [1]. Among these notions of security, IND-CCA2 is strong and very useful for encryption schemes [21]. As an alternative perspective on designing IND-CCA2 encryption, the notion of plaintext-awareness (PA) is brought in [1, 2, 4]. PA0 (respectively, PA1) requires that it be infeasible for an adversary to yield ciphertexts without knowing the associated messages, even if he can make a single oracle query (respectively, a polynomial number of oracle queries). Further, PA2 captures eavesdropping capability by providing

the adversary with an additional encryption oracle that returns ciphertexts whose decryption he may not know. Relations among these notions (PA0, PA1, PA2, IND-CPA, IND-CCA1, and IND-CCA2) are studied in [2]. Especially, PA2 plus IND-CPA implies IND-CCA2.

This paper introduces a new security notion, nonce-awareness (NA), for encryption to provide another perspective on designing IND-CCA2 encryption. Intuitively nonce-awareness requires that it be infeasible for an adversary to yield ciphertexts without knowing the associated nonce, even if he can make a polynomial number of oracle (including encryption oracle) queries. We classify nonce-awareness into two classes: NA1 and NA2. In NA1 plaintext can be extracted using only the associated nonce, but cannot in NA2. Obviously NA1 implies PA2. We also show that two famous IND-CCA2 encryption schemes are NA1: the signed ElGamal encryption [19] and the Cramer-Shoup scheme [6, 7].

The rest of this paper is organized as follows. In Section 2, basic terms and nonce-awareness are defined. Then in Section 3, the signed ElGamal encryption [19] and the Cramer-Shoup scheme [6, 7] are discussed. Section 4 concludes.

2 Key Terms

We first review basic terms, and then define nonce-awareness.

2.1 Basic Terms

Notation. We denote by ε the empty string, by $|m|$ the length of a string m, and by [] the empty list. Given a list L and an element c, $L@c$ denotes the list consisting of the elements in L followed by c. Let A be an algorithm. The notation $state[A]$ denotes the state information of A. By $A(\cdot)$ we denote that A has one input. By $A(\cdot,\ldots,\cdot)$ we denote that A has several inputs. A may be deterministic or probabilistic. If A is a probabilistic algorithm then $A(\cdot,\ldots,\cdot;R)$ denotes that A takes R as random coins. The notation $y \leftarrow A(x)$ denotes that y is obtained by running A on input x. By $A^O(x)$ we denote that A may query the oracle O on the input x. The notation $x \stackrel{u}{\leftarrow} S$, for a set S, means that x is randomly selected from S according to a uniform probability distribution. If α is neither an algorithm nor a set then $x \leftarrow \alpha$ is a assignment statement. Let B be a boolean function. The notation $(B(y_n) : \{y_i \leftarrow A_i(x_i)\}_{1\le i\le n})$ denotes the event that $B(y_n)$ is TRUE after the value y_n is obtained by successively running algorithms $A_1,\ldots,A_n$ on inputs $x_1,\ldots,x_n$. The statement

$$\Pr[B(y_n) : \{y_i \leftarrow A_i(x_i)\}_{1\le i\le n}] = p$$

means that the probability that $B(y_n)$ is TRUE after the value y_n is obtained by running algorithms $A_1,\ldots,A_n$ on inputs $x_1,\ldots,x_n$ is p, where the probability is over the random choices of the probabilistic algorithms involved.

Definition 1 (Negligible Functions)**.** *We call a function $f : \mathbb{N} \to \mathbb{R}$ negligible if for every positive polynomial $P(\cdot)$, there exists an n_0 such that for all $n > n_0$,*

$$f(n) < \frac{1}{P(n)}.$$

Definition 2 (DDH Assumption)**.** *Let G be a group of large prime order q. For any polynomial-time algorithm A that outputs a*

single bit, we define σ to be

$$\begin{aligned}&|Pr[A(G,q,g_1,g_2,u_1,u_2)=1\\&\quad \mid g_1,g_2,u_1,u_2 \text{ chosen randomly from } G]\\&-Pr[A(G,q,g_1,g_2,u_1,u_2)=1\\&\quad \mid g_1,g_2 \text{ chosen randomly from } G,\\&\quad \text{and } u_1=g_1^r \text{ and } u_2=g_2^r\\&\quad \text{for random } r\in\mathbb{Z}_q]|\end{aligned}$$

The decision Diffie-Hellman (DDH) assumption is that, for all polynomial-time algorithms A, σ is negligible as a function of the security parameter.

Definition 3. *An asymmetric encryption scheme is a triple of algorithms $(\mathcal{G},\mathcal{E},\mathcal{D})$.*

- *Key generation algorithm $\mathcal{G}$: This is a probabilistic polynomial-time algorithm $\mathcal{G}(1^k)=(sk,pk)$, where 1^k is a secure parameter, sk and pk are a pair of decryption and encryption keys, each of size $O(k^a)$ for $a\in\mathbb{N}$ a constant.*
- *Encryption algorithm $\mathcal{E}$: This is often a probabilistic algorithm $\mathcal{E}(pk,\ m)=c$, where m is a message in the message space $\mathcal{M}$, and c is the corresponding ciphertext in the ciphertext space $\mathcal{C}$.*
- *Decryption algorithm $\mathcal{D}$: This is a deterministic algorithm $\mathcal{D}(sk,c)=m$ (i.e., $\mathcal{D}(sk,\ \mathcal{E}(pk,m))=m$) for every $m\in\mathcal{M}$, where sk and pk are a pair of decryption and encryption keys.*

If algorithm $\mathcal{E}$ is probabilistic then the encryption scheme is called probabilistic encryption.

Definition 4 (IND-CPA, IND-CCA1, IND-CCA2)**.** *[1] Let $(\mathcal{G},\mathcal{E},\mathcal{D})$ be an asymmetric encryption scheme and let $\mathcal{A}=(A_1,A_2)$ be an adversary. For $atk\in\{CPA,CCA1,CCA2\}$ and $k\in\mathbb{N}$ let*

$$\begin{aligned}\sigma=Pr[&b'=b:\\&(sk,pk)\leftarrow\mathcal{G}(1^k),\\&(m_0,m_1,s)\leftarrow A_1^{O_1}(pk)\\&\quad \text{where } |m_0|=|m_1|,\\&b\overset{u}{\leftarrow}\{0,1\},\\&c\leftarrow\mathcal{E}(pk,m_b),\\&b'\leftarrow A_2^{O_2}(s,m_0,m_1,pk,c)]-\frac{1}{2},\end{aligned}$$

where

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 $O_2(\cdot)=\mathcal{D}(sk,\cdot)$.

An asymmetric encryption scheme $(\mathcal{G},\mathcal{E},\mathcal{D})$ is secure in the sense of IND-atk if for every probabilistic polynomial-time adversary $\mathcal{A}$, and all sufficiently large k, the adversary's advantage σ is negligible in the security parameter k.

Plaintext-awareness is formally defined using two experiments. We first describe the two experiments [2, 9].

The REAL experiment:

1. $(pk,sk)\leftarrow\mathcal{G}(1^k)$;
   $CLIST\leftarrow[]$;
   Choose random coins $R[A],R[P]$ for A,P, respectively;
   $State[P]\leftarrow\varepsilon$.

2. Run A on input pk and coins $R[A]$ until it halts.

   - If A makes query $(decryption,c)$, then
     $m\leftarrow\mathcal{D}(sk,c)$;
     return m to A.

Note that A may not query the decryption oracle with any ciphertext appearing on $CLIST$.

- If A makes query $(encryption, aux)$ with query information aux, then
  $(m, state[P]) \leftarrow P(aux, state[P]; R[P])$;
  $c \leftarrow \mathcal{E}(pk, m)$;
  $CLIST \leftarrow CLIST@c$;
  return c to A.

3. Let x denote the output of A.

The EXTR experiment:

1. $(pk, sk) \leftarrow \mathcal{G}(1^k)$;
   CLIST$\leftarrow$ [];
   Choose random coins $R[A]$, $R[P]$, $R[A^*]$ for A, P, A^*, respectively;
   $state[P] \leftarrow \varepsilon$;
   $state[A^*] \leftarrow (pk, R[A])$.

2. Run A on input pk and coins $R[A]$ until it halts.

   - If A makes query $(decryption, c)$, then
     $(m, state[A^*]) \leftarrow A^*(c, CLIST, state[A^*]; R[A^*])$;
     return m to A.
     Note that A may not query the decryption oracle with any ciphertext appearing on $CLIST$.
   - If A makes query $(encryption, aux)$ with query information aux, then
     $(m, state[P]) \leftarrow P(aux, state[P]; R[P])$;
     $c \leftarrow \mathcal{E}(pk, m)$;
     $CLIST \leftarrow CLIST@c$;
     return c to A.

3. Let x denote the output of A.

Definition 5 (Plaintext-awareness)**.** *An asymmetric encryption scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is plaintext-aware2 (PA2) if for any polynomial-time ciphertext creator A, there exists a polynomial-time pa2-extractor A^* such that for all polynomial-time plaintext creators P and polynomial time distinguisher D the advantage σ of A relative to P, D and A^**

$$\sigma = |Pr[D(x) = 1|A\ plays\ REAL] - Pr[D(x) = 1|A\ plays\ EXTR]|$$

is negligible in the security parameter k. Distinguisher D tries to distinguish between the cases that A interacts with the REAL experiment or the EXTR experiment.

An asymmetric encryption scheme is plaintext-aware1 (PA1) if for any polynomial-time ciphertext creator A that makes no encryption queries, there exists a polynomial-time pa1-extractor A^* such that for all polynomial time distinguisher D the advantage σ of A relative to D and A^*

$$\sigma = |Pr[D(x) = 1|A\ plays\ REAL] - Pr[D(x) = 1|A\ plays\ SIMU]|$$

is negligible in the security parameter k.

Plaintext-aware0 (PA0) is the same as PA1 except that ciphertext creator A makes exactly one decryption query.

Definition 6 (Simulatable Encryption Scheme)**.** *[9] An asymmetric encryption scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is simulatable if there exist two polynomial-time Turing machines (f, f^{-1}) such that:*

- *f is a deterministic Turing machine that takes the public key pk and an element $r \in \{0,1\}^\ell$ as input, and outputs elements of $\mathcal{C}$. For simplicity, f will be represented as a function from $\{0,1\}^\ell$ to $\mathcal{C}$ and the public key input will be suppressed.*

- *f^{-1} is a probabilistic Turing machine that takes the public key pk and an element $c \in \mathcal{C}$ as input, and outputs elements of $\{0,1\}^{\ell}$. Again, f^{-1} will be represented as a function from $\mathcal{C}$ to $\{0,1\}^{\ell}$ and the public key input will be suppressed.*
- *$f(f^{-1}(c)) = c$ for all $c \in \mathcal{C}$.*
- *There exists no polynomial-time attacker A that has a non-negligible advantage in winning the following experiment:*
  1. *The challenger generates a key pair $(pk, sk) = \mathcal{G}(1^k)$ and randomly chooses a bit $b \in \{0,1\}$.*
  2. *The attacker executes A on the input pk. The attacker has access to an oracle $\mathcal{O}_f$ that takes no input, generates a random element $r \in \{0,1\}^{\ell}$, and returns r if $b = 0$ and $f^{-1}(f(r))$ if $b = 1$. The attacker terminates by outputting a guess b' for b.*

  The attacker wins if $b = b'$. The attacker's advantage is defined to be:

$$|Pr[b = b'] - 1/2|.$$

- *There exists no polynomial-time attacker A that has a non-negligible advantage in winning the following experiment:*
  1. *The challenger generates a key pair $(pk, sk) = \mathcal{G}(1^k)$, an empty list CLIST, and a bit b chosen randomly from $\{0,1\}$.*
  2. *The attacker executes A on the input pk. The attacker has access to two oracles:*
     - *An encryption oracle that takes a message $m \in \mathcal{M}$ as input and returns an encryption c. If $b = 0$, then the oracle returns $c = \mathcal{E}(pk, m)$. If $b = 1$, then the oracle returns $c = f(r)$, for some randomly chosen $r \in \{0,1\}^{\ell}$. In either case c is added to CLIST.*
     - *A decryption oracle that takes an encryption $c \in \mathcal{C}$ as input and returns $\mathcal{D}(sk, c)$. The attacker may not query the decryption oracle on any $c \in CLIST$.*

     The attacker terminates by outputting a guess b' for b.

  The attacker wins if $b = b'$. The attacker's advantage is defined to be:

$$|Pr[b = b'] - 1/2|.$$

A family of hash functions is said to be universal one-way if it is computationally infeasible for an adversary to choose an input x, draw a function H at random from the family, and then find a different input y such that $H(x) = H(y)$ [16]. Such hash function families are also called target collision-resistant. Note that a stronger notion is that of a collision-resistant family of hash functions. Here, it is computationally infeasible for an adversary to to find a pair (x, x') with $x \neq x'$ such that $H(x) = H(x')$ if H is chosen at random from a family of hash functions [8].

The random oracle model provides a mathematical model of an ideal hash function [3]. In this model, a hash function h is chosen randomly from a family of hash functions, and we are only permitted oracle access to the random function h to obtain a random hash value. This means that we are not given a formula to compute the value of the function h. Hence, the only way to compute a value $h(x)$ is to query the oracle (random function).

In the generic model, generic algorithms for group G do not exploit any special properties of the encodings of group elements, other than the property that each group element is encoded as a unique string [20]. The data of a generic algorithm is partitioned into group elements in G and non-group data. In this paper, a generic adversary A — attacking an encryption scheme — is an interactive algorithm that interacts with a decryption oracle. The following generic steps are counted:

- group operations,
- queries to the hash oracle H,
- interactions with a decryption oracle.

Adversary A selects the next generic step depending on the non-group input and on previous collisions of group elements.

2.2 Nonce-awareness

Let $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ be an asymmetric probabilistic encryption scheme. To formalize nonce-awareness, we fist define two experiments.

The REAL experiment:

1. $(pk, sk) \leftarrow \mathcal{G}(1^k)$;
   $CLIST \leftarrow []$;
   Choose random coins $R[\mathcal{E}], R[A], R[P]$ for $\mathcal{E}$, A, P, respectively;
   $state[P] \leftarrow \varepsilon$.

2. Run A on input pk and coins $R[A]$ until it halts.

   - If A makes query $(nonce, c)$, then nonce oracle returns r to A where r is the associated nonce for c. Note that A may not query any ciphertext appearing on $CLIST$.
   - If A makes query $(encryption, aux)$ with query information aux, then
     $(m, state[P]) \leftarrow P(aux, state[P]; R[P])$;
     $c \leftarrow \mathcal{E}(pk, m; R[\mathcal{E}])$;
     $CLIST \leftarrow CLIST@c$;
     return c to A.

3. Let x denote the output of A.

The EXTR experiment:

1. $(pk, sk) \leftarrow \mathcal{G}(1^k)$;
   CLIST$\leftarrow$ [];
   Choose random coins $R[\mathcal{E}]$, $R[A]$, $R[P]$, $R[A^*]$ for $\mathcal{E}$, A, P, A^*, respectively;
   $state[P] \leftarrow \varepsilon$;
   $state[A^*] \leftarrow (pk, R[A])$.

2. Run A on input pk and coins $R[A]$ until it halts.

   - If A makes query $(nonce, c)$, then $(r, state[A^*]) \leftarrow A^*(c, CLIST, state[A^*]; R[A^*])$; return r to A. Note that A may not query any ciphertext appearing on $CLIST$.
   - If A makes query $(encryption, aux)$ with query information aux, then
     $(m, state[P]) \leftarrow P(aux, state[P]; R[P])$;
     $c \leftarrow \mathcal{E}(pk, m; R[\mathcal{E}])$;
     $CLIST \leftarrow CLIST@c$;
     return c to A.

3. Let x denote the output of A.

Definition 7 (Nonce-awareness)**.** *An asymmetric encryption scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is nonce-aware (NA) if for any polynomial-time ciphertext creator A, there exists a polynomial-time na-extractor A^* such that for all polynomial-time plaintext creators P and polynomial-time*

distinguisher D the advantage σ of A relative to P, D and A^*

$$\sigma = |Pr[D(x) = 1|A\ plays\ REAL] - Pr[D(x) = 1|A\ plays\ EXTR]|$$

is negligible in the security parameter k. Distinguisher D tries to distinguish between the cases that A interacts with the REAL experiment or the EXTR experiment,

If existence of na-extractor implies that of pa2-extractor, then nonce-awareness is called NA1, otherwise NA2.

3 Two NA1 Encryption Schemes

3.1 The Signed ElGamal Encryption Scheme [19]

- **Setup the system parameters:**
  1. Two prime numbers p and q such that $q|(p-1)$.
  2. $g \in \mathbb{Z}_p^*$ has order q. Let $G = \langle g \rangle$ be a group of order q.
  3. A random hash function H that maps $\{0,1\}^*$ to $\mathbb{Z}_q$.
- **Key generation algorithm:** Pick a random number $x \in \mathbb{Z}_q^*$ as the private key. Compute the corresponding public key by $y = g^x$.
- **Encryption:** Given a message $m \in G$, pick random numbers $r, t \in \mathbb{Z}_q^*$ and compute $c = (c_1, c_2, v_1, v_2)$ as

$$\begin{aligned} c_1 &= g^r, \\ c_2 &= my^r, \\ v_1 &= H(c_1, c_2, g^t), \\ v_2 &= t + v_1 r \bmod q. \end{aligned}$$

  Then, the ciphertext c of m is (c_1, c_2, v_1, v_2). Note that r is the nonce used once.
- **Decryption:** Given (c_1, c_2, v_1, v_2), compute $v_1' = H(c_1, c_2, g^{v_2} c_1^{-v_1})$ and then test if $v_1' = v_1$. If that test succeeds, then output $m = c_2/c_1^x$; otherwise, terminate.

Schnorr and Jakobsson prove the following theorem [19].

Theorem 8. *Let the attacker A be given g, y, distinct messages m_0, m_1, a target ciphertext c_b corresponding to m_b for a random bit $b \in \{0,1\}$, and oracles for H and for decryption. Then a generic A using t generic steps cannot predict b with a better probability than $\frac{1}{2} + \frac{t^2}{q}$. The probability space consists of the random x, H, b, and the coin tosses r of the encipherer.*

Theorem 8 proves that the signed ElGamal encryption is IND-CCA2 secure in the random oracle model and the generic model.

We now show that the signed ElGamal encryption is NA1. The proof of Theorem 8 in [19, Theorem 1] shows that there exists a generic extractor A^* that extracts the nonce $r = \log_g c_1$ from a valid ciphertext produced by A. Moreover, given r, the plaintext m can be extracted in one generic step. Thus, the signed ElGamal encryption is — in a generic way — NA1. We have the following theorem.

Theorem 9. *The signed ElGamal encryption is NA1 in the random oracle model and the generic model.*

3.2 The Cramer-Shoup Encryption Scheme [6, 7]

Choose a large prime p such that $p - 1 = 2q$, where q is also prime. The group G is the subgroup of order q in $\mathbb{Z}_p^*$.

- **Key generation:** Pick two random elements $g_1, g_2 \in G$ and five random elements $x_1, x_2, y_1, y_2, z \in \mathbb{Z}_q^*$. Compute $c = g_1^{x_1} g_2^{x_2}$, $d = g_1^{y_1} g_2^{y_2}$ and $e = g_1^z$.

Pick a target collision-resistant hash function $H : G^3 \mapsto \mathbb{Z}_q$. The private key is (z, x_1, x_2, y_1, y_2) and the corresponding public key is (g_1, g_2, e, c, d, H).

- **Encryption:** Given a message $m \in G$, pick a random number $r \in \mathbb{Z}_q^*$ and compute (c_1, c_2, v_1, v_2) as

$$c_1 = g_1^r,$$
$$c_2 = me^r.$$
$$v_1 = g_2^r,$$
$$v_2 = c^r d^{rh}, \text{ where } h = H(c_1, c_2, v_1).$$

  The ciphertext c of m is (c_1, c_2, v_1, v_2). Note that r is the nonce used once.

- **Decryption:** Given (c_1, c_2, v_1, v_2), compute $h = H(c_1, c_2, v_1)$ and then test if

$$c_1^{x_1+y_1h} v_1^{x_2+y_2h} = v_2.$$

  If that test succeeds, then output $m = c_2/c_1^z$; otherwise, terminate.

A simple implementation to encode a message is also suggested in [6]: restrict a message to be an element of the set $\{1, \ldots, q\}$, and encode it by squaring it modulo p, giving us an element in G. A message can be recovered from its encoding by computing the unique square root of its encoding modulo p that is in the set $\{1, \ldots, q\}$.

Cramer and Shoup prove the following theorem [6, 7].

Theorem 10. *The above encryption scheme is IND-CCA2 assuming that (1) the hash function H is chosen from a target collision-resistant family, and (2) the DDH problem is hard in the group G.*

Now we show that the Cramer-Shoup encryption scheme is NA1. Dent proves that the scheme is simulatable under the assumptions that the DDH problem is hard in the group G and the hash function H is chosen from a target collision-resistant family [9, Section 4.1]. By the definition of a simulatable encryption scheme, there exit two polynomial-time algorithms (f, f^{-1}) such that random element r and $f^{-1}(f(r))$ are indistinguishable. Also $\mathcal{E}(pk, m)$ and $f(r)$ are indistinguishable. Hence, r and $f^{-1}(\mathcal{E}(pk, m))$ are indistinguishable. We can regard f^{-1} as the na-extractor, so the scheme is nonce-aware. Further, given f^{-1}, we can construct pa2-extractor by computing $m = c_2 e^{-f^{-1}(c)}$. This is because indistinguishability of r and $f^{-1}(c)$ implies that of c_2e^{-r} and $c_2e^{-f^{-1}(c)}$. Accordingly the Cramer-Shoup scheme is NA1. We have the following theorem.

Theorem 11. *The above encryption scheme is NA1 assuming that (1) the hash function H is chosen from a target collision-resistant family, and (2) the DDH problem is hard in the group G.*

4 Conclusions

This paper puts forth a new security notion, nonce-awareness, to capture the idea behind IND-CCA2 encryption schemes. We also show that two notable IND-CCA2 encryption schemes are NA1. These results point out another perspective on designing IND-CCA2 encryption schemes.

References:

[1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology—CRYPTO '98*, volume 1462, pages 26–45, 1998.

[2] M. Bellare and A. Palacio. Towards plaintext-aware public-key encryption without random oracles. In P. J. Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 48–62. Springer, 2004.

[3] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.

[4] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In A. D. Santis, editor, *Advances in Cryptology—EUROCRYPT 94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer-Verlag, 1995, 9–12 May 1994.

[5] M. Bellare and A. Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In M. J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pages 519–536. Springer, 1999.

[6] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *CRYPTO*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer, 1998.

[7] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

[8] I. B. Damgård. Payment systems and credential mechanisms with provable security against abuse by individuals. In *Advances in Cryptology—CRYPTO '88*, volume 403, pages 328–335. Springer-Verlag, 1990.

[9] A. W. Dent. The cramer-shoup encryption scheme is plaintext aware in the standard model. In S. Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 289–307. Springer, 2006.

[10] D. Dolev, C. Dwork, and M. Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

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

[12] Y.-D. Lyuu and M.-L. Wu. A fully public-key traitor-tracing scheme. *WSEAS Transactions on Circuits*, 1(1):88–93, 2002.

[13] Y.-D. Lyuu and M.-L. Wu. Attacks on a threshold proxy signature scheme based on the rsa cryptosystem. *WSEAS Transactions on Information Science and Applications*, 1:1041–1044, 2004.

[14] Y.-D. Lyuu and M.-L. Wu. Cryptanalysis of an elgamal-like cryptosystem for enciphering large messages. *WSEAS Transactions on Information Science and Applications*, 1:1079–1081, 2004.

[15] S. Micali, C. Rackoff, and B. Sloan. The notion of security for probabilistic cryptosystems. *SIAM Journal on Computing*, 17(2):412–426, Apr. 1988.

[16] M. Nao and M. Yung. Universal one-way hash function and their cryptographic application. In *Proceedings of the 21th Annual Symposium on Theory of Computing (STOC)*, pages 33–43. ACM Press, 1989.

[17] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attack. In *Proceedings of*

the 22nd Annual Symposium on Theory of Computing (STOC), pages 427–437. ACM, 1990.

[18] C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Advances in Cryptology—CRYPTO '91*, volume 576, pages 433–444. Springer-Verlag, 1992.

[19] C.-P. Schnorr and M. Jakobsson. Security of signed elgamal encryption. In *ASIACRYPT*, pages 73–89, 2000.

[20] V. Shoup. Lower bounds for discrete logarithms and related problems. In *Advances in Cryptology—EUROCRYPT '97*, volume 1233, pages 256–266, 1997.

[21] V. Shoup. Why chosen ciphertext security matters. Research Report RZ 3076 (#93122), IBM Research, Nov. 1998.