Stateful Key Encapsulation Mechanism

Peng Yang,[†1] Rui Zhang,[†2] Kanta Matsuura[†1]
and Hideki Imai[†2]

The concept of stateful encryption was introduced to reduce computation cost of conventional public key encryption schemes. Bellare et al. proposed one stateful encryption scheme in random oracle model which can save one exponentiation from two, and another scheme in the standard model which can save one exponentiation from three. To remove the gap assumption in Bellare's random oracle scheme, Yang et al. showed a trade-off between assumption and computation.

Above, all the schemes were built in the same manner: using an IND-CCA secure symmetric key encryption to achieve data privacy, and the symmetric key being provided by a key encapsulation. This means the provable security of all above schemes depends on the security of both symmetric key encryption and key encapsulation. In this paper, we first formalize the key encapsulation part, i.e., we propose a new primitive named stateful key encapsulation mechanism. Then, we show how to achieve stateful encryption by composing our primitive and symmetric encryption in a generic way.

1. Introduction

Public key encryption (PKE) is a very important tool for securing digital communicabilities. On the opposite of convenient key management functionalities, PKE schemes are often very slow compared with *symmetric encryption* (SE). In resource-constrained environment like mobile communications and sensor networks, this disadvantage of PKE will be quite undesirable, since system performance will drop greatly due to the high computational cost from frequent discrete modular exponentiations.

To improve the encryption performance of PKE, Bellare, Kohno and Shoup[4] introduced the concept of *stateful PKE* (SPKE) in ACM-CCS'06, where a sender maintains some state information. Without loss of generality, the state information is divided into two parts: the secret part and the public part. Then the encryption algorithm takes as input not only a message and the public key of receiver, but also his current secret state to produce a ciphertext. As a result, the sender's computational cost for encryption is dramatically reduced. Decryption performance remains unchanged from stateless scheme, and the receivers need not even necessarily to notice if the sender is stateful if the public state is included in the ciphertext. Note that no such state information is required for either the sender or the receiver in conventional public key encryption schemes.

Regarding the security notions, the standard chosen ciphertext security (CCA)[8],[11] is modified to adjust a single-sender-multiple-receiver network, which in turn implies security of more general settings. According to whether the adversary is required to know the secret keys of the players other than its target, the model is further classified into *known secret key* (KSK) and *unknown secret key* (USK) settings. Apparently, the USK model is stronger and seems more realistic. This paper mainly focuses on the USK security model.

1.1 Related works.

Bellare, Kohno and Shoup introduced the model of SPKE and proposed two constructions based on DHIES[1] and Kurosawa-Desdmet[9]. Yang, Zhang and Matsuura proposed variants of SPKE and SIBE schemes[14], trading assumptions/generality with computation costs. On the other hand, Baek, Zhou and Bao[2] proposed a "generic" construction, and demonstrated many efficient instantiations. We remark that the "generic" construction of 2) requires additionally that underlying *key encapsulation mechanism* (KEM)[13] meets two non-standard properties: "partitioned" and "reproducibility". Thus their approach is not necessarily a real simplification for scheme designing.

An *identity based encryption* (IBE) scheme is a special public key encryption scheme, where public keys can be arbitrary strings, introduced by Shamir[12] to simplify public key certificate management. The model of *stateful IBE* (SIBE) was first formalized by Phong, Matsuoka and Ogata[10], as the stateful counterpart of IBE. Yang et al.[15] introduced the concept of *stateful identity based key encapsulation mechanism* (SIBKEM), and showed how to employ this primitive

[†1] The University of Tokyo.
[†2] Research Center for Information Security, Advanced Industrial Science and Technology.
[⋆1] This work was supported by the Mitsubishi Foundation (Research Grants in the Natual Sciences).

to generically construct SIBE.

1.2 Our motivation and contributions.

This paper aims at the key encapsulation part of SPKE. We formalize this part as a cryptographic primitive named *stateful key encapsulation mechanism* (SKEM), which eventually enables a modular design approach for SPKE schemes, together with IND-CCA secure symmetric encryption. We formally give a composition theorem for such approach.

2. Preliminaries

In this section, we review the security models of stateful public key encryption, and symmetric encryption.

2.1 Conventions

NOTATIONS. Let $y \leftarrow A(x_1, ..., x_n)$ denote the experiment of assigning the result of A to y. If $\mathcal{S}$ is a finite set then let $x \leftarrow \mathcal{S}$ denote the operation of picking an element at random and uniformly from $\mathcal{S}$. If α is neither an algorithm nor a set then let $x \leftarrow \alpha$ denote a simple assignment statement. Denote PPT as a *probabilistic polynomial-time*.

NEGLIGIBLE FUNCTION. We say a function $\epsilon : \mathbb{N} \rightarrow \mathbb{R}$ is negligible if for every constant $c \geq 0$ there exits an integer k_c such that $\epsilon(k) < k^{-c}$ for all $k > k_c$.

2.2 Stateful Public Key Encryption

The first SPKE scheme was shown by Bellare, Kohno and Shoup[4]. Here, we review the model and then define the IND-CCA security in the USK model. Note that currently there is no SPKE scheme considering security in the CPA sense.

2.2.1 Algorithms

An SPKE scheme is specified by five algorithms. $\mathcal{SPKE} = \{$Setup, KeyGen, NwSt, Enc, Dec$\}$, where

Setup: The randomized setup algorithm takes as input security parameter 1^λ where $\lambda \in \mathbb{N}$. It outputs the system parameters *sp*. It also specifies the message space $\mathcal{M}$ by *sp*. ($\mathcal{M}$ may be included in *sp*.) We write $sp \leftarrow \mathsf{Setup}(1^\lambda)$.

KeyGen: The (possibly randomized) key generation algorithm takes as input *sp*. It outputs a key pair (pk, sk), where *pk* is a public key and *sk* is the corresponding secret key of *pk*. *pk* will be published to every participant in the system, while *sk* will be securely sent to its owner. We write $(pk, sk) \leftarrow \mathsf{KeyGen}(sp)$.

NwSt: The randomized new state algorithm takes as input *sp*. It outputs a new state *st* of a sender. We write $st \leftarrow \mathsf{NwSt}(sp)$.

Enc: The randomized encryption algorithm computes the corresponding ciphertext *c* of a plaintext *m* on *sp*, *pk* and *st*, where *pk* is the receiver's public key. We write $c \leftarrow \mathsf{Enc}(sp, pk, st, m)$.

Dec: The deterministic decryption algorithm recovers the plaintext *m* from the a ciphertext *c* on *sp* and *sk*. We write $m \leftarrow \mathsf{Dec}(sp, sk, c)$.

2.2.2 Security

We establish the IND-CCA (indistinguishability against adaptive chosen ciphertext attack) game for SPKE between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. In this game, the PPT adversary $\mathcal{A}$ tries to distinguish which plaintext was encrypted. The game is described as follows.

Setup: $\mathcal{C}$ takes the security parameter λ and runs Setup of SPKE. It then runs KeyGen to obtain a key pair (pk_1, sk_1) as the target. It passes the the resulting system parameters *sp* and the target public key pk_1 to $\mathcal{A}$ and keeps the secret key sk_1 as secret. $\mathcal{C}$ also sends all of the other secret keys $\{sk_2, \cdots, sk_n\}$ in the system to $\mathcal{A}$, where $sk_i \neq sk_1$. This captures the fact that $\mathcal{A}$ may corrupt all the entities other than his attack target. The state *st* is decided a-priori by $\mathcal{C}$.

Phase 1: $\mathcal{A}$ issues two types of queries $q_1, \cdots, q_i$ where a query is one of

⋄ Encryption queries on a public key and a message (pk_i, m), where $1 \leq i \leq n$. $\mathcal{C}$ responds with ciphertext *c* of *m* under public key pk_i and the current state *st*.

⋄ Decryption queries on a ciphertext *c*. $\mathcal{C}$ responds with the plaintext *m* of *c*, which is encrypted under the target public key pk_1.

These queries may be asked adaptively, that is, each query q_i may depends on the replies to $q_1, \cdots, q_{i-1}$.

Challenge: Once $\mathcal{A}$ decides that phase 1 is over, he outputs two equal length plaintext m_0, m_1. Then $\mathcal{C}$ flips a coin $b \in \{0,1\}$ and sets $c^* \leftarrow \mathsf{Enc}(sp, pk_1, st, m_b)$. $\mathcal{C}$ returns c^* to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ issues more queries $q_{i+1}, \cdots, q_j$ where a query is one of

$\diamond$ Encryption queries on a public key and a message (pk_i, m). $\mathcal{C}$ responds as in phase 1.

$\diamond$ Decryption queries on a ciphertext $c \neq c^*$. $\mathcal{C}$ responds as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

We refer to such an adversary $\mathcal{A}$ as an $\mathsf{IND\text{-}CCA}$ adversary. $\mathcal{A}$'s advantage in this $\mathsf{IND\text{-}CCA}$ game is defined to be $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[b = b'] - 1/2|$. We say that an SPKE scheme is secure in the sense of $\mathsf{IND\text{-}CCA}$ if the advantage is negligible for any PPT algorithm $\mathcal{A}$.

2.3 Symmetric Encryption

Here, we simply review the definition and security requirements of symmetric encryption (SE).

An SE scheme consists of three algorithms, $\mathcal{SE} = (\mathsf{K}, \mathsf{E}, \mathsf{D})$. The randomized key generation algorithm K takes as input the security parameter λ and outputs a session key dk. We write $dk \leftarrow \mathsf{K}(\lambda)$. The (possibly randomized) encryption algorithm E takes as input a session key dk and a plaintext m and computes a ciphertext C. We write $C \leftarrow \mathsf{E}(dk, m)$. The decryption algorithm D takes as input a session key dk and a ciphertext C and outputs a plaintext m (or "$\perp$" for invalid). We write $m/\perp \leftarrow \mathsf{D}(dk, C)$. The standard consistency constraint is that $\forall dk : m \leftarrow \mathsf{D}(dk, \mathsf{E}(dk, m))$.

Symmetric encryption scheme must guarantee indistinguishability against chosen ciphertext attack. We establish an $\mathsf{IND\text{-}CCA}$ game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The game is described as follows.

Setup: $\mathcal{C}$ takes the security parameter λ, runs K to obtain a random key dk, and flips a coin $b \leftarrow \{0, 1\}$.

Query: $\mathcal{A}$ issues two types of queries $q_1, \cdots, q_i$ where a query is one of

$\diamond$ Left-or-right queries on two messages (m_0, m_1). $\mathcal{C}$ responds with ciphertext $C \leftarrow \mathsf{E}(dk, m_b)$.

$\diamond$ Decrypt-or-reject queries on a ciphertext C. If $b = 1$, then $\mathcal{C}$ responds with the message $m \leftarrow \mathsf{D}(dk, C)$; otherwise $\mathcal{C}$ responds with $\perp$. The restriction is that C must be different from the output from left-or-right queries.

Guess: Finally, $\mathcal{A}$ outputs a bit $b' \in \{0, 1\}$.

$\mathcal{A}$'s advantage in this $\mathsf{IND\text{-}CCA}$ game is defined to be $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[b = b'] - 1/2|$. We say that an SE scheme is secure if the advantage is negligible for any PPT algorithm $\mathcal{A}$. In this paper, we require SE to be multiple time secure, and such SE schemes can be generically built from standard block ciphers and message authentication codes (MAC)[3].

3. Stateful Key Encapsulation Mechanism

In this section, we introduce the model and security notions of SKEM. Roughly speaking, SKEM is the "stateful version" of *conventional key encapsulation mechanism* (KEM). In particular, in SKEM, the sender maintains a state information. And for a specified public key, the session key encapsulated by the sender remains the same unless the state is updated. Since it is deterministic, SKEM seems to being capturing different security aspect from KEM, i.e., the adversary can issue neither encapsulation query nor decapsulation query on the target public key.

3.1 Algorithms

A SKEM scheme is specified by five algorithms. $\mathcal{SKEM} = \{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{NwSt}, \mathsf{Enc}, \mathsf{Dec}\}$.

Setup: The randomized setup algorithm takes as input security parameter 1^λ where $\lambda \in \mathbb{N}$. It outputs the system parameters sp which will be announced to all party involved in the system. It also specifies the key space $\mathcal{SHK}$ by sp. ($\mathcal{SHK}$ may be included in sp.) We write $sp \leftarrow \mathsf{Setup}(1^\lambda)$.

KeyGen: The randomized key generation algorithm takes as input sp. It outputs a key pair (pk, sk), where pk is a public key, and sk is the corresponding secret key.

NwSt: The randomized new state algorithm takes as input sp. It outputs a new state st of a sender. We write $st \leftarrow \mathsf{NwSt}(sp)$.

Enc: The deterministic encapsulation algorithm takes as input sp, pk and st, where pk is the receiver's public key. It outputs the corresponding ciphertext c of a session key dk. We write $(c, dk) \leftarrow \mathsf{Enc}(sp, pk, st)$.

Dec: The deterministic decapsulation algorithm takes as sp, sk and a ciphertext c. It outputs the session key dk. We write $dk \leftarrow \mathsf{Dec}(sp, sk, c)$.

3.2 IND-CCA Security

We establish the $\mathsf{IND\text{-}CCA}$ (indistinguishability against adaptive chosen ciphertext attack) game for SKEM between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. In this game, the PPT adversary $\mathcal{A}$ tries to distinguish if $\mathcal{C}$ gives him a valid session key or a random key. The game is described as follows.

Setup: $\mathcal{C}$ takes the security parameter λ and runs Setup of SPKE. It then runs KeyGen to obtain a key pair (pk_1, sk_1) as the target. It passes the the resulting system parameters sp and the target public key pk_1 to $\mathcal{A}$ and keeps the secret key sk_1 as secret. $\mathcal{C}$ also sends all of the other secret keys $\{sk_2, \cdots, sk_n\}$ in the system to $\mathcal{A}$, where $sk_i \neq sk_1$. This captures the fact that $\mathcal{A}$ may corrupt all the entities other than his attack target. The state st is decided a-priori by $\mathcal{C}$.

Phase 1: $\mathcal{A}$ issues two types of queries $q_1, \cdots, q_i$ where a query is one of

$\diamond$ Encapsulation queries on a public key pk_i, where $1 \leq i \leq n$. $\mathcal{C}$ responds with ciphertext c and a decryption key dk under id and the current state st.

$\diamond$ Decapsulation queries on a ciphertext c. $\mathcal{C}$ responds with the decryption key dk of c, which is encapsulated under the target public key pk_1.

These queries may be asked adaptively, that is, each query q_i may depends on the replies to $q_1, \cdots, q_{i-1}$.

Challenge: Once $\mathcal{A}$ decides that phase 1 is over, $\mathcal{C}$ computes a valid key-ciphertext pair (c^*, dk_1^*) and flips a coin $b \in \{0,1\}$. If $b = 0$, then $\mathcal{C}$ chooses a random key dk_0^* from the key space and returns (c^*, dk_0^*) to $\mathcal{A}$; otherwise $\mathcal{C}$ returns (c^*, dk_1^*).

Phase 2: $\mathcal{A}$ issues more queries $q_{i+1}, \cdots, q_j$ where a query is one of

$\diamond$ Encapsulation queries on a public key pk_i. $\mathcal{C}$ responds as in phase 1.

$\diamond$ Decapsulation queries on a ciphertext $c \neq c^*$). $\mathcal{C}$ responds as in phase 1.

Guess: Finally, $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$.

We refer to such an adversary $\mathcal{A}$ as an $\mathsf{IND\text{-}CCA}$ adversary. $\mathcal{A}$'s advantage in this $\mathsf{IND\text{-}CCA}$ game is defined to be $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[b = b'] - 1/2|$. We say that an SKEM scheme is secure in the sense of $\mathsf{IND\text{-}CCA}$ if the advantage is negligible for any PPT algorithm $\mathcal{A}$.

4. Composition Theorem

By combining an $\mathsf{IND\text{-}CCA}$ secure $\mathcal{SKEM} = \{\mathsf{SKEM.Setup}, \mathsf{SKEM.KeyGen}, \mathsf{SKEM.NwSt}, \mathsf{SKEM.Enc}, \mathsf{SKEM.Dec}\}$ and an $\mathsf{IND\text{-}CCA}$ secure $\mathcal{SE} = \{\mathsf{SE.K}, \mathsf{SE.E}, \mathsf{SE.D}\}$, we can obtain an $\mathsf{IND\text{-}CCA}$ secure $\mathcal{SPKE} = \{\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{NwSt}, \mathsf{Enc}, \mathsf{Dec}\}$. We omit composition details since it is straightforward. At a high level, the SPKE sender uses $\mathsf{SE.E}$ to encrypt a message by using the key dk encapsulated by $\mathsf{SKEM.Enc}$, and the SPKE receiver runs $\mathsf{SE.D}$ to decrypt with dk recovered by $\mathsf{SKEM.Dec}$.

Theorem 1 Suppose $\mathcal{SKEM}$ is $\mathsf{IND\text{-}CCA}$ secure, and $\mathcal{SE}$ is $\mathsf{IND\text{-}CCA}$ secure. Then the hybrid encryption scheme $\mathcal{SPKE}$ is $\mathsf{IND\text{-}CCA}$ secure.

Proof. We employ the game-based proof technique.

Game 0. Fix an efficient adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. We define Game 0 to be the attack game by $\mathcal{A}$ in the definition of $\mathsf{IND\text{-}CCA}$ for SPKE. For proof convenience, we describe Game 0 as follows.

$$\begin{aligned}
&sp \leftarrow \mathsf{Setup}(1^\lambda);\\
&(pk_1, sk_1) \leftarrow \mathsf{KeyGen}(sp); \cdots; (pk_n, sk_n) \leftarrow \mathsf{KeyGen}(sp);\\
&st \leftarrow \mathsf{NwSt}(sp);\\
&(m_0, m_1) \leftarrow \mathcal{A}_1^{\mathcal{O}}(sp, sk_2, \cdots, sk_n);\\
&b \leftarrow \{0,1\};\\
&(c^*, dk_1^*) \leftarrow \mathsf{Enc}(sp, st, pk_1);\\
&C^* \leftarrow \mathsf{E}(dk_1^*, m_b);\\
&b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(sp, sk_2, \cdots, sk_n, c^*, C^*)
\end{aligned}$$

In the above, we define E_0 to be the event that $b' = b$. Thus $\mathcal{A}$'s advantage is $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[E_0] - 1/2|$.

Game 1. The difference from Game 0 is that instead of encrypt m_b with dk_1^*, we encrypt it with randomly chosen $dk_0^* \in \mathcal{SHK}$. We describe Game 1 as follows. The box shows the difference.

```
$sp \leftarrow \mathsf{Setup}(1^\lambda);$
$(pk_1, sk_1) \leftarrow \mathsf{KeyGen}(sp); \cdots; (pk_n, sk_n) \leftarrow \mathsf{KeyGen}(sp);$
$st \leftarrow \mathsf{NwSt}(sp);$
$(m_0, m_1) \leftarrow \mathcal{A}_1^{\mathcal{O}}(sp, sk_2, \cdots, sk_n);$
$b \leftarrow \{0, 1\};$
$(c^*, dk_1^*) \leftarrow \mathsf{Enc}(sp, st, pk_1);$
$\boxed{dk_0^* \leftarrow \mathcal{SHK}; C^* \leftarrow \mathsf{E}(dk_0^*, m_b);}$
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(sp, sk_2, \cdots, sk_n, c^*, C^*)$
```

Let E_1 be the event that $b' = b$ in Game 1.

Claim 1. $|\Pr[E_1] - 1/2| = \mathbf{Adv}_{\mathcal{B}_1}(\lambda)$. Here $\mathbf{Adv}_{\mathcal{B}_1}(\lambda)$ is the advantage of an adversary against $\mathcal{SE}$, and this advantage is assumed to be negligible. This follows from the fact that in Game 1, the encryption key dk_0^* is completely randomly distributed in $\mathcal{SHK}$.

Claim 2. $|\Pr[E_0] - \Pr[E_1]| = \mathbf{Adv}_{\mathcal{B}_2}(\lambda)$. Here $\mathbf{Adv}_{\mathcal{B}_2}(\lambda)$ is the advantage of an adversary against $\mathcal{SKEM}$, and this advantage is assumed to be negligible. The proof of Claim 2 is essentially the observation that in Game 0, the pair (c^*, dk_1^*) is real output from encapsulation algorithm, while in Game 1, a random dk_0^* is given instead. In this case, $\mathcal{A}$ should not notice the difference under the assumption that $\mathcal{SKEM}$ is secure. Rigorously, we construct a distinguishing algorithm $\mathcal{B}_2$ as follows.

```
Distinguisher  $\mathcal{B}_2(c^*, dk^*)$
    $sp \leftarrow \mathsf{SKEM.Setup}(1^\lambda);$
    $st \leftarrow \mathsf{SKEM.NwSt}(sp);$
    $(m_0, m_1) \leftarrow \mathcal{A}_1^{\mathcal{O}}(sp, sk_2, \cdots, sk_n);$
    $b \leftarrow \{0, 1\}; C^* \leftarrow \mathsf{E}(dk^*, m_b);$
    $b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(sp, sk_2, \cdots, sk_n, c^*, C^*);$
    if  $b' = b$  then output  1  else output  0
```

It is obvious that $\mathcal{B}_2$ interpolates between Game 0 and Game 1. If the input of $\mathcal{B}_2$ is the real output from encapsulation algorithm, then it works as same as Game 0. If the input of $\mathcal{B}_2$ is a ciphertext and a random key, then it works as same as Game 1.

Thus, the advantage of $\mathcal{B}_2$ against $\mathcal{SKEM}$ is equal to $|\Pr[E_0] - \Pr[E_1]|$. This completes the proof of Claim 2.

Combining Claim 1 and Claim 2, we have that $\mathbf{Adv}_{\mathcal{A}}(\lambda) = \mathbf{Adv}_{\mathcal{B}_1}(\lambda) + \mathbf{Adv}_{\mathcal{B}_2}(\lambda)$. Since $\mathcal{SKEM}$ and $\mathcal{SE}$ are secure, thus $\mathcal{A}$' advantage $\mathbf{Adv}_{\mathcal{A}}(\lambda)$ against $\mathcal{SPKE}$ is negligible. This completes the proof of Theorem 1. □

5. Conclusions

This paper introduced a cryptographic primitive named stateful key encapsulation mechanism. We also discussed how to achieve a stateful public key encryption scheme by composing this primitive and an IND-CCA secure symmetric key encryption.

References

1) Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The oracle Diffie-Hellman assumptions and an analysis of DHIES. In *CT-RSA'01*, volume 2020 of *LNCS*, pages 143–158. Springer, 2001.

2) Joonsang Baek, Jianying Zhou, and Feng Bao. Generic constructions of stateful public key encryption and their applications. In *ACNS'08*, volume 5037 of *Lecture Notes in Computer Science*, pages 75–93, 2008.

3) M.Bellare and C.Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *ASIACRYPT'00*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545, 2000. Full version appeared in[5].

4) Mihir Bellare, Tadayoshi Kohno, and Victor Shoup. Stateful public-key cryptosystems: how to encrypt with one 160-bit exponentiation. In *ACM CCS'06*, pages 380–389. ACM, 2006.

5) Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *J. Cryptology*, 21(4):469–491, 2008. Preliminary version appeared in[3].

6) D.Boneh and M.Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003. Full version of[7].

7) Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil pairing. In *Advances in Cryptology - CRYPTO '01*, volume 2139 of *LNCS*, pages 213–229. Springer, 2001. Extended abstract of[6].

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

9) Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO '04*, volume 3152 of *LNCS*, pages 426–442. Springer, 2004.

10) Le Trieu Phong, Hiroto Matsuoka, and Wakaha Ogata. Stateful identity-based

encryption scheme: Faster encryption and decryption. In *ASIACCS'08*, pages 381–388. ACM, 2008.

11) 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 of *LNCS*, pages 433–444. Springer, 1991.

12) A.Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology - CRYPTO '84*, volume 196 of *LNCS*, pages 47–53. Springer, 1984.

13) Victor Shoup. A standard for public-key encryption. ISO 18033-2, 2006.

14) P.Yang, R.Zhang, and K.Matsuura. Stateful public key encryption: How to remove gap assumptions and maintaining tight reductions. In *The 2008 International Symposium on Information Theory and its Applications (ISITA '08)*, 2008.

15) P.Yang, R.Zhang, K.Matsuura, and H.Imai. Generic construction of stateful identity based encryption. In *SCIS 2009*, 2009.