3-2010

Efficient CCA-Secure PKE from Identity-Based Techniques

Junzuo LAI
Shanghai Jiaotong University

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Shengli LIU
Shanghai Jiaotong University

Weidong KOU
Xidian University

DOI: https://doi.org/10.1007/978-3-642-11925-5_10

Efficient CCA-Secure PKE from Identity-Based Techniques

Junzuo Lai[1], Robert H. Deng[2], Shengli Liu[1], and Weidong Kou[3]

[1] Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai 200030, China
{laijunzuo,slliu}@sjtu.edu.cn
[2] School of Information Systems,
Singapore Management University, Singapore 178902
robertdeng@smu.edu.sg
[3] School of Computer Science and Technology
Xi Dian University, Xi'an 710071, China
kou_weidong@yahoo.com.cn

Abstract. Boneh, Canetti, Halevi, and Katz showed a general method for constructing CCA-secure public key encryption (PKE) from any selective-ID CPA-secure identity-based encryption (IBE) schemes. Their approach treated IBE as a black box. Subsequently, Boyen, Mei, and Waters demonstrated how to build a direct CCA-secure PKE scheme from the Waters IBE scheme, which is adaptive-ID CPA secure. They made direct use of the underlying IBE structure, and required no cryptographic primitive other than the IBE scheme itself. However, their scheme requires long public key and the security reduction is loose. In this paper, we propose an efficient PKE scheme employing identity-based techniques. Our scheme requires short public key and is proven CCA-secure in the standard model (without random oracles) with a tight security reduction, under the Decisional Bilinear Diffie-Hellman (DBDH) assumption. In addition, we show how to use our scheme to construct an efficient threshold public key encryption scheme and a public key encryption with non-interactive opening (PKENO) scheme.

Keywords: Chosen Ciphertext Security, Public Key Encryption, Identity-Based Encryption.

1 Introduction

Chosen-ciphertext security (CCA-security, for short) [31,16] is now considered as a standard notion of security for public key encryption (PKE) in practice. There have been several efficient PKE schemes shown to be secure in the random oracle (RO) model [3]. Unfortunately, the RO model is heuristic, and a proof of security in the RO model does not directly imply anything about the security of a PKE scheme in the real world. In fact, it has been demonstrated that there exist cryptographic schemes which are secure in the RO model but which are inherently insecure when the random oracle is instantiated with any real hash

J. Pieprzyk (Ed.): CT-RSA 2010, LNCS 5985, pp. 132–147, 2010.

function [8,29,19,2]. Throughout this paper, we focus on PKE schemes whose security are proven in the standard model (without random oracles).

Dolev, Dwork, and Naor [16] were the first to come up with a CCA-secure PKE scheme in the standard model. Later Cramer and Shoup [11] proposed the first practical CCA-secure PKE scheme in the standard model, under the Decisional Diffie-Hellman (DDH) assumption. Interestingly, Elkind and Sahai [17] showed that both techniques can be viewed as special cases of a single paradigm.

Canetti, Halevi, and Katz [9] presented a new paradigm for constructing CCA-secure PKE schemes using IBE as a building block. The idea is to use, for each encryption, a fresh random verification key of a one-time signature scheme as the "identity" for IBE encryption. In order to tie the IBE ciphertext to this verification key, the ciphertext is signed using the corresponding signing key. If the IBE scheme is selective-ID CPA secure then the resulting PKE scheme is CCA secure. Boneh and Katz [6] further improved the efficiency of the scheme by using a MAC instead of a one-time signature. Kiltz [25] showed that a tag-based encryption (TBE) scheme is sufficient for the transformation in [9] to obtain a CCA-secure PKE scheme.

Boyen, Mei, and Waters [7] showed how to build a direct CCA-secure PKE scheme from the Waters IBE scheme [33]. Unlike the Canetti-Halevi-Katz (CHK) scheme [9] and the Boneh-Katz (BK) scheme [6] that use IBE as a black box, their approach is endogenous, very simple, and compact. They constructed a CCA-secure PKE scheme, referred to as the BMW scheme, in which a ciphertext consists of just three group elements without attached signature or MAC. Compared with the CHK scheme and the BK scheme, the main difference is to use the first two elements of the ciphertext to determine a one-time "identity", instead of a fresh random "identity" generated by a one-time signature as in the CHK scheme or encapsulation as in the BK scheme. When proving security of the scheme, they took advantage of the CPA security of the Waters IBE scheme in the adaptive-ID security model (as opposed to the weaker selective-ID model). The drawback of the BMW scheme, however, is that the user needs *long* public key and the security is reduced only *loosely* to the Decisional Bilinear Diffie-Hellman (DBDH) assumption. Note that, an inefficient security reduction would imply either a lower security level or the requirement of larger key and ciphertext sizes to obtain the same security level.

1.1 Hybrid Encryption

Cramer and Shoup [12,32] formalized the notion of *hybrid encryption*, where a public key cryptosystem is used to encapsulate the (session) key of a symmetric cipher which is subsequently used to conceal the data. This is also known as the KEM/DEM approach. A folklore composition theorem (formalized in [12]) shows that if both KEM and DEM are CCA-secure then the hybrid encryption is CCA-secure. Kurosawa and Desmedt [27] came up with a hybrid encryption scheme improving the performance of the Cramer-Shoup scheme both in computational efficiency and in ciphertext length. Abe, Gennaro, Kurosawa and Desmedt [1] established the Tag-KEM/DEM framework, and explained the

security of Kurosawa-Desmedt scheme in this framework. Hofheinz and Kiltz [21] presented another paradigm for constructing hybrid encryption with strictly weakened KEM. The DDH assumption still is required for these extensions except for one of Hofheinz and Kiltz's schemes which depends on the *n-linear* DDH assumption.

Kiltz [26] presented a practical CCA-secure KEM scheme whose security is proven under the gap hashed Diffie-Hellman (GHDH) assumption. Cash, Kiltz and Shoup [10] proposed CCA-secure hybrid encryption schemes under the computational Diffie-Hellman (CDH) or hased Diffie-Hellman (HDH) assumption by using the twin DH problem (which is also applicable to a wide range of cryptographic primitives). Note that the CDH and HDH assumptions are weaker than the DDH assumption. Based on Naor-Pinkas broadcast encryption (BE) scheme [28], Hanaoka and Kurosawa [20] proposed more efficient CCA-secure hybrid encryption schemes under the CDH or HDH assumption. Recently, Hofheinz and Kiltz [22] proposed a practical CCA-secure hybrid encryption scheme whose security can be reduced to the assumption that factoring is intractable. However, all these hybrid encryption schemes are not suited for constructing threshold public key encryption and public key encryption with non-interactive opening schemes.

1.2 Our Contribution

In this paper, we propose a more efficient PKE scheme employing identity-based techniques. The proposed scheme has small public key size and is proven CCA-secure in the standard model with a tight security reduction, under the DBDH assumption. We follow a similar method in the proof simulation as that in the CHK, BK and BMW schemes. After the step phase there is a certain set of well-formed ciphertexts that the simulator can decrypt corresponding to "identities" that the simulator knows the private keys. The remainder of the well-formed ciphertexts, that the simulator cannot decrypt corresponding to "identities" for which the simulator does not know the private keys, can be used as challenge ciphertexts in the simulation.

Our scheme has the desirable property that it allows the validity of ciphertexts to be checked publicly. Using this property, we extend our scheme to an efficient threshold public key encryption scheme and an efficient PKE with non-interactive opening (PKENO) scheme. An overview comparing the efficiency of our PKE scheme to those of other PKE schemes employing identity-based techniques is given in Table 1.

1.3 Organization

The rest of the paper is organized as follows. In Section 2, we present some definitions and a related complexity assumption. We describe and analysis our PKE scheme in Section 3. In Section 4, we introduce two extensions of practical interest to our PKE scheme. Finally, we state our conclusion in Section 5.

Table 1. Comparison of public key encryption schemes employing identity-based techniques. "exp" denotes an exponentiation operation. (Some of the exponentiations are actually multi-exponentiations.) "pr" denotes a pairing operation. The CHK [9] and BK [6] schemes are instantiated with the first Boneh-Boyen IBE scheme from [4]. (Kiltz [24] showed that the CHK transformation maps the first and second Boneh-Boyen IBE schemes from [4] to nearly one single encryption scheme.)

Scheme	PK size	Encryption	Decryption	Ciphertext size	TPKE	PKENO
CHK[9]	$3\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	3 exp + Sig	1 exp + 1 pr + Ver	$2\|\mathbb{G}\|+1\|\mathbb{G}_T\|$+vk+sig	$\surd$	$\surd$
BK[6]	$3\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	3 exp	1 exp + 1 pr	$2\|\mathbb{G}\|+1\|\mathbb{G}_T\|$+com+tag	$\times$	$\times$
BMW[7]	$162\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	3 exp	1 exp + 1 pr	$2\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$\surd$	$\surd$
Ours	$4\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	3 exp	1 exp + 1 pr	$2\|\mathbb{G}\|+1\|\mathbb{G}_T\|+1\|\mathbb{Z}_p\|$	$\surd$	$\surd$

2 Preliminaries

For a group $\mathbb{G}$, we denote the size of a group-element representation as $|\mathbb{G}|$. We say that a function $f(\lambda)$ is *negligible* if for every $c > 0$ there exists an λ_c such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_c$.

2.1 Bilinear Pairings

Let $\mathbb{G}$ be a cyclic multiplicative group of prime order p and $\mathbb{G}_T$ be a cyclic multiplicative group of the same order p. A bilinear pairing is a map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ with the following properties:

- Bilinearity: $\forall g_1, g_2 \in \mathbb{G}, \forall a, b \in \mathbb{Z}_p^*$, we have $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$;
- Non-degeneracy: There exist $g_1, g_2 \in \mathbb{G}$ such that $e(g_1, g_2) \neq 1$;
- Computability: There exists an efficient algorithm to compute $e(g_1, g_2)$ for $\forall g_1, g_2 \in \mathbb{G}$.

2.2 Complexity Assumption

Definition 1 (DBDH Problem). *Given a group $\mathbb{G}$ of prime order p with generator g and elements $g^a, g^b, g^c \in \mathbb{G}$, $e(g,g)^z \in \mathbb{G}_T$ where a, b, c, z are selected uniformly at random from $\mathbb{Z}_p^*$. A fair binary coin $\beta \in \{0, 1\}$ is flipped. If $\beta = 1$, it outputs the tuple $(g, g^a, g^b, g^c, T = e(g,g)^{abc})$. If $\beta = 0$, it outputs the tuple $(g, g^a, g^b, g^c, T = e(g,g)^z)$. The Decisional Bilinear Diffie-Hellman (DBDH) problem is to guess the value of β.*

An adversary $\mathcal{A}$ has at least an ϵ advantage in solving the DBDH problem if

$$|\mathsf{Pr}[\mathcal{A}(g^a, g^b, g^c, T = e(g,g)^{abc}) = 1] - \mathsf{Pr}[\mathcal{A}(g^a, g^b, g^c, T = e(g,g)^z) = 1]| \geq \epsilon$$

where the probability is over the randomly chosen a, b, c, z and the random bits consumed by $\mathcal{A}$. We refer to the distribution on the left as $\mathcal{P}_{BDH}$ and the one on the right as $\mathcal{R}_{BDH}$.

Definition 2 (DBDH assumption). *We say that the* (ϵ, t)*-DBDH assumption holds in a group* $\mathbb{G}$ *if no algorithm running in time at most* t *can solve the DBDH problem in* $\mathbb{G}$ *with advantage at least* ϵ*.*

2.3 Collision-Resistant Hashing

Formally, we say that a function $H : X \to Y$ is a target-collision resistant (CR) hash function, if for all PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{CR}}_{\mathcal{A}}(\lambda)$ is negligible in λ, where $\mathsf{Adv}^{\mathsf{CR}}_{\mathcal{A}}(\lambda) = \mathsf{Pr}[x, x' \leftarrow \mathcal{A}(H) : x' \neq x \wedge H(x') = H(x)]$.

2.4 Public Key Encryption

A public key encryption scheme is a tuple of algorithms described as follows:

$\mathsf{KeyGen}(\lambda)$. Takes as input a security parameter λ. It outputs a public/private key pair (PK, SK).

$\mathsf{Encrypt}(\mathsf{PK}, m)$. Takes as input a public key PK and a message m. It outputs a ciphertext.

$\mathsf{Decrypt}(\mathsf{SK}, C)$. Takes as input a private key SK and a ciphertext C. It outputs a plaintext message or the special symbol $\perp$ meaning that the ciphertext is invalid.

We insist that all public key encryption schemes satisfy the obvious correctness condition (that decryption "undoes" encryption).

The strongest and commonly accepted notion of security for a public key encryption scheme is that of indistinguishability against an adaptive chosen ciphertext attack ($\mathsf{IND\text{-}CCA}$). It is defined using the following game between an attack algorithm $\mathcal{A}$ and a challenger.

Setup. The challenger runs $\mathsf{KeyGen}(\lambda)$ to obtain a public/private key pair (PK, SK). It gives the public key PK to the adversary.

Query phase 1. The adversary $\mathcal{A}$ adaptively issues decryption queries C. The challenger responds with $\mathsf{Decrypt}(\mathsf{SK}, C)$.

Challenge. The adversary $\mathcal{A}$ submits two (equal length) messages m_0, m_1. The challenger selects a random bit $\beta \in \{0, 1\}$, sets $C^* = \mathsf{Encrypt}(\mathsf{PK}, m_\beta)$ and sends C^* to the adversary as its challenge ciphertext.

Query phase 2. The adversary continues to adaptively issue decryption queries C, as in Query phase 1, but with the natural constraint that the adversary does not request the decryption of C^*.

Guess. The adversary $\mathcal{A}$ outputs its guess $\beta' \in \{0, 1\}$ for β and wins the game if $\beta = \beta'$.

We define $\mathcal{A}$'s advantage in attacking the public key encryption scheme PKE with the security parameter λ as $\mathsf{Adv}^{\mathsf{PKE}}_{\mathcal{A}}(\lambda) = |\mathsf{Pr}[\beta = \beta'] - \frac{1}{2}|$.

Definition 3. *We say that a public key encryption scheme* PKE *is* (t, q, ϵ)*-*$\mathsf{IND\text{-}CCA}$ *secure, if for all* t*-time algorithms* $\mathcal{A}$ *making at most* q *decryption queries have advantage at most* ϵ *in winning the above game.*

2.5 Public Key Encryption with Non-interactive Opening

A public key encryption with non-interactive opening (PKENO) scheme is a tuple of algorithms described as follows:

- $\mathsf{KeyGen}(\lambda)$. Takes as input a security parameter λ. It outputs a public/private key pair (PK, SK).
- $\mathsf{Encrypt}(\mathsf{PK}, m)$. Takes as input a public key PK and a message m. It outputs a ciphertext.
- $\mathsf{Decrypt}(\mathsf{SK}, C)$. Takes as input a private key SK and a ciphertext C. It outputs a plaintext message or the special symbol $\perp$ meaning that the ciphertext is invalid.
- $\mathsf{Prove}(\mathsf{SK}, C)$. Takes as input a private key SK and a ciphertext C. It outputs a proof π or the special symbol $\perp$ meaning that the ciphertext is invalid.
- $\mathsf{Ver}(\mathsf{PK}, C, m, \pi)$. Takes as input a public key PK, a ciphertext C, a message m and a proof π. It outputs a result $res \in \{0, 1\}$ meaning accepted and rejected proof, respectively. In particular $1 \leftarrow \mathsf{Ver}(\mathsf{PK}, C, \perp, \pi)$ must be interpreted as the verifier being convinced that C is an invalid ciphertext.

We insist that all PKENO schemes satisfy the obvious correctness condition (that decryption "undoes" encryption). In addition, we require, for a honestly generated key pair (PK, SK) and all ciphertexts C, $1 \leftarrow \mathsf{Ver}(\mathsf{PK}, C, \mathsf{Decrypt}(\mathsf{SK}, C), \mathsf{Prove}(\mathsf{SK}, C))$.

The notion of security for a PKENO scheme is indistinguishability against chosen-ciphertext and prove attacks ($\mathsf{IND\text{-}CCPA}$) and satisfies computational proof soundness. $\mathsf{IND\text{-}CCPA}$ is defined using the following game between an attack algorithm $\mathcal{A}$ and a challenger.

Setup. The challenger runs $\mathsf{KeyGen}(\lambda)$ to obtain a public/private key pair (PK, SK). It gives the public key PK to the adversary.

Query phase 1. The adversary $\mathcal{A}$ adaptively issues decryption or proof queries on C. The challenger responds with $\mathsf{Decrypt}(\mathsf{SK}, C)$ or $\mathsf{Prove}(\mathsf{SK}, C)$.

Challenge. The adversary $\mathcal{A}$ submits two (equal length) messages m_0, m_1. The challenger selects a random bit $\beta \in \{0, 1\}$, sets $C^* = \mathsf{Encrypt}(\mathsf{PK}, m_\beta)$ and sends C^* to the adversary as its challenge ciphertext.

Query phase 2. The adversary continues to adaptively issue decryption or proof queries C, as in Query phase 1, but with the natural constraint that decryption or proof queries on C^* are not allowed.

Guess. The adversary $\mathcal{A}$ outputs its guess $\beta' \in \{0, 1\}$ for b and wins the game if $\beta = \beta'$.

We define $\mathcal{A}$'s advantage as $\mathsf{Adv}^{\mathsf{IND\text{-}CCPA}}_{\mathsf{PKENO},\mathcal{A}}(\lambda) = |\mathsf{Pr}[\beta = \beta'] - \frac{1}{2}|$.

Definition 4. *We say that a PKENO scheme is IND-CCPA secure, if for every adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{IND\text{-}CCPA}}_{\mathsf{PKENO},\mathcal{A}}(\cdot)$ is negligible.*

Computational proof soundness is defined using the following game between an attack algorithm $\mathcal{A}$ and a challenger.

Setup. The challenger runs $\mathsf{KeyGen}(\lambda)$ to obtain a public/private key pair (PK, SK). It gives the key pair (PK, SK) to the adversary.
Challenge. The adversary $\mathcal{A}$ submits a message m. The challenger sends $C = \mathsf{Encrypt}(\mathsf{PK}, m)$ to the adversary.
Output. The adversary $\mathcal{A}$ outputs (m', π').

We define $\mathcal{A}$'s advantage in forging proof by $\mathsf{Adv}^{\mathsf{snd}}_{\mathsf{PKENO},\mathcal{A}}(\lambda) = \mathsf{Pr}[1 \leftarrow \mathsf{Ver}(\mathsf{PK}, C, m', \pi') \wedge m' \neq m]$.

Definition 5. *We say that a PKENO scheme satisfies computational proof soundness, if for every adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{snd}}_{\mathsf{PKENO},\mathcal{A}}(\cdot)$ is negligible.*

Definition 6. *We say that a PKENO scheme is secure, if it is IND-CCPA secure and satisfies computational proof soundness.*

2.6 Threshold Public Key Encryption

A threshold public key encryption (TPKE) scheme is a tuple of algorithms described as follows:

$\mathsf{Setup}(n, k, \lambda)$. Takes as input the number of decryption servers n, a threshold k where $1 \leq k \leq n$ and a security parameter λ. It outputs a public key PK, a verification key VK and private key $\mathsf{SK} = (\mathsf{SK}_1, \ldots, \mathsf{SK}_n)$ which is a vector of n private key shares. The verification key VK is used to check validity of responses from decryption servers.
$\mathsf{Encrypt}(\mathsf{PK}, m)$. Takes as input a public key PK and a message m. It outputs a ciphertext.
$\mathsf{ShareDecrypt}(\mathsf{SK}_i, C)$. Takes as input a private key share SK_i and a ciphertext C. It outputs a decryption share $\mu_i = (i, d_{C,i})$ or the special symbol $(i, \perp)$.
$\mathsf{ShareVerify}(\mathsf{VK}, C, \mu_i)$. Takes as input the verification key VK, a ciphertext C and a decryption share μ_i. It outputs `valid` meaning that μ_i is a valid decryption share of C or `invalid`.
$\mathsf{Combine}(\mathsf{PK}, \mathsf{VK}, C, \{\mu_1, \ldots, \mu_k\})$. Takes as input the public key PK, the verification key VK, a ciphertext C and k decryption shares $\mu_1, \ldots, \mu_k$. It outputs a plaintext message or the special symbol $\perp$.

We require, for all ciphertext C, $\mathsf{ShareVerify}(\mathsf{VK}, C, \mathsf{ShareDecrypt}(\mathsf{SK}_i, C)) =$ `valid`. In addition, let $\mu_1, \ldots, \mu_k$ are k distinct valid decryption shares of C, where $C = \mathsf{Encrypt}(\mathsf{PK}, m)$, then we require $\mathsf{Combine}(\mathsf{PK}, \mathsf{VK}, C, \{\mu_1, \ldots, \mu_k\}) = m$.

Security against chosen ciphertext attack is defined using the following game between an attack algorithm $\mathcal{A}$ and a challenger.

Init. The adversary outputs a set $S \subset \{1, \ldots, n\}$ of $k-1$ decryption servers to corrupt.
Setup. The challenger runs $\mathsf{Setup}(n, k, \lambda)$ to obtain a triple (PK, VK, SK). It gives PK, VK, and all (j, SK_j) for $j \in S$ to the adversary.

Query phase 1. The adversary $\mathcal{A}$ adaptively issues decryption queries (C, i). The challenger responds with $\mathsf{ShareDecrypt}(\mathsf{SK}_i, C)$.

Challenge. The adversary $\mathcal{A}$ submits two (equal length) messages m_0, m_1. The challenger selects a random bit $\beta \in \{0,1\}$, sets $C^* = \mathsf{Encrypt}(\mathsf{PK}, m_\beta)$ and sends C^* to the adversary as its challenge ciphertext.

Query phase 2. The adversary continues to adaptively issue decryption queries (C, i), as in Query phase 1, but with the natural constraint that the adversary may not request the decryption of C^*.

Guess. The adversary $\mathcal{A}$ outputs its guess $\beta' \in \{0,1\}$ for b and wins the game if $\beta = \beta'$.

We define $\mathcal{A}$'s advantage as $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{TPKE}}(\lambda) = |\mathsf{Pr}[\beta = \beta'] - \frac{1}{2}|$.

Definition 7. *We say that a threshold public key encryption scheme TPKE is secure, if for every adversary, the advantage $\mathsf{Adv}_{\mathcal{A}}^{TPKE}(\cdot)$ is negligible.*

3 The Proposed PKE Scheme

Our scheme is motivated by the recent signature scheme by Hohenberger and Waters [23]. Recall that in the CHK, BK and BMW schemes, for each encryption, the encryptor first generates a one-time "identity", and then encrypts the message with respect to the "identity". In the CHK and BK schemes, the one-time "identity" is generated randomly by the encryptor; in the BMW scheme, the first two elements of a ciphertext are hashed to form the one-time "identity". In our proposed PKE scheme, we make use of two "identities". One "identity" is generated randomly as in the CHK and BK schemes, while the other "identity" is generated based on the approach in the BMW scheme. The benefit of doing this is twofold. Compared with the CHK and BK schemes, our ciphertexts are short without attached signature or MAC; and compared with the BMW scheme, our scheme has small public key size and is proven secure with a tight security reduction.

Our scheme consists of the following algorithms:

$\mathsf{KeyGen}(\lambda)$. Given the security parameter λ, a bilinear map group system $\langle p, \mathbb{G}, \mathbb{G}_T, e\rangle$ is constructed. Pick a generator g of $\mathbb{G}$, select random $\alpha, x, y, z \in \mathbb{Z}_p$ and set $g_1 = g^\alpha, u = g^x, v = g^y, d = g^z$. Next, choose random element $g_2 \in \mathbb{G}$. Finally, choose a collision-resistant hash function $H : \mathbb{G}_T \times \mathbb{G} \to \mathbb{Z}_p$. The published public key is

$$\mathsf{PK} = (p, \mathbb{G}, \mathbb{G}_T, e, g, H,\ Z = e(g_1, g_2),\ u, v, d),$$

and the private key is $\mathsf{SK} = (g_2^\alpha, x, y, z)$.

$\mathsf{Encrypt}(\mathsf{PK}, m)$. Given PK and a message $m \in \mathbb{G}_T$, randomly choose $s, r \in \mathbb{Z}_p$ and compute

$$C_0 = m \cdot Z^s,\ C_1 = g^s,\ C_2 = (u^t v^r d)^s,$$

where $t = H(C_0, C_1)$. Finally, output the ciphertext $C = (C_0, C_1, C_2, r) \in \mathbb{G}_T \times \mathbb{G}^2 \times \mathbb{Z}_p$.

$\mathsf{Decrypt}(\mathsf{SK}, C)$. Given $\mathsf{SK} = (g_2^\alpha, x, y, z)$ and a ciphertext $C = (C_0, C_1, C_2, r)$, compute $t = H(C_0, C_1)$. Then check whether

$$(C_1)^{tx+ry+z} = C_2.$$

If not, output $\perp$, else output

$$C_0 / e(C_1, g_2^\alpha).$$

Theorem 1. *The above public key encryption scheme is* (t, q, ϵ) *IND-CCA secure, assuming the* (t', ϵ')*-DBDH assumption holds in* $\mathbb{G}$ *(the multiplicative group of prime order* p*), where*

$$t' = t + O(q),\ \epsilon' \geq \epsilon - \mathsf{Adv}_{\mathcal{A}}^{CR} - q/p.$$

Proof. Suppose there exists a (t, q, ϵ)-IND-CCA adversary $\mathcal{A}$ against our public key encryption scheme. We are going to construct another PPT $\mathcal{B}$ that makes use of $\mathcal{A}$ to solve the DBDH problem with probability at least ϵ' and in the time at most t'.

$\mathcal{B}$ is given as input a random 5-tuple (g, g^a, g^b, g^c, T) that is either sampled from $\mathcal{P}_{BDH}$ (where $T = e(g,g)^{abc}$) or from $\mathcal{R}_{BDH}$ (where T is uniform and independent in $\mathbb{G}_T$). Algorithm $\mathcal{B}$'s goal is to output 1 if $T = e(g,g)^{abc}$ and 0 otherwise. Algorithm $\mathcal{B}$ runs $\mathcal{A}$ executing the following steps.

Setup. $\mathcal{B}$ chooses random $x_v, x_d, y_u, y_v, y_d \in \mathbb{Z}_p$ and sets $g_1 = g^a, g_2 = g^b, u = g^b g^{y_u}, v = g^{bx_v} g^{y_v}, d = g^{bx_d} g^{y_d}$. Then, choose a target-collision resistant hash function $H : \mathbb{G}_T \times \mathbb{G} \to \mathbb{Z}_p$. The public key

$$\mathsf{PK} = (p, \mathbb{G}, \mathbb{G}_T, e, g, H,\ Z = e(g_1, g_2),\ u, v, d)$$

is passed to $\mathcal{A}$. The private key is $\mathsf{SK} = (g_2^\alpha = g_2^a = g^{ab}, x = b + y_u, y = bx_v + y_v, z = bx_d + y_d)$ which is unknown to $\mathcal{B}$.

Query phase 1. When $\mathcal{A}$ issues decryption query on a ciphertext $C = (C_0, C_1, C_2, r)$, $\mathcal{B}$ first computes $t = H(C_0, C_1)$ and checks whether

$$e(C_1, u^t v^r d) = e(g, C_2).$$

If not, output $\perp$. Check whether $t + rx_v + x_d = 0$. If so, $\mathcal{B}$ aborts and randomly outputs a bit, else chooses random $\gamma \in \mathbb{Z}_p$ and computes

$$d_C^1 = g_1^{-(ty_u + ry_v + y_d)/(t + rx_v + x_d)} (u^t v^r d)^\gamma,$$
$$d_C^2 = g_1^{-1/(t + rx_v + x_d)} g^\gamma.$$

Let $\tilde{\gamma} = \gamma - \frac{a}{(t + rx_v + x_d)}$. Then we have

$$d_C^1 = g_2^a (u^t v^r d)^{\tilde{\gamma}},\ d_C^2 = g^{\tilde{\gamma}}.$$

Finally, $\mathcal{B}$ outputs

$$C_0 \cdot e(C_2, d_C^2) / e(C_1, d_C^1).$$

Challenge. The adversary $\mathcal{A}$ outputs two equal-length plaintexts (m_0, m_1). $\mathcal{B}$ flips a fair coin, $\beta \in \{0,1\}$ and constructs the ciphertext as follows:

1. It computes

$$C_0^* = m_\beta \cdot T,\ C_1^* = g^c, t^* = H(C_0^*, C_1^*).$$

2. Then, it sets $r^* = -(t^* + x_d)/x_v$ and computes $C_2^* = (g^c)^{(t^* y_u + r^* y_v + y_d)}$.
3. Finally, return the ciphertext $C^* = (C_0^*, C_1^*, C_2^*, r^*)$.

Since $C^* = (m_\beta \cdot T,\ g^c,\ (u^{t^*} v^{r^*} d)^c,\ r^*)$, the challenge ciphertext is a valid encryption of m_β with the correct distribution whenever $T = e(g,g)^{abc} = e(g_1, g_2)^c$ (as is the case when the input 5-tuple is sampled from $\mathcal{P}_{BDH}$). On the other hand, when T is uniform and independent in $\mathbb{G}_T$ (which occurs when the input 5-tuple is sampled from $\mathcal{R}_{BDH}$) the challenge ciphertext C^* is independent of β in the adversary's view.

Query phase 2. $\mathcal{A}$ continues to adaptively issue decryption query $C = (C_0, C_1, C_2, r)$, $\mathcal{B}$ performs the following steps:

1. Check if $C = C^*$. If so, output $\perp$.
2. Check if $C = (C_0, C_1^*, C_2^*, r^*)$ and $H(C_0, C_1) = t^*$. If so, $\mathcal{B}$ aborts and randomly outputs a bit.
   Note that, if $\mathcal{A}$ were able to produce such a ciphertext, this would represent a collision in the hash function H, and so the probability that this event occurs is negligible.
3. Check if $t + rx_v + x_d = 0$ where $t = H(C_0, C_1)$. If so, $\mathcal{B}$ aborts and randomly outputs a bit, else $\mathcal{B}$ responds as in Query phase 1.

   Observe that the values x_v and x_d are initially hidden by blinding factors y_v and y_d, respectively.

   When the adversary $\mathcal{A}$ issues decryption query $C = (C_0, C_1, C_2, r)$:
   - if $e(C_1, u^t v^r d) \neq e(g, C_2)$, $\mathcal{B}$ outputs $\perp$ and do not leak any information about either x_v or x_d.
   - else $e(C_1, u^t v^r d) = e(g, C_2)$, $\mathcal{B}$ computes $(d_C^1 = g_2^a (u^t v^r d)^{\tilde{\gamma}}, d_C^2 = g^{\tilde{\gamma}})$ and outputs

$$\begin{aligned} C_0 \cdot \frac{e(C_2, d_C^2)}{e(C_1, d_C^1)} &= C_0 \cdot \frac{e(C_2, g^{\tilde{\gamma}})}{e(C_1, g_2^a (u^t v^r d)^{\tilde{\gamma}})} \\ &= C_0 \cdot \frac{e(C_2, g)^{\tilde{\gamma}}}{e(C_1, g_2^a) \cdot e(C_1, u^t v^r d)^{\tilde{\gamma}}} = \frac{C_0}{e(C_1, g_2^a)}. \end{aligned}$$

   So, the adversary could not obtain any information about either x_v or x_d from the decryption queries.

   For the challenge ciphertext, the adversary could obtain the information that $t^* + r^* x_v + x_d = 0$. However, there are exactly p possible (x_v, x_d) pairs that satisfy this equation and each of them are equally likely.

   Thus, information-theoretically, the probability that $t + rx_v + x_d = 0$ is at most $1/p$.

Guess. The adversary $\mathcal{A}$ outputs a bit β'. $\mathcal{B}$ concludes its own game by outputting a guess as follows. If $\beta' = \beta$ then $\mathcal{B}$ outputs 1 meaning $T = e(g,g)^{abc}$. Otherwise, it outputs 0 meaning $T \neq e(g,g)^{abc}$.

The probability that $\mathcal{B}$ does not abort during the simulation is at most $\mathsf{Adv}^{\mathsf{CR}}_{\mathcal{A}}+ q/p$. When the input 5-tuple is sampled from $\mathcal{P}_{BDH}$ (where $T = e(g,g)^{abc}$) and $\mathcal{B}$ does not abort then $\mathcal{A}$'s view is identical to its view in a real attack game. On the other hand, when the input 5-tuple is sampled from $\mathcal{R}_{BDH}$ (where T is uniform in $\mathbb{G}_T$) and $\mathcal{B}$ does not abort then the advantage that $\mathcal{A}$ wins the attack game is $1/2$. The running time of $\mathcal{A}$ is dominated by the paring computation in response to $\mathcal{A}$'s decryption queries.

This concludes the proof of Theorem 1.

4 Practical Extensions

In this section, we describe two interesting extensions to our PKE scheme. In the following, we only present the extended schemes. Their security proofs can be performed in a similar manner as in Section 3 and are therefore omitted.

4.1 Public Key Encryption with Non-interactive Opening

Public key encryption with non-interactive opening (PKENO) was recently introduced in [13,14] as a means to enable publicly-verifiable decryption. In a PKENO scheme, the receiver of a ciphertext C can, convincingly and without interaction, reveal what the result was of decryption C, without compromising the confidentiality of non-opened ciphertexts. The construction of PKENO can be obtained by using public key encryption with witness-recovering decryption (PKEWR) [30]. Here the receiver can efficiently reconstruct the "randomness" that was used for encryption. This randomness then serves as the proof. Verification performs re-encrypting using the randomness and the message. The proof is valid if the result equals the ciphertext. The existing constructions of PKEWR [30] in the standard model, however, are relatively inefficient since the ciphertext size is linear in the message length.

Damgård, Hofheinz, Kiltz and Thorbek [14] proposed two efficient constructions of PKENO schemes. The first proposal is a generic construction and resembles the CHK transformation [9]. The idea is to use, for each PKENO encryption, a fresh random verification key of a one-time signature scheme as the "identity" for IBE encryption. The private key corresponding to the "identity" serves as the proof. Verification performs decryption using the private key. The second proposal is a concrete scheme based on the CCA-secure key encapsulation mechanism by Boyen, Mei and Waters [7]. Recently, Galindo [18] showed the second scheme in [14] is insecure and proposed a fix based on direct CCA-secure PKE from identity-based techniques by Boyen, Mei and Waters [7]. Their scheme needs *long* public keys. Based on our PKE scheme, we propose a more efficient PKENO scheme as detailed in the following.

$\mathsf{KeyGen}(\lambda)$. Given the security parameter λ, a bilinear map group system $\langle p, \mathbb{G}, \mathbb{G}_T, e\rangle$ is constructed. Pick a generator g of $\mathbb{G}$, select random $\alpha, x, y, z \in \mathbb{Z}_p$ and set $g_1 = g^{\alpha}, u = g^{x}, v = g^{y}, d = g^{z}$. Next, choose random element

$g_2 \in \mathbb{G}$. Finally, choose a collision-resistant hash function $H : \mathbb{G}_T \times \mathbb{G} \to \mathbb{Z}_p$. The published public key is

$$\mathsf{PK} = (p, \mathbb{G}, \mathbb{G}_T, e, g, H,\ Z = e(g_1, g_2),\ u, v, d),$$

and the private key is $\mathsf{SK} = (g_2^\alpha, x, y, z)$.

$\mathsf{Encrypt}(\mathsf{PK}, m)$. Given PK and a message $m \in \mathbb{G}_T$, randomly choose $s, r \in \mathbb{Z}_p$ and compute

$$C_0 = m \cdot Z^s = m \cdot e(g_1, g_2)^s,\ C_1 = g^s,\ C_2 = (u^t v^r d)^s,$$

where $t = H(C_0, C_1)$. Finally, output the ciphertext $C = (C_0, C_1, C_2, r) \in \mathbb{G}_T \times \mathbb{G}^2 \times \mathbb{Z}_p$.

$\mathsf{Decrypt}(\mathsf{SK}, C)$. Given $\mathsf{SK} = (g_2^\alpha, x, y, z)$ and a ciphertext $C = (C_0, C_1, C_2, r)$, compute $t = H(C_0, C_1)$. Then check whether

$$(C_1)^{tx+ry+z} = C_2.$$

If not, output $\perp$, else output

$$C_0 / e(C_1, g_2^\alpha).$$

$\mathsf{Prove}(\mathsf{SK}, C)$. Given $\mathsf{SK} = (g_2^\alpha, x, y, z)$ and a ciphertext $C = (C_0, C_1, C_2, r)$, compute $t = H(C_0, C_1)$. Then check whether

$$(C_1)^{tx+ry+z} = C_2.$$

If not, output $\perp$, else randomly choose $\gamma \in \mathbb{Z}_p$ and output $\pi = (d_C^1, d_C^2) \in \mathbb{G}^2$, where

$$d_C^1 = g_2^\alpha (u^t v^r d)^\gamma,\ d_C^2 = g^\gamma.$$

$\mathsf{Ver}(\mathsf{PK}, C, m, \pi)$. Given PK, a ciphertext $C = (C_0, C_1, C_2, r)$, a message m and a proof $\pi = (d_C^1, d_C^2)$, compute $t = H(C_0, C_1)$. Then check whether

$$e(C_1, u^t v^r d) = e(g, C_2),\ e(g, d_C^1) = Z \cdot e(u^t v^r d, d_C^2) \text{ and}$$
$$m = C_0 \cdot e(C_2, d_C^2) / e(C_1, d_C^1).$$

If not, output 0, else output 1.

4.2 Threshold Public Key Encryption

In a threshold public key encryption (TPKE) scheme [15], the private key corresponding to a public key is shared among a set of n decryption servers. In such a scheme, a message is encrypted and sent to a group of decryption servers, in such a way that the cooperation of at least k of them (where k is the threshold) is necessary in order to recover the original message. In a non-interactive threshold scheme, no communication is needed amongst the decryption servers performing the partial decryptions. Such schemes have many applications in situations where one cannot fully trust a unique person, but possibly a pool of individuals.

Recall that the Cramer-Shoup scheme [11] provides efficient CCA-secure encryption without random oracles. The scheme requires that the private key be used to check ciphertext validity during decryption. In a threshold environment none of the decryption servers possess the private key needed to perform this validity check. Consequently, constructing a threshold version of the Cramer-Shoup scheme is non-trivial.

Boneh, Boyen and Halevi [5] showed that CCA-secure threshold public key encryption schemes (without random oracles) are easier to derive from selective-ID CPA secure identity based encryption than from the Cramer-Shoup paradigm. The main reason is that in the IBE-to-CCA transformation [9], the validity check performed during decryption requires only the public key. Consequently, each decryption server can check ciphertext validity on its own and only release a partial decryption of valid ciphertexts. Note that the more efficient transformation of Boneh and Katz [6] does not have this property and is thus less suitable for threshold decryption.

Boyen, Mei and Waters [7] gave a very simple and efficient CCA-secure threshold key encapsulation mechanism (KEM) based on the Boneh-Boyen IBE framework. However, designing a full threshold PKE from a threshold KEM is not an easy task. Let us have a glimpse on it. A standard (hybrid) PKE scheme can be obtained by using the KEM to securely transport a random session key that is fed into a symmetric encryption scheme to encrypt the plaintext message. If both the KEM and the symmetric encryption scheme are chosen-ciphertext secure, then the resulting hybrid PKE is also chosen-ciphertext secure. A symmetric encryption scheme secure against chosen-ciphertext attacks can be built from relatively weak primitives, i.e. from any (one-time) symmetric encryption scheme by essentially adding a MAC. Unfortunately, sharing a MAC is not trivial in general, and will often lead to costly computations.

In our PKE scheme, the decryptor needs to verify the ciphertext before attempting to decrypt it. This check is efficiently performed using a single exponentiation in $\mathbb{G}$, but requires knowledge of the private key (the exponents x, y, z). In fact, the validity check could have been performed publicly, using additional application of the bilinear map, by checking whether $e(C_1, u^t v^r d) = e(g, C_2)$. Since under such a modification the ciphertext validity check no longer requires the private key, our PKE scheme is suitable for non-interactive threshold decryption. The following is the detailed construction of the threshold version of our PKE scheme. It bears some resemblance to the threshold schemes in [5] due to its roots in identity-based techniques.

$\mathsf{Setup}(n, k, \lambda)$. Given the security parameter λ, a bilinear map group system $\langle p, \mathbb{G}, \mathbb{G}_T, e\rangle$ is constructed. Select random generators g, g_2, u, v, d of $\mathbb{G}$ and a random degree $k-1$ polynomial $f \in \mathbb{Z}_p[X]$. Set $\alpha = f(0) \in \mathbb{Z}_p$ and $g_1 = g^\alpha$. Choose a collision-resistant hash function $H : \mathbb{G}_T \times \mathbb{G} \to \mathbb{Z}_p$. The published public key is

$$\mathsf{PK} = (p, \mathbb{G}, \mathbb{G}_T, e, g, H,\ Z = e(g_1, g_2),\ g_2, u, v, d).$$

For $i = 1, \ldots, n$ the secret key SK_i of server i is defined as $\mathsf{SK}_i = g_2^{f(i)}$. The public verification key VK is the n-tuple $(g^{f(1)}, \ldots, g^{f(n)})$.

$\mathsf{Encrypt}(\mathsf{PK}, m)$. Given PK and a message $m \in \mathbb{G}_T$, randomly choose $s, r \in \mathbb{Z}_p$ and compute

$$C_0 = m \cdot Z^s = m \cdot e(g_1, g_2)^s,\ C_1 = g^s,\ C_2 = (u^t v^r d)^s,$$

where $t = H(C_0, C_1)$. Finally, output the ciphertext $C = (C_0, C_1, C_2, r) \in \mathbb{G}_T \times \mathbb{G}^2 \times \mathbb{Z}_p$.

$\mathsf{ShareDecrypt}(\mathsf{SK}_i, C)$. Given SK_i and a ciphertext $C = (C_0, C_1, C_2, r)$, decryption server i computes $t = H(C_0, C_1)$. Then check whether

$$e(C_1, u^t v^r d) = e(g, C_2).$$

If not, output $\mu_i = (i, \perp)$, else randomly choose $\gamma \in \mathbb{Z}_p$ and output the decryption share $\mu_i = (i, (d^1_{C,i}, d^2_{C,i}))$, where

$$d^1_{C,i} = \mathsf{SK}_i \cdot (u^t v^r d)^\gamma,\ d^2_{C,i} = g^\gamma.$$

$\mathsf{ShareVerify}(\mathsf{VK}, C, \mu_i)$. Given $\mathsf{VK} = (h_1, \ldots, h_n)$ where $h_i = g^{f(i)}$, a ciphertext $C = (C_0, C_1, C_2, r)$ and a decryption share $\mu_i = (i, (d^1_{C,i}, d^2_{C,i}))$ of the ciphertext C, compute $t = H(C_0, C_1)$. Then check whether

$$e(C_1, u^t v^r d) = e(g, C_2) \text{ and } e(g, d^1_{C,i}) = e(h_i, g_2) \cdot e(u^t v^r d, d^2_{C,i}).$$

If not, output `invalid`, else output `valid`.

$\mathsf{Combine}(\mathsf{PK}, \mathsf{VK}, C, \{\mu_1, \ldots, \mu_k\})$. Given PK, VK, a ciphertext $C = (C_0, C_1, C_2, r)$ and the partial decryptions $\mu_1, \ldots, \mu_k$, first check that all decryption shares $\mu_i = (i, (d^1_{C,i}, d^2_{C,i}))$ bear distinct server indices i, and that they are all *valid*, i.e., that all $\mathsf{ShareVerify}(\mathsf{VK}, C, \mu_i) = \texttt{valid}$; otherwise output $\perp$. Without loss of generality, assume that the shares $\mu_1, \ldots, \mu_k$ were generated by the decryption servers $i = 1, \ldots, k$, respectively. Then compute the Lagrange coefficients $\lambda_1, \ldots, \lambda_k \in \mathbb{Z}_p$ so that $\alpha = f(0) = \sum_{i=1}^k \lambda_i f(i)$, and set

$$d^1_C = \prod_{i=1}^k (d^1_{C,i})^{\lambda_i},\ d^2_C = \prod_{i=1}^k (d^2_{C,i})^{\lambda_i}.$$

Finally, output

$$C_0 \cdot e(C_2, d^2_C) / e(C_1, d^1_C).$$

5 Conclusions

We described an efficient CCA-secure public key encryption scheme whose performance is competitive with previous CCA-secure public key encryption schemes employing identity-based techniques. Our scheme is based on the identity-based encryption schemes of Boneh and Boyen [4], and the signature scheme of Hohenberger and Waters [23]. In addition, we showed that our scheme is well suited for constructing TPKE and PKENO schemes. In fact, our approach can be applied to obtain more efficient CCA-secure hierarchical identity based encryption (HIBE) scheme based on the Waters CPA-secure HIBE scheme [33].

Acknowledgement

We are grateful to the anonymous reviewers for their helpful comments. This work is partially funded by National Natural Science Foundation of China (No. 60873229) and Shanghai Rising-star Program (No. 09QA1403000), and also supported in part by the Office of Research, Singapore Management University.

References

1. Abe, M., Gennaro, R., Kurosawa, K., Shoup, V.: Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 128–146. Springer, Heidelberg (2005)
2. Bellare, M., Boldyreva, A., Palacio, A.: An Uninstantiable Random-Oracle-Model Scheme for a Hybrid-Encryption Problem. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 171–188. Springer, Heidelberg (2004)
3. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proc. of ACM CCS 1993, pp. 62–73. ACM Press, New York (1993)
4. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
5. Boneh, D., Boyen, X., Halevi, S.: Chosen ciphertext secure public key threshold encryption without random oracles. In: Pointcheval, D. (ed.) CT-RSA 2006. LNCS, vol. 3860, pp. 226–243. Springer, Heidelberg (2006)
6. Boneh, D., Katz, J.: Improved efficiency for CCA-secure cryptosystems built using identity-based encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
7. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: Proc. of ACM CCS 2005, pp. 320–329. ACM Press, New-York (2005)
8. Canetti, R., Goldreich, O., Halevi, S.: The Random Oracle Model Revisited. In: Proceedings of STOC 1998. ACM, New York (1998)
9. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
10. Cash, D., Kiltz, E., Shoup, V.: The twin Diffie-Hellman problem and applications. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 127–145. Springer, Heidelberg (2008)
11. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
12. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
13. Damgård, I., Thorbek, R.: Non-interactive proofs for integer multiplication. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 412–429. Springer, Heidelberg (2007)
14. Damgård, I., Hofheinz, D., Kiltz, E., Thorbek, R.: Public-key encryption with non-interactive opening. In: Malkin, T.G. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 239–255. Springer, Heidelberg (2008)

15. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 307–315. Springer, Heidelberg (1990)
16. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. In: Proc. of STOC 1991, pp. 542–552 (1991)
17. Elkind, E., Sahai, A.: A unified methodology for constructing public-key encryption schemes secure against adaptive chosen-ciphertext attack. Cryptology ePrint Archive, Report 2002/042 (2002), `http://eprint.iacr.org/`
18. Galindo, D.: Breaking and Repairing Damgård *et al.* Public Key Encryption Scheme with Non-interactive Opening. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 389–398. Springer, Heidelberg (2009)
19. Goldwasser, S., Tauman, Y.: On the (In)security of the Fiat-Shamir Paradigm. In: Proc. of FOCS. IEEE, Los Alamitos (2003)
20. Hanaoka, G., Kurosawa, K.: Efficient Chosen Ciphertext Secure Public Key Encryption under the Computational Diffie-Hellman Assumption. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 308–325. Springer, Heidelberg (2008)
21. Hofheinz, D., Kiltz, E.: Secure hybrid encryption from weakened key encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (2007)
22. Hofheinz, D., Kiltz, E.: Practical Chosen Ciphertext Secure Encryption from Factoring. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 313–332. Springer, Heidelberg (2009)
23. Hohenberger, S., Waters, B.: Realizing Hash-and-Sign Signatures under Standard Assumptions. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 333–350. Springer, Heidelberg (2009)
24. Kiltz, E.: On the Limitations of the Spread of an IBE-to-PKE Transformation. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T.G. (eds.) PKC 2006. LNCS, vol. 3958, pp. 274–289. Springer, Heidelberg (2006)
25. Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
26. Kiltz, E.: Chosen-ciphertext secure key-encapsulation based on gap hashed Diffie-Hellman. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 282–297. Springer, Heidelberg (2007)
27. Kurosawa, K., Desmedt, Y.: A new paradigm of hybrid encryption scheme. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Heidelberg (2004)
28. Naor, M., Pinkas, B.: Efficient trace and revoke schemes. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 1–20. Springer, Heidelberg (2001)
29. Nielsen, J.B.: Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Non-committing Encryption Case. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 111–126. Springer, Heidelberg (2002)
30. Peikert, C., Waters, B.: Lossy Trapdoor Functions and Their Applications. In: STOC 2008, pp. 187–196. ACM, New York (2008)
31. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992)
32. Shoup, V.: Using hash functions as a hedge against chosen ciphertext attack. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 275–288. Springer, Heidelberg (2000)
33. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)