URDP: General Framework for Direct CCA2 Security from any Lattice-Based PKE Scheme

Roohallah Rastaghi

r.rastaghi59@gmail.com

22 February 2013

Abstract. Design efficient lattice-based cryptosystem secure against adaptive chosen ciphertext attack (IND-CCA2) is a challenge problem. To the date, full CCA2-security of all proposed lattice-based cryptosystems achieved by using a generic transformations such as either strongly unforgeable one-time signature schemes (SU-OT-SS), or a message authentication code (MAC) and weak form of commitment. The drawback of these schemes is that encryption requires *separate encryption*. Therefore, the resulting encryption scheme is not sufficiently efficient to be used in practice and it is inappropriate for many applications such as small ubiquitous computing devices with limited resources such as smart cards, active RFID tags, wireless sensor networks and other embedded devices.

In this work, for the first time, we introduce an efficient universal random data padding (URDP) scheme, and show how it can be used to construct a *direct* CCA2-secure encryption scheme from *any* worst-case hardness problems in (ideal) lattice in the standard model, resolving a problem that has remained open till date. This novel approach is a *black-box* construction and leads to the elimination of separate encryption, as it avoids using general transformation from CPA-secure scheme to a CCA2-secure one. IND-CCA2 security of this scheme can be tightly reduced in the standard model to the assumption that the underlying primitive is an one-way trapdoor function.

Keywords: Post-quantum cryptography, Lattice-based PKE scheme, Universal random data padding, CCA2-security, Standard model

1 Introduction

Devising quantum computer will enable us to break public-key cryptosystems based on integer factoring (IF) and discrete logarithm (DL) problems[17] . Under this future threat, it is important to search for secure PKEs based on the other problem. Lattice-based PKE schemes hold a great promise for post-quantum cryptography, as they enjoy very strong security proofs based on worst-case hardness, relatively efficient implementations, as well as great simplicity and, lately, their promising potential as a platform for constructing advanced functionalities.

The ultimate goal of public-key encryption is the production of a simple and efficient encryption scheme that is provably secure in a strong security model under a weak and reasonable computational assumption. The accepted notion for the security of a public-key encryption scheme is semantically secure against adaptive chose ciphertext attack (i.e. IND-CCA2) [13] . In this scenario, the adversary has seen the *challenge ciphertext* before having access to the decryption oracle. The adversary is not allowed to ask the decryption of the challenge ciphertext, but can obtain the decryption of *any* relevant cryptogram (even modified ones based on the challenge ciphertext). A cryptosystem is said to be CCA2-secure if the cryptanalyst fails to obtain any partial information about the plaintext relevant to the challenge ciphertext.

1.1 Related work

In order to design CCA2-secure lattice-based encryption schemes, a lot of successes were reached. There are two approach for constructing CCA2-secure lattice-based cryptosystems in the standard model. Existing CCA2-secure schemes exhibit various incomparable tradeoffs between key size and error rate.

-*CCA-secure cryptosystem based on lossy trapdoor functions.* Peikert and Waters [11] showed for the first time how to construct CCA2-secure encryption scheme from a primitive called a *lossy* ABO trapdoor function family, along with a SU-OT-SS. They showed how to construct this primitive based on the learning with error (LWE) problem. This result is particularly important as it gives for the first time a CCA-secure cryptosystem based on the worst-case hardness of lattice problems. It has public-keys of size $\mathcal{O}(n^2)$ bits and relies on a quite small LWE error rate of $\alpha = \mathcal{O}(1/n^4)$. Subsequently, Peikert [12] showed how to construct a correlation-secure trapdoor function family from the LWE problem, and used it within the Rosen-Segev scheme [15] to obtain another lattice-based CCA-secure scheme. Unfortunately, the latter scheme also suffers from long public-key and ciphertext length of $\mathcal{O}(n^3)$ bits, but uses a better error rate of $\mathcal{O}(1/n)$ in the security parameter n, even if applied in the Ring-LWE setting. Recently, Micciancio and Peikert [10] give new methods for generating simpler, tighter, faster and smaller trapdoors in cryptographic lattices to achieve a CCA-secure cryptosystem. Their construction give a CCA-secure cryptosystem that enjoys the best of all prior constructions, which has $\mathcal{O}(n^2)$ bit public-keys, uses error rate $\mathcal{O}(1/n)$. Recently, Steinfeld et al. [18] introduced the first CCA2-secure variant of the NTRU [9] in the standard model with a provable security from worst-case problems in *ideal lattices*. They construct a CCA-secure scheme using the lossy trapdoor function, which they generalize it to the case of $(k-1)$-of-k-correlated input distributions.

-*CCA-secure cryptosystem based on IBE.* More constructions of IND-CCA2 secure lattice-based encryption schemes can be obtained by using the lattice-based selective-ID secure identity-based encryption (IBE) schemes of [1,2,3,4,7,14,16,19] within the generic constructions of [5,6], and a SU-OT-SS or commitment scheme.

All the above schemes use generic transformations from CPA to CCA2 security in the standard model, e.g., Dolev et al. approach [8], Canetti et al. paradigm [6] or Boneh et al. approach [5]. They typically involve either a SU-OT-SS or a MAC and commitment schemes to make the ciphertext authentic and non-malleable. So, the resulting encryption scheme requires *separate encryption* and thus, it is not sufficiently efficient to be used in practice and inappropriate for many applications such as small ubiquitous computing devices with limited resources such as smart cards, active RFID tags, wireless sensor networks and other embedded devices.

Till date, there is no generic *direct* transformation from *any* lattice-based one-way trapdoor cryptosystem (i.e., worst-case hardness problem in lattice) to a CCA2-secure one. In this work, for the first time, we show how to construct a CCA2-secure cryptosystem directly based on the worst-case hardness problems in lattice, resolving a problem that has remained open till date.

1.2 Our contributions

Our approach has several main benefits:

- It introduce a new generic asymmetric padding-based scheme. The main novelty is that our approach can be applied to *any* conjectured (post-quantum) one-way trapdoor cryptosystem.
- Our approach yields the first known *direct* CCA2-secure PKE scheme from worst-case hardness problems in lattice.
- The proposed approach is a "black-box" construction, which making it more efficient and technically simpler than those previously proposed. The publick/secret keys are as in the original scheme and the encryption/decryption complexity are comparable to the original scheme.
- This novel approach leads to the elimination of using generic transformations from CPA-secure schemes to a CCA2-secure one.
- Our CCA2-security proof is tightly based on the assumption that the underlying primitive is a trapdoor one-way function. So, the scheme's *consistency* check can be directly implemented by the *simulator* without having access to some external gap-oracle as in previous schemes [1,2,3,7,10,11,12,14,16,18,19]. Thus, our proof technique is fundamentally different from all known approaches to obtain CCA2-security in the lattice-based cryptosystems.
- Additionally, this scheme can be used for encryption of *arbitrary-length* long messages without employing the hybrid encryption method and symmetric encryption.

Organization. The rest of this manuscript is organized as follows: In the following section, we briefly explain some notations and definitions. Then, in Section

3, we introduce our proposed scheme. Security and performance analysis of the proposed scheme will be discussed in Section 4.

2 Preliminary

2.1 Notation

We will use standard notation. If x is a string, then $|x|$ denotes its length. If $k \in \mathbb{N}$, then $\{0, 1\}^k$ denote the set of k-bit strings, 1^k denote a string of k ones and $\{0, 1\}^*$ denote the set of bit strings of finite length. $y \leftarrow x$ denotes the assignment to y of the value x. For a set S, $s \leftarrow S$ denote the assignment to s of a uniformly random element of S. For a deterministic algorithm $\mathcal{A}$, we write $x \leftarrow \mathcal{A}^{\mathcal{O}}(y, z)$ to mean that x is assigned the output of running $\mathcal{A}$ on inputs y and z, with access to oracle $\mathcal{O}$. We denote by $\Pr[E]$ the probability that the event E occurs. If a and b are two strings of bits, we denote by $a\|b$ their concatenation. The bit-length of a denoted by $\mathsf{Len}(a)$, $\mathsf{Lsb}_{x_1}(a)$ means the right x_1 bits of a and $\mathsf{Msb}_{x_2}(a)$ means the left x_2 bits of a.

2.2 Definitions

Definition 1 (**Public-key encryption scheme**). *A public-key encryption scheme (PKE) is a triple of probabilistic polynomial time (PPT) algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *such that:*

- Gen *is a probabilistic polynomial-time key generation algorithm which takes a security parameter* 1^n *as input and outputs a public key pk and a secret-key sk. We write* $(pk, sk) \leftarrow \mathsf{Gen}(1^n)$. *The public key specifies the message space* $\mathcal{M}$ *and the ciphertext space* $\mathcal{C}$.
- Enc *is a (possibly) probabilistic polynomial-time encryption algorithm which takes as input a public key pk, a* $m \in \mathcal{M}$ *and random coins r, and outputs a ciphertext* $C \in \mathcal{C}$. *We write* $\mathsf{Enc}(pk, m; r)$ *to indicate explicitly that the random coins r is used and* $\mathsf{Enc}(pk, m)$ *if fresh random coins are used.*
- Dec *is a deterministic polynomial-time decryption algorithm which takes as input a secret-key sk and a ciphertext* $C \in \mathcal{C}$, *and outputs either a message* $m \in \mathcal{M}$ *or an error symbol* $\perp$. *We write* $m \leftarrow \mathsf{Dec}(C, sk)$.
- *(Completeness) For any pair of public and secret-keys generated by Gen and any message* $m \in \mathcal{M}$ *it holds that* $\mathsf{Dec}(sk, \mathsf{Enc}(pk, m; r)) = m$ *with overwhelming probability over the randomness used by* Gen *and the random coins r used by* Enc.

Definition 2 (**Padding scheme**). *Let* ν, ρ, k *be three integers such that* $\nu + \rho \leq k$. *A padding scheme* Π *consists of two mappings* $\pi : \{0,1\}^\nu \times \{0,1\}^\rho \rightarrow \{0,1\}^k$

and $\hat{\pi} : \{0,1\}^k \rightarrow \{0,1\}^\nu \times \{0,1\}^\rho \cup \{\perp\}$ *such that* π *is injective and the following consistency requirement is fulfilled:*

$$\forall m \in \{0,1\}^\nu, r \in \{0,1\}^\rho : \hat{\pi}(\pi(m,r)) = m.$$

Definition 3 (**CCA2-security).** *A public-key encryption scheme PKE is secure against adaptive chosen-ciphertext attacks (i.e. IND-CCA2) if the advantage of any two-stage PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *in the following experiment is negligible in the security parameter* k*:*

$\mathbf{Exp}^{cca2}_{\mathrm{PKE},\mathcal{A}}(k)$*:*

$(pk, sk) \leftarrow \mathsf{Gen}(1^k)$

$(m_0, m_1, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(sk,.)}(pk) \quad \text{s.t.} \quad |m_0| = |m_1|$

$b \leftarrow \{0,1\}$

$C^* \leftarrow \mathsf{Enc}(pk, m_b)$

$b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}(sk,.)}(C^*, \mathsf{state})$

if $b = b^{'}$ *return 1, else return 0.*

The attacker may query a decryption oracle with a ciphertext C *at any point during its execution, with the exception that* $\mathcal{A}_2$ *is not allowed to query* $\mathrm{Dec}(sk, .)$ *with* C^**. The decryption oracle returns* $b^{'} \leftarrow \mathcal{A}_2^{\mathrm{Dec}(sk,\,.)}(C^*, \mathsf{state})$*. The attacker wins the game if* $b = b'$ *and the probability of this event is defined as* $\Pr[\mathrm{Exp}^{\,cca2}_{\,\mathrm{PKE},\,\mathcal{A}}\,(k)]$*. We define the advantage of* $\mathcal{A}$ *in the experiment as*

$$\mathsf{Adv}^{\mathrm{IND-CCA2}}_{\mathrm{PKE},\,\mathcal{A}}\,(k) = \left| \Pr[\mathrm{Exp}^{\,cca2}_{\,\mathrm{PKE},\,\mathcal{A}}\,(k) = 1] - \frac{1}{2} \right|.$$

3 The proposed cryptosystem

In this section, we introduced our proposed CCA2-secure encryption scheme. Our scheme is a precoding-based algorithm which can transform any one-way trapdoor cryptosystem to a CCA2-secure one in the standard model. Precoding includes a permutation and pad some random obscure-data to the message bits.

3.1 The proposed idea

Let we can decide to encrypt message $m \in \{0,\,1\}^n$. At first, we perform a random encoding to the message bits. To do this, we uniformly choose $r = (r_1, \ldots, r_k) \in_R \{0,1\}^k$ with $k \ll n$ at random, and, suppose $\mathsf{wt}(r) = h$ be the its Hamming weigh. If n/h is an integer, then we can divide m into h blocks.

Otherwise, in order to divide m into h blocks, we must pad a random binary string (RBS) with length $h\,.\lceil n/h\rceil - n$ to the right of m. In each cases, we can divide m into h blocks $d_1\|d_2\|\ldots\|d_h$ with equal binary length $v = \lceil n/h\rceil$ where $d_h = \mathsf{Lsb}_{(n-(h-1)\,.\lceil n/h\rceil)}\,(m)\|\mathsf{RBS}$. Therefore, if $h \mid n$, then $\mathsf{RBS} = \varphi$ (the empty set) and $d_h = \mathsf{Lsb}_{(n-(h-1)\,.\lceil n/h\rceil)}\,(m)$, else, RBS is a random block with binary length $h\,.\lceil n/h\rceil - n$ and we have $d_h = \mathsf{Lsb}_{(n-(h-1)\,.\lceil n/h\rceil)}\,(m)\|\mathsf{RBS}$.

Now, we perform a random permutation and pad some random obscure blocks (ROBs) with equal binary length s into the message blocks d_i, $1 \le i \le h$ using padding scheme $\pi :\ \{0,\,1\} \times \{0,\,1\}^v \to \{0,1\}^v \times \{0,1\}^s$, which can be defined as follows:

$$\pi(r_i,\,d_i) = d_i^{'} = \begin{cases} d_{\sum_{j=1}^{i} r_j} & \text{if } r_i = 1 \\ \\ \mathsf{ROB} & \text{if } r_i = 0 \end{cases},\ 1 \le i \le k.$$

Notice that in order to prevent excessive increase in the message length, we can choose s small enough. The message $m^{'} = (d_1^{'}\|d_2^{'}\|\ldots\|d_k^{'})$ is called encoded message. We summarize encoding process in algorithm 1.

Algorithm 3.1: Random Encoding Algorithm.

Input: $m = (m_1,\ldots,m_n)$, $r \in_R \{0,1\}^k$ with $n \gg k$.

Output: Encoded message $m' = (d_1^{'}\|d_2^{'}\|\ldots\|d_k^{'})$.

SETUP:

1. $h \leftarrow \mathsf{wt}(r)$.
2. If $h \mid n$ then $v \leftarrow n/h$;

   else $v \leftarrow \lceil n/h\rceil$ and choose a RBS with binary length $h\cdot\lceil n/h\rceil - n$, and

   $$m \leftarrow (m_1,\ldots,m_n\| \underbrace{\mathsf{R\ B\ S}}_{h\cdot\lceil n/h\rceil - n}).$$
3. Divide m into h blocks $(d_1\|d_2\|\ \ldots\|d_h)$ with equal $\mathsf{Len}(d_i) = v$, $1 \le i \le h$.

PERMUTATION AND PADDING:

1. Uniformly choose integer s at random.
2. For $i = 1$ to k do;

   if $r_i = 1$ then $d_i^{'} \leftarrow d_{\sum_{j=1}^{i} r_j}$,

   else $d_i^{'} \leftarrow \mathsf{ROB}$ with binary length s.

   Return $m^{'} = (d_1^{'}\|d_2^{'}\|\ \ldots\|d_k^{'})$.

We illustrate algorithm (3.1) with small example. Suppose $m = (m_1,\ldots,m_{1117})$ and $r = (0,1,0,1,1,0,1,0,1,1,1,0,1,1,1,0,1,0)$.

SETUP:

We have $|m| = n = 1117$, $k = 18$ and $h = \sum_{i=1}^{k} r_i = 11$. Since $11 \nmid 1117$ so we must pad a RBS with binary length $h\,.\lceil n/h \rceil - n = 5$ to the right of m. If we uniformly chose $1, 0, 1, 1, 0$ at random, we have $m = (m_1, \ldots, m_{1117}, \underbrace{1, 0, 1, 1, 0}_{h\,.\lceil n/h \rceil - n})$. Since $h = 11$, the algorithm divides m into 11 blocks with equal length $v = \lceil n/h \rceil = 102$. We have $m = (\underbrace{m_1, \ldots, m_{102}}_{d_1} \| \underbrace{m_{103}, \ldots, m_{204}}_{d_2} \| \ldots \| \underbrace{m_{1020}, \ldots, m_{1117}, 1, 0, 1, 1, 0}_{d_{11}})$, where $\mathsf{Lsb}_{(n-(n-1)\cdot\lceil n/h \rceil)}(m) = \mathsf{Lsb}_{97}(m) = m_{1020}, \ldots, m_{1117}$.

PERMUTATION AND PADDING:

Firstly, we choose random integer s, say $s = 4$. We have

$r_1 = 0$, thus $d_1^{'} \leftarrow \mathsf{ROB}\#1 = (0, 1, 1, 0)$, where $(0, 1, 1, 0)$ is randomly chosen by algorithm 3.1.

$r_2 = 1$, thus $d_2^{'} \leftarrow d_{\sum_{j=1}^{2} r_j} = d_1$.

$r_3 = 0$, thus $d_3^{'} \leftarrow \mathsf{ROB}\#2 = (1, 0, 1, 0)$, where $(1, 0, 1, 0)$ is randomly chosen by algorithm 3.1.

$\vdots$

$r_{17} = 1$, thus $d_{17}^{'} \leftarrow d_{\sum_{j=1}^{17} r_j} = d_{11}$.

$r_{18} = 0$, thus $d_{18}^{'} \leftarrow \mathsf{ROB}\#(k-h) = 7 = (0, 0, 1, 0)$, where $(0, 0, 1, 0)$ is randomly chosen by algorithm 3.1.

$l - h$ ROB blocks with equal length $s = 4$ are combined with the message blocks $d_i, 1 \leq i \leq h$, to produce the encoded message $m^{'} = (d_1^{'} \| d_2^{'} \| \ldots \| d_k^{'})$. In the final, the algorithm outputs $m^{'}$ as $m^{'} = (\underbrace{0, 1, 1, 0}_{d_1^{'}} \| \underbrace{m_1, \ldots, m_{102}}_{d_2^{'}} \| \underbrace{1, 0, 1, 0}_{d_3^{'}} \| \ldots \| \underbrace{m_{1020}, \ldots, m_{1117}, 1, 0, 1, 1, 0}_{d_{17}^{'}} \| \| \underbrace{0, 0, 1, 0}_{d_{18}^{'}})$.

As we see, the *length* and the *position* of the message blocks d_i are correlated to the *number* and the *position* of the random bits $r_i = 1$ respectively, and completely random.

3.2 The proposed scheme

Now, we are ready to define our proposed encryption scheme. Given a secure lattice-based encryption scheme $\Pi_{\text{lbe}} = (\mathsf{Gen}_{\text{lbe}}, \mathsf{Enc}_{\text{lbe}}, \mathsf{Dec}_{\text{lbe}})$, we construct a IND-CCA2 secure encryption scheme $\Pi_{\text{cca2}} = (\mathsf{Gen}_{\mathsf{cca2}}, \mathsf{Enc}_{\mathsf{cca2}}, \mathsf{Dec}_{\mathsf{cca2}})$ as follows. This scheme can be used for encryption of *arbitrary-length* long messages.
System parameters. $n, k \in \mathbb{N}$, where $n \gg k$.

Key generation. Let $\mathsf{Gen}_{\mathrm{lbe}}$ be the Lattice-based key generator. On security parameter 1^k, the generator $\mathsf{Gen}_{\mathsf{cca2}}$ runs $\mathsf{Gen}_{\mathrm{lbe}}(1^k)$ to obtain

$$sk = sk_{\mathrm{lbe}} \quad \text{and} \quad pk = pk_{\mathrm{lbe}}.$$

Encryption. To encrypt message $m \in \{0, 1\}^n$ with $n \gg k$, $\mathsf{Enc}_{\mathsf{cca2}}(pk, m)$ works as follows.

- Uniformly chooses $r \in_R \{0,1\}^k$ at random and computes its Hamming weight $\mathsf{wt}(r) = h$.
- Randomly chooses small integer s and executes algorithm (3.1) for generate encoded message $m^{'} = (d_1^{'} \| d_2^{'} \| \ldots \| d_l^{'})$ from message m.
- Suppose y be the corresponding decimal value of $m^{'}$. Computes

$$C_1 = y \cdot h, \quad C_2 = \mathsf{Enc}_{\mathrm{lbe}}(pk, r)$$

and outputs the ciphertext $C = (C_1, C_2)$.

To handle CCA2-security and non-malleability related issues, we strictly correlate the message bits m_i, $1 \leq i \leq n$ to the randomness r via encoding process. The value of y also correlates to the random binary string r via its Hamming weight $h = \mathsf{wt}(r)$. So, the CCA2 adversary for extract the message blocks d_i from C_1 must first recover *exactly* the same random binary string r from lattice-based cryptosystem which is impossible, if the underlying lattice-based one-way trapdoor cryptosystem be secure.

Decryption. $\mathsf{Dec}_{\mathsf{cca2}}(sk, C)$ for extract message m performs the following steps.

- Computes random binary vector r as $r = \mathsf{Dec}_{\mathrm{lbe}}(C_2, sk)$ and $h = \sum_{i=1}^{k} r_i$.
- Computes $y = C_1/h$.
- Checks whether

$$\mathsf{Len}(y) \stackrel{?}{=} h \cdot \lceil n/h \rceil \tag{1}$$

  holds, and rejects if not (*consistency* check). If (1) hold, computes $v = \lceil n/h \rceil$ and binary coded decimal (BCD) m' of y.
- Computes $s = (|m'| - hv)/(k - h)$ and rejects the ciphertext if s is not an integers (verify whether the padding information is correct or not).
- The *lengths* and *position* of the message/ROB blocks are explicit, therefore, $\mathsf{Dec}_{\mathsf{cca2}}$ simply can separate ROB blocks from encoded message m' and extract message blocks d_i, $1 \leq i \leq h$ with the following algorithm.

Algorithm 3.2: Message Extractor.
Input: $r = (r_1, \ldots, r_k)$, integers h, v, s and encoded message m'.
Output: Retrieved message $m = (d_1 \| d_2 \| \ldots \| d_h)$

1. For $i = 1$ to k do

   If $r_i = 0$, then $m' \leftarrow \mathsf{Lsb}_{(|m'|-s)}(m')$,

   else $d_{\sum_{j=1}^{i} r_j} \leftarrow \mathsf{Msb}_v(m')$ and $m' \leftarrow \mathsf{Lsb}_{(|m'|-v)}(m')$;

2. $m \leftarrow (d_1 \| d_2 \| \ldots \| d_{\sum_{j=1}^{k} r_j})$, where $\sum_{i=1}^{k} r_i = h$.

3. If $h \nmid n$, then $m \leftarrow \mathsf{Msb}_n(m)$ (remove right $(h.\lceil n/h \rceil - n)$ bits of m).

Return "m".

4 Security and performance analysis

4.1 Security analysis

In this subsection, we proof the CCA2-security of the proposed cryptosystem which is built using the pre-coding approach with a secure lattice-based encryption scheme.

Theorem 1. *: Let $\Pi_{\text{lbe}} = (\mathsf{Gen}_{\text{lbe}}, \mathsf{Enc}_{\text{lbe}}, \mathsf{Dec}_{\text{lbe}})$ be a secure lattice-based encryption scheme, then the proposed scheme is CCA2-secure in the standard model.*

In the proof of security, we exploit the fact that for a well-formed ciphertext, we can recover the message if we know the randomness r that was used to create the ciphertext.

Proof: Suppose that $C^* = (C_1^*, C_2^*)$ be the challenge ciphertext. Let S_i be the event that the adversary $\mathcal{A}$ wins in Game i. Here is the sequence of games.

Game 0. We define Game 0 which is an interactive computation between an *adversary* $\mathcal{A}$ and a *simulator*. This game is usual CCA2 game used to define CCA2-security, in which the simulator provides the adversary's environment.
Initially, the simulator runs the key generation algorithm and gives the public-key to the adversary. The adversary submits two messages m_0, m_1 with $|m_0| = |m_1|$ to the simulator. The simulator chooses $b \in \{0, 1\}$ at random, and encrypts m_b, obtaining the challenge ciphertext $C^* = (C_1^*, C_2^*)$. The simulator gives C^* to the adversary. We denote by r^*, $h^* = \mathsf{wt}(r^*)$, $v^* = \lceil n/h^* \rceil$, s^* and $y^* = \mathsf{DV}(m'^*)$ where

$$m'^* = \mathsf{Encode}(m_b, r^*, s^*) \tag{2}$$

the corresponding intermediate quantities computed by the encryption algorithm, where DV means the decimal value. The only restriction on the adversary's requests is that after it makes a challenge request, the subsequent decryption requests must not be the same as the challenge ciphertext. At the end of the game, the adversary $\mathcal{A}$ outputs $\tilde{b} \in \{0, 1\}$. Let S_0 be the event that $\tilde{b} = b$. Since Game 0 is identical to the CCA2 game we have that

$$\left|\Pr[S_0] - \frac{1}{2}\right| = \mathsf{Adv}^{cca2}_{\Pi,\mathcal{A}}(k).$$

and, our goal is to prove that this quantity is negligible.

Game 1. Define Game 1 as identical with Game 0, except that $h = h^*$.
Lemma 1. *There exists an efficient adversary $\mathcal{A}_1$ such that:*

$$|\Pr[S_1] - \Pr[S_0]| \leq \mathsf{Adv}^{\mathrm{lbe}}_{\Pi,\mathcal{A}_1}(k). \tag{3}$$

By the assumption that the lattice-based encryption scheme is secure, we have that $\mathsf{Adv}^{\mathrm{lbe}}_{\Pi,\mathcal{A}_1}(k)$ is negligible.

Proof: Let $\mathsf{negl}(k) = |\Pr[S_1] - \Pr[S_0]|$. We can easily build an adversary $\mathcal{A}_1$ who hopes to recover m_b from Game 1. In this game, the adversary $\mathcal{A}_1$ queries on input$(C_1, C_2) \neq (C_1^*, C_2^*)$, while $h = h^*$. The simulator takes as input (C_1, C_2), $h = h^*$ and computes $r = \mathsf{Dec}_{\mathrm{lbe}}(C_2, \cdot) \neq r^*$, $y = C_1/h^* \neq y^*$ and so $m^{'} \neq m'^*$. If $|m^{'}|$ is not equal to obvious value $h^* \cdot \lceil n/h^* \rceil$, then the simulator rejects the ciphertext in (1). Since $m' \neq m'^*$, thus $s = (|m'| - h^* \cdot v)/(k - h^*) \neq s^*$ and the simulator rejects the ciphertext if s is not an integers. Furthermore, since the *position* of the message/ROB blocks ($r \neq r^*$) and the ROB blocks *length* s are not explicit, so, the output of algorithm (3.2)) is not identical to m_b. Therefore, if the lattice-based encryption scheme is secure (i.e., the adversary cannot recover r^* from it), then the $\mathcal{A}_1$'s advantage of this game is exactly equal to $\mathsf{negl}(k)$. By definition of $\mathsf{Adv}^{\mathrm{lbe}}_{\Pi,\mathcal{A}_1}(k)$, we have $\mathsf{negl}(k) \leq \mathsf{Adv}^{\mathrm{lbe}}_{\Pi,\mathcal{A}_1}(k)$.

Remark 1. *Notice that if one of the message extractor algorithm (3.2) inputs (i.e., r^*, v^*, s^* and m'^*) is not a legitimate input, then the output of its is not identical to m_b.*
Remark 2. *Notice that in order to query from the simulator, the CCA2 adversary cannot modified C_2 based on the challenge ciphertext C_2^* (well-formed decryption queries). Since for correctly retrieve m_b, the simulator must know the exact value of randomness r^*. So, if the lattice-based encryption scheme is secure, then the advantage of the CCA2 adversary is negligible.*

Game 2. Define Game 2 as identical with Game 1, except that $C_1 = C_1^*$.
Lemma 2. *There exists an efficient adversary $\mathcal{A}_2$ such that:*

$$|\Pr[S_2] - \Pr[S_1]| \leq \mathrm{Adv}^{\mathrm{lbe}}_{\Pi,\mathcal{A}_2}(k) \tag{4}$$

By the assumption that the lattice-based encryption scheme is secure, we have that $\mathsf{Adv}^{\mathrm{lbe}}_{\mathcal{A}_2}(k)$ is negligible.

Proof: Let $\mathsf{negl}(k) = |\Pr[S_2] - \Pr[S_0]|$. Consider the adversary $\mathcal{A}_2$ who aims to recover m_b from this game. In this game, the adversary $\mathcal{A}_2$ uniformly chooses $C_2 \neq C_2^*$ at random and queries on input $C = (C_1^*, C_2)$, $h = h^*$. In this case, the decryption simulator computes $r = \mathsf{Dec}_{\mathrm{lbe}}(C_2, \cdot) \neq r^*$. It also computes

$y = C_1/h = y^*$, $v = v^*$, $s = s^*$. Although the message/ROB blocks *length* and the encoded message m' are explicit, but since the *position* of the message/ROB blocks are not explicit, $r \neq r^*$, thus the outputs of algorithm (3.2)) is not identical to m_b. So, if the lattice-based encryption scheme is secure, then the $\mathcal{A}_2$'s advantage of this game is equal to $\mathsf{negl}(k)$. By definition of $\mathsf{Adv}^{\mathrm{lbe}}_{\Pi,\mathcal{A}_2}(k)$, we have $\mathsf{negl}(k) \leq \mathsf{Adv}^{\mathrm{lbe}}_{\Pi,\mathcal{A}_2}(k)$.

Game 3. Define Game 3 as identical with Game 0, except that $C_2 = C_2^*$.
Lemma 3. *There exists an efficient adversary $\mathcal{A}_3$ such that*

$$|\Pr[S_3] - \Pr[S_0]| \leq \mathsf{Adv}_{\Pi,\mathcal{A}_3}(k). \tag{5}$$

Proof: Suppose $\mathsf{negl}(k) = |\Pr[S_3] - \Pr[S_0]|$. We can easily build an adversary $\mathcal{A}_3$ who wishes to recover m_b from Game 3. In this game, the adversary $\mathcal{A}_3$ uniformly chooses $C_1 \neq C_1^*$ at random and queries on input (C_1, C_2^*). In this case, the simulator computes $r = \mathsf{Dec}_{\mathrm{lbe}}(C_2^*, \cdot) = r^*$, $h = h^*$, $y = C_1/h^* \neq y^*$ and so $m' \neq m'^*$. If $\mathsf{Len}(y) = |m'|$ is not equal to obvious value $h^* \cdot \lceil n/h^* \rceil$, then the simulator rejects the ciphertext in (1). Since $m' \neq m'^*$, thus $s = (|m'| - h^* \cdot v)/(k - h^*) \neq s^*$, and the simulator rejects the ciphertext if s is not an integers. Furthermore, since the ROB blocks *length* s and the encoded message m' are not explicit, thus the outputs of algorithm (3.2)) is not identical to m_b and so, the $\mathcal{A}_3$'s advantage of this game is equal to $\mathsf{negl}(k)$. By definition of $\mathsf{Adv}^{\mathrm{lbe}}_{\Pi,\mathcal{A}_3}(k)$, we have $\mathsf{negl}(k) \leq \mathsf{Adv}^{\mathrm{lbe}}_{\Pi,\mathcal{A}_3}(k)$.

Lemma 4. *We claim that*

$$|\Pr[S_3]| = 1/2. \tag{6}$$

Proof: Game 3 same as Game 0, except that the component C_1 of the queried ciphertext $C = (C_1, C_2^*)$ is not computed by equation (2) but rather chosen uniformly at random. So, the queried ciphertext C is statistically independent from the challenge bit b. Thus, the $\mathcal{A}_3$'s advantage in Game 3 is obviously 0, and

$$|\Pr[S_2]| = \frac{1}{2}$$

Completing the Proof:

We can write

$$\begin{aligned} |\Pr[S_0]| =& |\Pr[S_0] + \Pr[S_0] - \Pr[S_0] + \Pr[S_1] - \Pr[S_1] + \Pr[S_2] - \Pr[S_2] + \\ & \Pr[S_3] - \Pr[S_3]| \end{aligned}$$

So we have

$$\begin{aligned} |\Pr[S_0]| \leq & |\Pr[S_3]| + |\Pr[S_3] - \Pr[S_0]| + |\Pr[S_2] - \Pr[S_1]| + |\Pr[S_1] - \Pr[S_0]| + \\ & |\Pr[S_2] - \Pr[S_0]| \end{aligned}$$

We have

$$|\Pr[S_2] - \Pr[S_0]| \leq |\Pr[S_2] - \Pr[S_1]| + |\Pr[S_1] - \Pr[S_0]| \tag{7}$$

From equations (3,4,5,6,7) we have:

$$|\Pr[S_0] - 1/2| \leq \mathsf{Adv}_{\Pi,\,\mathcal{A}_3}(k) + 2\mathrm{Adv}^{\mathsf{lbe}}_{\Pi,\,\mathcal{A}_2}(k) + 2\mathrm{Adv}^{\mathsf{lbe}}_{\Pi,\,\mathcal{A}_1}(k)$$

By assumption, the right-hand side of the above equation is negligible, which finishes the proof.

4.2 Performance analysis

The performance-related issues can be discussed with respect to the computational complexity of key generation, key sizes, encryption and decryption speed, and information rate. The proposed cryptosystem features fast encryption and decryption. The time for computing encoded message is negligible compared to the time for computing $(\mathsf{Enc}_{\mathrm{lbe}}, \mathsf{Dec}_{\mathrm{lbe}})$. Encryption roughly needs one application of $\mathsf{Enc}_{\mathrm{lbe}}$ together a multiplication, and decryption roughly needs one application of $\mathsf{Dec}_{\mathrm{lbe}}$ together a division. The public/secret keys are as in the original scheme. The length of the ciphertext is equal to $n + (k - h)s + k$. The information rate (i.e., the ratio of the binary length of plaintext to that of the ciphertext) is equal to $n/(n + (k - h)s + k)$, and for $n \gg k$ and small integer s, it is close to one. Compared to other CCA2-secure lattice-based schemes were introduced today, our scheme is very simple and more efficient.

5 Conclusion

We construct the first direct CCA2-secure variant of the lattice-based PKE scheme, in a *black-box* manner, with a provable security from worst-case hardness problems in (ideal) lattices. This novel approach is very simple and more efficient and leads to the elimination of using SU-OT-SSs or MACs for transformations CPA-secure schemes to a CCA2-secure one. We showed that this scheme has extra advantages, namely, its IND-CCA security remains tightly related (in the standard model) to the worst-case hardness problems in lattice. Additionally, this scheme can be used for encryption of long messages without employing the hybrid encryption method and symmetric encryption.

Acknowledgment

Suggestions and comments are welcome.

References

1. S. Agrawal, D. Boneh, and X. Boyen. Efficient Lattice (H) IBE in the Standard Model. In *EUROCRYPT'2010*, LNCS, Vol. 6110, pp.553-572, 2010.

2. S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *CRYPTO'2010,*, LNCS, Vol. 6223, pp. 98-115, 2010.
3. S. Agrawal, X. Boyeny, V. Vaikuntanathanz, P. Voulgarisx and H. Wee. Fuzzy Identity Based Encryption from Lattices. Cryptology ePrint Archive, Report 2011/414, 2011. `http://eprint.iacr.org/2011/414`
4. M. Bellare, E. Kiltz, C. Peikert and B. Waters. Identity-Based (Lossy) Trapdoor Functions and Applications. In *EUROCRYPT'2012*, LNCS, Vol. 7237, pp. 228-245, 2012.
5. D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *SIAM J. Comput.*, Vol.36(5), pp. 1301-1328, 2007.
6. R. Canetti, S. Halevi, and J Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *Eurocrypt'2004*, LNCS, Vol. 3027, pp. 207-222, 2004.
7. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Journal of Cryptology*, Vol. 25(4), pp 601-639, 2012.
8. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *SIAM J. Comput.*, Vol. 30(2), pp.391437, 2000.
9. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: a ring based public key cryptosystem. In *ANTS-III*, LNCS, Vol. 1423, pp. 267-288, 1998.
10. D. Micciancio and C. Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT'2012*, LNCS, Vol. 7237, pp 700-718, 2012.
11. C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC'2008*, pp. 187-196, 2008.
12. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC'2009,* pp. 333-342, 2009.
13. C. Rackoff and D. Simon. Noninteractive Zero-knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *CRYPTO '91*, LNCS, Vol. 576, pp. 433-444, 1992.
14. M. Rückert. Adaptively secure identity-based identification from lattices without random oracles. In *SCN'2010*, LNCS, Vol.6280, pp. 345-362, 2010.
15. A. Rosen and G. Segev. Chosen-Ciphertext Security via Correlated Products. In *TCC'2009*, LNCS, Vol. 5444, pp. 419-436, 2009.
16. J. H. Seo and K. Emura. Revocable Identity-Based Encryption Revisited: Security Model and Construction. In *PKC'2013*, LNCS, Vol.7778, pp. 216-234, 2013.
17. W. P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In *SIAM J. Comput.*, Vol. 26, pp. 1484-1509, 1997.
18. R. Steinfeld, S. Ling, J. Pieprzyk, C. Tartary, H. Wang. NTRUCCA: How to Strengthen NTRUEncrypt to Chosen-Ciphertext Security in the Standard Model. In *PKC'2012*, LNCS, Vol. 7293, pp. 353-371, 2012.**??**
19. K. Xagawa. Improved (Hierarchical) Inner Product Encryption from Lattices. In *PKC'2013*, LNCS, Vol.7778, pp. 235-252, 2013.