Tightly Secure Ring-LWE Based Key Encapsulation with Short Ciphertexts

Martin R. Albrecht[1], Emmanuela Orsini[2], Kenneth G. Paterson[1], Guy Peer[3], and Nigel P. Smart[2]

[1] Royal Holloway, University of London
[2] University of Bristol
[3] Dyadic Security

Abstract. We provide a tight security proof for an IND-CCA Ring-LWE based Key Encapsulation Mechanism that is derived from a generic construction of Dent (IMA Cryptography and Coding, 2003). Such a tight reduction is not known for the generic construction. The resulting scheme has shorter ciphertexts than can be achieved with other generic constructions of Dent or by using the well-known Fujisaki-Okamoto constructions (PKC 1999, Crypto 1999). Our tight security proof is obtained by reducing to the security of the underlying Ring-LWE problem, avoiding an intermediate reduction to a CPA-secure encryption scheme. The proof technique maybe of interest for other schemes based on LWE and Ring-LWE.

1 Introduction

The possible advent of a quantum computer would immediately render insecure the vast majority of currently deployed public key cryptography. Hence, over the last few years, there has been considerably effort in trying to establish new public key encryption and signature schemes which are presumably resistant to the threat of quantum computers. Indeed, the US standards body NIST last year launched a Post Quantum Crypto (PQC) Project and published a call for submissions of quantum-resistant public-key cryptographic algorithms [NIS17].

Among the leading candidates for post-quantum public key encryption (PKE) schemes are those based on the Learning with Errors (LWE) problem and its ring equivalent (Ring-LWE). Starting with the seminal work of Regev [Reg05], there has been considerable work on various aspects of designing public key encryption schemes based on LWE and Ring-LWE [LPR13, CMV+15], research into implementation aspects [LSR+15, RRVV15, RVM+14], research into attacks [LP11, BG14, KF15, DTV15, KMW16, DB16], and various applications to advanced cryptographic constructions such as Somewhat Homomorphic Encryption [BV11b, BV11a].

Much existing work has, however, concentrated on producing encryption schemes meeting only a basic level of security, namely IND-CPA security. The development of schemes achieving the much strong IND-CCA security notion has received less attention. Of course, given an IND-CPA scheme, we can apply

a standard off-the-shelf transform to obtain an IND-CCA scheme. For example, the Fujisaki–Okamoto transform in [FO99a] constructs an IND-CCA secure public-key encryption scheme (PKE) from an IND-CPA (or even one-way secure) secure PKE, if it is also γ-uniform (see Definition 2). This reduction is tight but comes at the cost of also encrypting, under the IND-CPA PKE, the concaternation of the message and a random seed of λ bits, where λ is the security parameter.[4]

Since public key encryption is not well-suited to the transmission of long messages, public key encryption is often used to transmit a symmetric key, which is then used in a one-time-secure Authenticated Encryption (AE) scheme to encrypt the actual message. This methodology is often called the KEM-DEM paradigm [CS03]. It only requires the construction of a key encapsulation mechanism (KEM) rather than a full PKE scheme, and this is usually somewhat easier or leads to more efficient solutions than designing or repurposing a PKE scheme. It turns out that there are general constructions for obtaining IND-CCA secure KEMs from weaker primitives.

In the context of producing a KEM, the Fujisaki–Okamoto transform can be applied by setting the "primary message" to be the random KEM key of size λ bits. Thus one obtains a total message size of $2\,\lambda$ bits to encrypt under the IND-CPA encryption scheme. However, in LWE schemes the underlying message size directly impacts on the overall ciphertext size and the additional λ bits of random seed produce a ciphertext expansion of more than λ bits.

Dent [Den03] provides a veritable smorgasbord of techniques for constructing KEMs from weakly secure PKE schemes, giving five constructions of IND-CCA secure KEMs in total. His constructions in Tables 1–3 require strong properties from an underlying IND-CPA secure PKE scheme. The construction in Table 4 of [Den03] requires OW-CPA security for a starting *deterministic* PKE scheme. This transformation is attractive, since the reduction given in [Den03, Theorem 8] is tight. On the other hand, ciphertexts are slightly expanded compared to the starting scheme, since they require the inclusion of an extra hash value (whose size must be at least twice the security parameter). It is possible to de-randomise any IND-CPA secure PKE scheme having large message space to achieve OW-CPA security, e.g. by setting the randomness r used during encryption as $r = H(m)$ for some random oracle $H(\cdot)$. The proof is a simple exercise. Thus Dent's Table 4 construction can be used with an LWE-style PKE scheme as a starting point, though again with a cost of some ciphertext expansion.

The construction in Table 5 of [Den03] and analysed in Theorems 5 and 9 for building IND-CCA secure KEMs is of more interest to us. The construction starts with an OW-CPA secure scheme, but a probabilistic one, and does not introduce any ciphertext overhead. On the other hand, it has a non-tight reduction: the security bound degrades by a factor $q_D + q_H + q_K$ where q_D is the number of decryption oracle queries and q_K resp. q_H is the number of key derivation resp. hash function queries (both modelled as a random oracle).

[4] In a post-quantum scheme the reader should have $\lambda = 256$ in mind.

In the spirit of a KEM-DEM construction is a second generic transform of Fujisaki and Okamoto, given in [FO99b, FO13]. This yields a hybrid encryption scheme, but it is not in the true KEM-DEM paradigm (since the KEM part depends on the message m). The underlying symmetric cipher need not be an AE scheme, but can simply be a one-time pad encryption of the message and the message is used to produce the required randomness for the KEM-like part. The method of [FO99b, FO13] has two advantages over [FO99a]: firstly a one-time pad is more space efficient than an AE scheme; secondly the public key component does not suffer from the ciphertext expansion noted above for LWE based schemes. However, these benefits come at a cost, because the associated security reduction is not tight. In particular, the security bound degrades by a factor of q_H, the number of queries made to a hash function H, modelled as a random oracle. We note that a tight reduction can be achieved [GMMV03], either by making stronger assumptions about the underlying primitives or when the underlying primitive permits plaintext checking.

Having a tight security reduction is a very desirable property in practice-oriented cryptographic primitives. Essentially, the tightness of a reduction determines the strength of the security guarantees provided by the security proof; in concrete security terms, a tight reduction shows that an algorithm breaking the security of the scheme can be used to solve an assumed-to-be-hard problem without any significant increase in the running time or loss in success probability. A tight proof thus ensures that breaking the scheme (within the respective adversarial model) is at least as hard as breaking the alleged hard computational problem. On the other hand, a non-tight reduction can only provide much weaker guarantees, giving rise to the argument that the primitive should be instantiated with larger security parameters in order to account for the non-tightness of the proof.

This discussion and the preceding analysis of Dent's constructions raises the natural question: is it possible to build an IND-CCA secure KEM from simpler primitives with a tight security reduction, and without introducing any ciphertext overhead beyond that of the DEM? In this paper, we provide a positive solution to this question.

To answer the question, we produce a new security analysis for Dent's second construction (as shown in [Den03, Table 5]) in Section 3. The analysis applies to the case where the underlying OW-CPA scheme is instantiated using a specific construction based on lattices associated to polynomial rings, and which is secure under a natural variant of the Ring-LWE assumption. We name the resulting IND-CCA secure KEM as LIMA (for LattIce MAthematics), cf. Section 2 for details. In contrast to the generic case handled in [Den03], our security reduction for the specific scheme is tight. This is possible because of some weakly homomorphic properties enjoyed by the underlying encryption scheme. Because it is based on applying Dent's second construction to a simpler scheme, LIMA has no ciphertext overhead beyond that simpler scheme. Thus, we find that tightness can be maintained, whilst still using a generic construction which at first sight

appears to be non-tight. Given the increased interest in LWE-based encryption our proof technique may be of interest in other schemes.

We overview the construction of LIMA here. We start from standard Ring-LWE encryption going back to [LPR10], based on a polynomial ring of dimension N, reduced with respect to a modulus q. The encryption consists of an Ring-LWE sample, consisting of two ring elements c_0, c_1, and thus has ciphertexts of bitsize $2 \cdot N \cdot \lceil \log_2 q \rceil$. For reference, the reader may think of $N = 1024$ and $\lceil \log_2 q \rceil = 17$. Assuming one bit can be encoded per polynomial coefficient, this size can be reduced to $N \cdot \lceil \log_2 q \rceil + \ell \cdot \lceil \log_2 q \rceil$ for ℓ-bit messages by truncating c_0. Thus, to transport a λ-bit key, a minimum of $(N + \lambda) \cdot \lceil \log_2 q \rceil$ bits of ciphertext need to be sent.[5]

In Table 1, we compare the tightness and ciphertext expansion of the various constructions mentioned above, as well as in this work. We let $|\mathsf{AE}(m)|$ denote the ciphertext size of a one-time AE encryption of a message m, which is roughly $|m| + \lambda'$ where λ' is the space needed for a post-quantum secure authentication code. For the [FO99a] scheme we assume that $|m|$ is too large to be encrypted directly under the transform, and thus the scheme needs to be used in a hybrid format.

Construction	Ciphertext Size	Tightness
[FO99a]	$(N + 2 \cdot \lambda) \cdot \lceil \log_2 q \rceil + \lvert\mathsf{AE}(m)\rvert$	$\varepsilon + \ldots$
[FO99b, FO13]	$(N + \lambda) \cdot \lceil \log_2 q \rceil + \lvert m \rvert$	$q_H \cdot \varepsilon$
[Den03, Table 4]	$(N + \lambda) \cdot \lceil \log_2 q \rceil + 2 \cdot \lambda + \lvert\mathsf{AE}(m)\rvert$	$\varepsilon + \ldots$
[Den03, Table 5]	$(N + \lambda) \cdot \lceil \log_2 q \rceil + \lvert\mathsf{AE}(m)\rvert$	$(q_D + q_H + q_K) \cdot \varepsilon$
This work (non-generic)	$(N + \lambda) \cdot \lceil \log_2 q \rceil + \lvert\mathsf{AE}(m)\rvert$	$\varepsilon + \ldots$

Table 1. Ring-LWE ciphertext sizes for various IND-CCA transforms.

Note that our security analysis, like all the prior mentioned works, is in the Random Oracle Model (ROM). To fully assess post-quantum security, one should instead analyse security in the Quantum ROM (QROM), as introduced in [BDF+11]. In this model, an adversary can make superposition queries to the Random Oracle, possibly giving it much greater power, and invalidating certain classical ROM proof techniques. One way to achieve QROM security for PKE and KEMs is to add extra hash values to ciphertexts, cf. [TU16] which does this in the context of the FO transform. This of course increases the ciphertext size and, currently, results in non-tight reductions. It is an important open question whether one can achieve QROM security for a Dent-like KEM construction with a tight reduction and without suffering any ciphertext overhead.

Finally, achieving IND-CCA security also requires handling decryption errors of genuine encryptions. In Ring-LWE systems a validly generated ciphertext *may* not decrypt correctly if the initial "error term" used to generate the ciphertext

[5] A few more bits can be saved by suppressing the least significant bits of c_1.

is so large that it produces a wrap-around with respect to the modulus q. There are two ways around this issue; either select q so large that the probability of this occuring is vanishingly small, i.e. $2^{-\lambda}$, or by truncating the distribution used to produce the error term. We note, though, that these two modifications are orthogonal to the refined security proof of Dent's construction given in this work, since in Dent's construction the decryption algorithm actually re-encrypts the ciphertext as part of its operation and so can detect whether such an issue occurs.

2 Ring-LWE Key Encapsulation

Our basic scheme is defined over a global ring $R = \mathbb{Z}[X]/(\Phi_m(X))$ for some cyclotomic polynomial $\Phi_m(X)$. We will let R_q denote the reduction of this ring modulo the integer q, i.e. $R_q = \mathbb{Z}_q[X]/(\Phi_m(X))$. We let $N = \phi(m)$ denote the degree of this ring. On the set $\mathbb{Z}_q$ we define the distribution χ_σ which selects an integer with probability approximated by a discrete Gaussian with standard deviation σ centred on 0. The parameters (N, q, σ) will heavily influence the security of the scheme, and so are functions of a security parameter λ. In this paper, we assume suitable choices of the parameters can be selected for given values of λ. As noted in the introduction, the reader may think of $N = 1024$ and $\lceil \log_2 q \rceil = 17$, while σ will be a small constant ≈ 3.2.

The distribution χ_σ can be extended to all of R_q by generating N values from χ_σ independently and then assigning these values to the coefficients of an element from R_q, in which case we write $a \leftarrow \chi_\sigma^N$. If we wish to select an element in R_q uniformly at random we will write $a \leftarrow R_q$. If we want to be precise about what random coins we use then we write $a \leftarrow_r R_q$.

To aid bandwidth efficiency we sometimes truncate a ring element to a vector of integers modulo q of smaller size. Given a ring element $a \in R_q$, representing the element

$$a = a_0 + a_1 \cdot X + \cdots + a_{N-1} \cdot X^{N-1}$$

we define, for $1 \leq T \leq N$,

$$\mathsf{Trunc}(a, T) = a_0 + a_1 \cdot X + \cdots + a_{T-1} \cdot X^{T-1}.$$

This is encoded, for transmission and storage, as the vector of T integers

$$a_0 \| a_1 \ldots \| a_{T-1}.$$

2.1 IND-CPA Secure PKE

To define our KEM we first define a basic PKE scheme which is only IND-CPA secure. We give this as a tuple of algorithms $(\mathsf{KeyGen}, \mathsf{Enc\text{-}CPA}, \mathsf{Dec\text{-}CPA})$.

KeyGen: Key generation proceeds as follows

1. $a \leftarrow R_q$.
2. $s \leftarrow \chi_\sigma^N$.
3. $e' \leftarrow \chi_\sigma^N$.
4. $b \leftarrow a \cdot s + e'$.
5. $\mathfrak{sk} \leftarrow s$.
6. $\mathfrak{pk} \leftarrow (a, b)$.
7. Return $(\mathfrak{pk}, \mathfrak{sk})$.

Enc-CPA($\mathbf{m}, \mathfrak{pk}, r$): The encryption mechanism takes as input the public key $\mathfrak{pk} = (a, b)$, a message $\mathbf{m} \in \{0,1\}^\ell$, and random coins r. We assume that $\ell = |\mathbf{m}| \leq N$. We map this bit string (interpreted as a bit-vector) to a ring element (with binary coefficients) via the function BV-2-RE($\mathbf{m}$), and perform the inverse mapping via a function RE-2-BV(μ). The function BV-2-RE takes a bit string of length ℓ and maps it to a polynomial whose first ℓ coefficients are the associated bits, and all other coefficients are zero. (Here we identify bit values with 0 and 1 mod q.)

1. $\mu \leftarrow$ BV-2-RE($\mathbf{m}$).
2. $v, e, d \leftarrow_r \chi_\sigma^N$.
3. $x \leftarrow d + \Delta_q \cdot \mu \pmod q$. (Here, $\Delta_q = \lfloor q/2 \rfloor$.)
4. $t \leftarrow b \cdot v + x$.
5. $c_0 \leftarrow \mathsf{Trunc}(t, \ell)$.
6. $c_1 \leftarrow a \cdot v + e$.
7. Return $\mathbf{c} = (c_0, c_1)$.

Note that c_0 is the ring element $b \cdot v + d + \Delta_q \cdot \mathbf{m}$ truncated to ℓ coefficients, thus the bit-size of a ciphertext is equal to approximately

$$(N + \ell) \cdot \log_2 q \approx (N + |\mathbf{m}|) \cdot \log_2 q.$$

Dec-CPA($\mathbf{c}, \mathfrak{sk}$): On input of a ciphertext $\mathbf{c} = (c_0, c_1)$, and a secret key $\mathfrak{sk} = s$ the decryption is performed as follows:

1. Define ℓ to be the length of c_0, i.e. the number of field elements used to represent c_0.
2. $v \leftarrow s \cdot c_1$.
3. $t \leftarrow \mathsf{Trunc}(s, \ell)$.
4. $f \leftarrow c_0 - t$.
5. Convert f into centered-representation. That is, let $f = (f_0, \ldots, f_{\ell-1})$ where each $f_i \in \mathbb{Z}_q$. For each i, if $0 \leq f_i \leq \frac{q-1}{2}$ then leave it unchanged. Else, if $\frac{q}{2} < f_i \leq q - 1$, then set $f_i \leftarrow f_i - q$ (over the integers).
6. $\mu \leftarrow \left| \left\lfloor \frac{2}{q} f \right\rceil \right|$ (i.e., round component-wise to the nearest integer and take the absolute value; the result will be a binary vector).
7. $\mathbf{m} \leftarrow$ RE-2-BV(μ).
8. Return $\mathbf{m}$.

We will prove that this PKE scheme is IND-CPA secure under an LWE-style assumption in Section 3.

2.2 IND-CCA Secure PKE

Before proceeding to define our KEM, we explain how to use the above IND-CPA-secure PKE scheme to obtain an IND-CCA secure PKE scheme using the Fujisaki—Okamoto transform of [FO99a]. This is for later comparison with our proposed IND-CCA secure KEM.

We take the tuple of algorithms $(\mathsf{KeyGen}, \mathsf{Enc\text{-}CPA}, \mathsf{Dec\text{-}CPA})$ and produce a new tuple (KeyGen, $\mathsf{Enc\text{-}CCA}$, $\mathsf{Dec\text{-}CCA}$). The key generation algorithm stays the same and we do not repeat it.

The original encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc\text{-}CPA}, \mathsf{Dec\text{-}CPA})$ can encrypt N-bit messages, while the IND-CCA scheme encrypts messages that are $N - \lambda$ bits in length. The encryption scheme makes use of a hash function H to produce the random coins r for the underlying IND-CPA secure scheme; we model H as a Random Oracle in the security analysis.

$\mathsf{Enc\text{-}CCA}(\mathbf{m}, \mathfrak{pk})$:

1. $s \leftarrow \{0,1\}^{\lambda}$.
2. $\mu \leftarrow \mathbf{m} \| s$.
3. $r \leftarrow H(\mu)$.
4. $(c_0, c_1) \leftarrow \mathsf{Enc\text{-}CPA}(\mu, \mathfrak{pk}, r)$.
5. Return $\mathbf{c} = (c_0, c_1)$.

$\mathsf{Dec\text{-}CCA}(\mathbf{c}, \mathfrak{sk})$:

1. $\mu \leftarrow \mathsf{Dec\text{-}CPA}(\mathbf{c}, \mathfrak{sk})$.
2. $\mathbf{m} \| s \leftarrow \mu$, where s is λ bits long.
3. $r \leftarrow H(\mu)$.
4. $\mathbf{c}' \leftarrow \mathsf{Enc\text{-}CPA}(\mu, \mathfrak{pk}, r)$.
5. If $\mathbf{c} \neq \mathbf{c}'$ then return $\perp$.
6. Return $\mathbf{m}$.

Note for this scheme the bit-size of a ciphertext is equal to approximately

$$(N + |\mathbf{m}| + \lambda) \cdot \log_2 q.$$

We provide a security theorem establishing the IND-CCA security of this PKE scheme in Section 3. This is based on the results of [FO99a].

2.3 LIMA: A CCA-Secure Key Encapsulation Mechanism

One could use the above encryption scheme directly as a KEM by simply using it to encrypt one-time $\ell \leq N - \lambda$ bit keys, with a resulting ciphertext size of $(N + \ell + \lambda) \cdot \log_2 q$ bits. However, the following scheme (which we call LIMA and which follows the generic construction methodology of [Den03, Table 5]), enables us to transmit a key with ℓ bits of entropy using a ciphertext of bit-size

$$(N + \ell) \cdot \log_2 q,$$

thus reducing by $\lambda \cdot \log_2 q$ the number of bits needed to represent a ciphertext. The method makes use not only of a random oracle H to produce the randomness needed for the encryption function, but also a key derivation function $K^{(\ell')}$ (also modelled as a random oracle) to produce the actual encapsulated key (which can be of any length ℓ'). Again the scheme is presented as a tuple of algorithms $\mathsf{LIMA} = (\mathsf{KeyGen}, \mathsf{Encap\text{-}CCA}, \mathsf{Decap\text{-}CCA})$ in which KeyGen is as for the basic encryption scheme above.

$\mathsf{Encap\text{-}CCA}(\ell, \ell', \mathfrak{pk})$: This takes as input a public key $\mathfrak{pk}$ and two bit lengths ℓ, ℓ', and outputs an encapsulation $\mathbf{c} = (c_0, c_1)$ and the key $\mathfrak{k} \in \{0,1\}^{\ell'}$ it encapsulates. The bit length ℓ controls the ciphertext size and the associated entropy in the output key $\mathfrak{k}$.

1. $x \leftarrow \{0,1\}^{\ell}$.
2. $r \leftarrow H(x)$.
3. $(c_0, c_1) \leftarrow \mathsf{Enc\text{-}CPA}(x, \mathfrak{pk}, r)$.
4. $\mathfrak{k} \leftarrow K^{(\ell')}(x)$.
5. Return $(\mathbf{c} = (c_0, c_1), \mathfrak{k})$.

$\mathsf{Decap\text{-}CCA}(\mathbf{c}, \mathfrak{sk})$: This takes as input a secret key key $\mathfrak{sk}$ and an encapsulation $\mathbf{c} = (c_0, c_1)$, and outputs the key $\mathfrak{k}$ it encapsulates.

1. $x \leftarrow \mathsf{Dec\text{-}CPA}(\mathbf{c}, \mathfrak{sk})$.
2. $r \leftarrow H(x)$.
3. $\mathbf{c}' \leftarrow \mathsf{Enc\text{-}CPA}(x, \mathfrak{pk}, r)$.
4. If $\mathbf{c} \neq \mathbf{c}'$ then return $\perp$.
5. $\mathfrak{k} \leftarrow K^{(\ell')}(x)$.
6. Return $\mathfrak{k}$.

The IND-CCA security of this KEM is established in the next section, with a tight reduction to an LWE-style hardness assumption.

3 Security Proofs

In this section we present the hard problem on which the security of our scheme LIMA rests, survey prior security results on the Fujisaki-Okamoto transform and Dent's construction, and finally present our tight proof of security for LIMA.

3.1 Hard Problems

We recall the definition of Ring-LWE problem in normal form [LPR10, MR09, ACPS09]. In the definition below we directly consider all elements in R_q instead of the appropriate dual and canonical spaces associated to with it.

Definition 1 (Ring-LWE). *Let χ_σ denote the distribution defined earlier. Consider the following experiment: a challenger picks $s \in \chi_\sigma^N \subset R_q$ and a bit $\beta \in \{0,1\}$. The adversary $\mathcal{A}$ is given an oracle which on empty input returns a pair $(a,b) \in R_q^2$, where if $\beta = 0$ the two elements are chosen uniformly at random, and if $\beta = 1$ the value a is chosen uniformly at random and b is selected such that $b = a \cdot s + e$ where $e \in \chi_\sigma^N \subset R_q$. At the end of the experiment the adversary outputs its guess β' as to the hidden bit β. For an adversary which makes n_Q calls to its oracle and running in time t, we define*

$$\mathsf{Adv}^{\mathsf{LWE}}(\mathcal{A}, n_Q, t) = 2 \cdot \Big| \Pr[\beta = \beta'] - \frac{1}{2} \Big|.$$

We conjecture that $\mathsf{Adv}^{\mathsf{LWE}}(\mathcal{A}, n_Q, t)$ is negligible for all adversaries.

Conjecture 1. For suitable choices of σ, N and q (which depend on the security parameter λ) we conjecture that $\epsilon = \mathsf{Adv}^{\mathsf{LWE}}(A, n_Q, t)$ is a negligible function in the security parameter λ. In particular, for all adversaries running in time t we have $t/\epsilon^2 \geq 2^\lambda$.

We note that in the conjecture above we normalize the running time by success probability as $1/\epsilon^2$ — instead of the more customary $1/\epsilon$ — because we are considering a decision problem.

3.2 Provable Security of the Basic Encryption Scheme

The IND-CPA security of our basic encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc\text{-}CPA}, \mathsf{Dec\text{-}CPA})$ is established in the following theorem.

Theorem 1. *In the random oracle model, if the LWE problem is hard, then the scheme* $(\mathsf{KeyGen}, \mathsf{Enc\text{-}CPA}, \mathsf{Dec\text{-}CPA})$ *is IND-CPA secure. In particular, if there is an adversary $\mathcal{A}$ against the IND-CPA security of* $(\mathsf{KeyGen}, \mathsf{Enc\text{-}CPA}, \mathsf{Dec\text{-}CPA})$ *in the random oracle model, then there are adversaries $\mathcal{B}$ and $\mathcal{A}'$ such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) \leq 2 \cdot \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{B}, 1, t) + 2 \cdot \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{A}', 2, t).$$

We provide a proof of this theorem in the appendix.

3.3 Provable Security of our IND-CCA Secure PKE scheme

Our construction of an IND-CCA secure encryption scheme uses the Fujisaki-Okamoto transform [FO99a] applied to our basic scheme. Before we can apply this transform, we first need to establish its γ-uniformity.

Definition 2 (γ-Uniformity). *Consider an IND-CPA encryption scheme given by the tuple of algorithms* $(\mathsf{KeyGen}, \mathsf{Enc\text{-}CPA}, \mathsf{Dec\text{-}CPA})$ *with* $\mathsf{Enc\text{-}CPA} : \mathcal{M} \times \mathcal{R} \longrightarrow \mathcal{C}$ *being the encryption function mapping messages and randomness to ciphertexts. Such a scheme is said to be γ uniform if for all public keys $\mathfrak{pk}$ output by* KeyGen*, all $m \in \mathcal{M}$ and all $c \in \mathcal{C}$ we have $\gamma(\mathfrak{pk}, m, c) \leq \gamma$, where*

$$\gamma(\mathfrak{pk}, m, c) = \Pr[r \in \mathcal{R} : c = \mathsf{Enc\text{-}CPA}(m, \mathfrak{pk}, r)].$$

The lemma below establishes that Ring-LWE-based encryption has low γ-uniformity.

Lemma 1. *Let* $(\mathsf{KeyGen}, \mathsf{Enc\text{-}CPA}, \mathsf{Dec\text{-}CPA})$ *with parameters* N, χ_σ, q *be the basic PKE scheme described in Section 2.1 and let* σ *such that* $\Pr[X = x \mid X \leftarrow_r \chi_\sigma] \leq 1/2$ *for any* x*, then this scheme is* γ*-uniform with* $\gamma \leq 2^{-N}$*.*

Proof. For simplicity, we consider the case of encryption without truncation, where we will prove a stronger bound. Our argument extends easily to the case of truncated ciphertexts. Recall that encryption c an be written as

$$\mathbf{c} = (c_0, c_1) = (b \cdot v + e, a \cdot v + d + \Delta_q \cdot \mu \pmod q).$$

Here μ is a deterministic encoding of the message $\mathbf{m}$. Recall also that $v, e, d \leftarrow_r \chi_\sigma^N$. We see that for fixed $\mathbf{m}$, and fixed $\mathbf{c} = (c_0, c_1)$, if v is also fixed, then d and e are determined (by solving a simple linear system of equations). Thus we can write (for a fixed public key) $d = f_1(v)$ and $e = f_2(v)$ for functions f_1, f_2 that depend on $\mathbf{m}$ and $\mathbf{c}$. Letting V, E, D denote random variables that are distributed as χ_σ^N, and letting $\mathbf{1}_g$ denote an indicator function for a predicate g, it follows that

$$\begin{aligned}
\gamma(\mathfrak{pk}, m, c) &= \Pr[(v, e, d) \leftarrow_r (\chi_\sigma^N)^3 : \mathbf{c} = \mathsf{Enc\text{-}CPA}(\mathbf{m}, \mathfrak{pk}, (v, e, d))] \\
&= \sum_{v,e,d} \mathbf{1}_{\mathbf{c}=\mathsf{Enc\text{-}CPA}(\mathbf{m},\mathfrak{pk},(v,e,d))} \cdot \Pr[(V, E, D) = (v, e, d)] \\
&= \sum_{v,e,d} \mathbf{1}_{\mathbf{c}=\mathsf{Enc\text{-}CPA}(\mathbf{m},\mathfrak{pk},(v,e,d))} \cdot \Pr[V = v] \cdot \Pr[E = e] \cdot \Pr[D = d] \\
&\leq 2^{-2N} \sum_{v,e,d} \mathbf{1}_{\mathbf{c}=\mathsf{Enc\text{-}CPA}(\mathbf{m},\mathfrak{pk},(v,e,d))} \cdot \Pr[V = v] \\
&= 2^{-2N} \sum_{v} \mathbf{1}_{\mathbf{c}=\mathsf{Enc\text{-}CPA}(\mathbf{m},\mathfrak{pk},(v,f_2(v),f_1(v)))} \cdot \Pr[V = v] \\
&\leq 2^{-2N} \sum_{v} 1 \cdot \Pr[V = v] \\
&= 2^{-2N}.
\end{aligned}$$

Here, we first used the independence of the random variables V, E, D to simplify, then the fact that if $X \sim \chi_\sigma^N$, then $\Pr[X = x] \leq 2^{-N}$ for any value x. After that, we used the fact that if v is fixed, then e and d are determined as functions of v to simplify the sum to one over a single variable v. Finally, we used the fact that the sum over a distribution's probabilities equals 1. □

Note that in our construction the condition $\forall x, \Pr[X = x \mid X \leftarrow_r \chi_\sigma] \leq 1/2$ is always satisfied by picking $\sigma > 1$. Also note that if we truncate c_0 to ℓ components then the above bound becomes $2^{-(N+\ell)}$ by considering d truncated to ℓ components directly as being sampled from χ_σ^ℓ.

Applying the main result (Theorem 3) of Fujisaki and Okamoto [FO99a], we obtain the following:[6]

[6] Using $k = N$ and $k_0 = 256$ in Theorem 3 of [FO99a].

Theorem 2. *Suppose that* (KeyGen, Enc-CPA, Dec-CPA) *is* (t', ϵ') *IND-CPA secure and* γ*-uniform. For any* q_H, q_D*, the scheme* (KeyGen, Enc-CCA, Dec-CCA)*, derived from (*KeyGen, Enc-CPA, Dec-CPA*) as in Section 2.2, is* (t, ϵ) *IND-CCA secure for any adversary making at most* q_H *queries to* H *(modelled as a random oracle) and at most* q_D *queries to the decryption oracle, where*

$$t = t' - q_H \cdot (T_{\mathsf{Enc}} + v \cdot N),$$
$$\epsilon = \epsilon' \cdot (1-\gamma)^{-q_D} + q_H \cdot 2^{-\lambda-1},$$

where T_{Enc} *is the running time of the encryption function and* v *is a constant.*

3.4 Provable Security of LIMA

As remarked earlier our KEM construction LIMA is obtained by applying the construction of Dent [Den03, Table 5]. This builds an IND-CCA secure KEM from a OW-CPA secure PKE scheme. By Theorem 1, we know that our underlying encryption scheme is IND-CPA secure. It also has large message space. It follows that it is OW-CPA secure. Directly applying the generic result [Den03, Theorem 5], we would obtain the following security theorem for LIMA.

Theorem 3. *Suppose there is an adversary* $\mathcal{A}$ *which breaks the IND-CCA security of LIMA in the random oracle model, with advantage* ϵ*, in time* t*, making at most* q_D *decapsulation queries,* q_H *queries to the random oracle implementing the PRG function and* q_K *queries to the random oracle implementing the KDF. Then there is an adversary* $\mathcal{B}$ *breaking the OW-CPA security of the underlying encryption scheme which runs in roughly the same time* t*, but which has advantage* ϵ' *where*

$$\epsilon' \geq \frac{1}{q_D + q_H + q_K} \cdot \left(\epsilon - \frac{q_D}{2^\ell} - \gamma \cdot q_D\right)$$

and ℓ *is the size of the message being encrypted in the underlying encryption scheme, i.e. the size of* x *in our construction.,*

The problem with this result is that it does not give a very tight reduction. We thus present a new tight proof of our construction, which is *not generic*, i.e. we make explicit use of the Ring-LWE based construction of the underlying encryption scheme.

Theorem 4. *In the random oracle model, if the LWE problem is hard then LIMA is an IND-CCA secure KEM. In particular if* $\mathcal{A}$ *is an adversary against the IND-CCA security of LIMA, then there are adversaries* $\mathcal{B}$ *and* $\mathcal{A}'$ *such that*

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}, t) \leq 2\gamma \cdot q_D + 2 \cdot \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{B}, 1, t) + 2 \cdot \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{A}', 2, t) + \frac{q_H + q_K}{2^\ell}$$

Proof. Consider the game $\mathbb{G}_0$, defined in Figure 1, defining IND-CCA security of our KEM construction. As this is run in the Random Oracle model we model

Game $\mathbb{G}_0$: IND-CCA Security of our KEM

1. $a \leftarrow R_q$
2. $s, e' \leftarrow \chi_\sigma$
3. $b \leftarrow a \cdot s + e'$.
4. $x \leftarrow \{0,1\}^\ell$
5. $(v, e, d) \leftarrow H(x)$.
6. $\mu \leftarrow \mathsf{BV\text{-}2\text{-}RE}(x)$.
7. $a' \leftarrow a \cdot v + e$.
8. $b' \leftarrow b \cdot v + d$.
9. $t \leftarrow b' + \Delta_q \cdot \mu$.
10. $c_0^* \leftarrow \mathsf{Trunc}(t, \ell)$.
11. $c_1^* \leftarrow a'$.
12. $\beta \leftarrow \{0,1\}$.
13. If $\beta = 0$ then $\mathfrak{k} \leftarrow \{0,1\}^{\ell'}$
14. Else $\mathfrak{k} \leftarrow K(x)$.
15. $\beta' \leftarrow \mathcal{A}((a,b),(c_0^*, c_1^*), \mathfrak{k})$.
    - If $\mathcal{A}$ calls it decapsulation oracle on a pair $\mathbf{c} = (c_0, c_1) \neq (c_0^*, c_1^*)$ then
      (a) $x' \leftarrow \mathsf{Dec\text{-}CPA}(\mathbf{c}', s)$.
      (b) $(v', e', d') \leftarrow H(x')$.
      (c) $\mu' \leftarrow \mathsf{BV\text{-}2\text{-}RE}(x')$.
      (d) $a'' \leftarrow a \cdot v' + e'$.
      (e) $b'' \leftarrow b \cdot v' + d'$.
      (f) $t' \leftarrow b'' + \Delta_q \cdot \mu'$.
      (g) $c_0' \leftarrow \mathsf{Trunc}(t, \ell)$.
      (h) $c_1' \leftarrow a''$.
      (i) If $\mathbf{c} \neq \mathbf{c}' = (c_0', c_1')$ then return $\perp$.
      (j) Return $\mathfrak{k}' \leftarrow K(x')$.
16. Output 1 if and only if $\beta = \beta'$.

Figure 1. Game $\mathbb{G}_0$: IND-CCA Security of our KEM

the PRG by a random oracle H, and the KDF by a random oracle K, each of which are maintained by the challenger as lists (H-List and K-List) of pairs of input/output values. We define the advantage in the usual way in this game

$$\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}) = 2 \cdot \left| \Pr[\beta = \beta'] - \frac{1}{2} \right| = 2 \cdot \left| \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_0] - \frac{1}{2} \right|.$$

We now make a game hop as follows. We replace the real decapsulation algorithm used in Game $\mathbb{G}_0$ to one which operates as in Figure 2. Note that as written the oracle takes time $O(q_H)$ to execute. However, by also storing the associated (c_0', c_1') in the H-List, we can obtain a logarithmic cost to evaluate the oracle. The game with this new decapsulation oracle is called $\mathbb{G}_1$. Clearly $\mathbb{G}_0$ and $\mathbb{G}_1$ are identical except when the adversary submits an encapsulation to the decapsulation oracle for which it has *not* queried the random oracle H on the underlying message x.

Decapsulation oracle in Game $\mathbb{G}_1$

1. For all tuples (x', v', e', d') on the H-List execute
   (a) $\mu' \leftarrow \mathsf{BV\text{-}2\text{-}RE}(x')$.
   (b) $a'' \leftarrow a \cdot v' + e'$.
   (c) $b'' \leftarrow b \cdot v' + d'$.
   (d) $t' \leftarrow b'' + \Delta_q \cdot \mu'$.
   (e) $c'_0 \leftarrow \mathsf{Trunc}(t, \ell)$.
   (f) $c'_1 \leftarrow a''$.
   (g) If $\mathbf{c} = \mathbf{c}' = (c'_0, c'_1)$ then return $\mathfrak{k}' \leftarrow K(x')$.
2. Return $\perp$.

Figure 2. Decapsulation oracle in Game $\mathbb{G}_1$

Let E denote the event that decapsulation of a ciphertext in Game $\mathbb{G}_0$ is correctly handled, but it is not correctly handled in Game $\mathbb{G}_1$. We have

$$\begin{aligned}\Pr[\mathcal{A} \text{ wins game } \mathbb{G}_0] &= \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_0 | E] \cdot \Pr[E] \\ &\quad + \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_0 | \neg E] \cdot \Pr[\neg E] \\ &\leq \Pr[E] + \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_0 | \neg E] \\ &\leq \gamma \cdot q_D + \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_1].\end{aligned}$$

Here we apply a union bound across each of the q_D decapsulation queries and use the fact that, for each decapsulation query, the probability of event E is bounded by γ, relating to the uniformity of the encryption scheme. This is because E occurs only if the value of x underlying the query $\mathbf{c}$ has not been queried to H, in which case the random value used to encrypt x is still uniformly random from the adversary's perspective; hence the probability that x actually encapsulates to $\mathbf{c}$ is bounded by γ.

We now make a game hop to the game in which instead of picking $b = a \cdot s + e'$ we select $b \in R_q$ uniformly at random. We call this game $\mathbb{G}_2$ and define it in Figure 3. If is then clear that if the adversary can distinguish playing $\mathbb{G}_1$ from $\mathbb{G}_2$ then it can solve the LWE problem. Thus we have, for some adversary $\mathcal{B}$,

$$\Big| \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_1] - \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2] \Big| = \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{B}, 1, t).$$

At this point in the proof of IND-CPA security for the basic PKE scheme we made a game hop to a game in which a' and b' are chosen uniformly at random, and then remarked that if the adversary can spot this hop then we can turn the adversary into an algorithm which attacks the LWE problem with two samples. The same direct approach cannot be used here, as the input to the random oracle H depends on the message. Thus an adversary could distinguish which game it is in, if it was able to recover the message x in some way.

Game $\mathbb{G}_2$

1. $a, b \leftarrow R_q$
2. $x \leftarrow \{0,1\}^{\ell}$
3. $(v, e, d) \leftarrow H(x)$.
4. $\mu \leftarrow \mathsf{BV\text{-}2\text{-}RE}(x)$.
5. $a' \leftarrow a \cdot v + e$.
6. $b' \leftarrow b \cdot v + d$.
7. $t \leftarrow b' + \Delta_q \cdot \mu$.
8. $c_0^* \leftarrow \mathsf{Trunc}(t, \ell)$.
9. $c_1^* \leftarrow a'$.
10. $\beta \leftarrow \{0,1\}$.
11. If $\beta = 0$ then $\mathfrak{k} \leftarrow \{0,1\}^{\ell'}$
12. Else $\mathfrak{k} \leftarrow K(x)$.
13. $\beta' \leftarrow \mathcal{A}((a,b),(c_0^*, c_1^*), \mathfrak{k})$.
    - If $\mathcal{A}$ calls it decapsulation oracle on a pair $\mathbf{c} = (c_0, c_1) \neq (c_0^*, c_1^*)$ then respond using the method from Game $\mathbb{G}_1$ above.
14. Output 1 if and only if $\beta = \beta'$.

Figure 3. Game $\mathbb{G}_2$

Instead of performing a game hop at this point we construct an adversary $\mathcal{A}'$, given in Figure 4, which uses the adversary $\mathcal{A}$ in game $\mathbb{G}_2$ to solve the same LWE problem. The algorithm $\mathcal{A}'$ is given as input (obtained via two calls to the LWE oracle) a tuple (a, b, a', b'), where a, b are chosen uniformly random in R_q, and is asked to distinguish whether (a', b') are also selected uniformly at random or whether $a' = a \cdot v + e$ and $b' = b \cdot v + d$ for some values $v, e, d \in \chi_\sigma$.

First note that the encapsulation which is passed to $\mathcal{A}$ by $\mathcal{A}'$ is not a valid encapsulation *of any key*, irrespective of what $\mathcal{A}'$'s input is. This is because, even if $\mathcal{A}'$'s input was a pair of LWE samples the randomness used to produce the samples did not come from applying H to the encoded message x.

Let F denote the event that the adversary $\mathcal{A}$ queries the random oracle H on the value x, and let G denote the event tht $\mathcal{A}$ queries the random oracle K on x. If neither F nor G occurs then $\mathcal{A}$ has no advantage in winning the Game $\mathbb{G}_2$, so we have

$$\begin{aligned}
&\Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2] \qquad (1)\\
&\quad = \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2 | F \vee G] \cdot \Pr[F \vee G \text{ in game } \mathbb{G}_2]\\
&\qquad + \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2 | \neg(F \vee G)] \cdot \Pr[\neg(F \vee G) \text{ in game } \mathbb{G}_2]\\
&\quad \leq \Pr[F \vee G \text{ in game } \mathbb{G}_2]\\
&\qquad + \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2 | \neg F \wedge \neg G \text{ in game } \mathbb{G}_2]\\
&\quad = \Pr[F \vee G \text{ in game } \mathbb{G}_2] + \frac{1}{2}. \qquad (2)
\end{aligned}$$

We examine the behaviour of $\mathcal{A}'$ when it is given the two different inputs.

Adversary $\mathcal{A}'$ breaking Lima-LWE

1. $x \leftarrow \{0,1\}^{\ell}$
2. $\mu \leftarrow \mathsf{BV\text{-}2\text{-}RE}(x)$.
3. $t \leftarrow b' + \Delta_q \cdot \mu$.
4. $c_0^* \leftarrow \mathsf{Trunc}(t, \ell)$.
5. $c_1^* \leftarrow a'$.
6. $\mathfrak{k} \leftarrow \{0,1\}^{\ell'}$
7. $\beta' \leftarrow \mathcal{A}((a,b),(c_0^*,c_1^*),\mathfrak{k})$.
   - If $\mathcal{A}$ calls it decapsulation oracle on a pair $\mathbf{c} = (c_0, c_1) \neq (c_0^*, c_1^*)$ then respond using the method from Game $\mathbb{G}_1$ above.
   - If $\mathcal{A}$ calls the random oracle H or the random oracle K on the value x then $\mathcal{A}'$ terminates and outputs 1, i.e. (a, b, a', b') is an LWE pair of samples.
8. If $\mathcal{A}$ terminates without making the random oracle calls above then $\mathcal{A}'$ outputs zero.

Figure 4. Adversary $\mathcal{A}'$ breaking Lima-LWE

– If the input to $\mathcal{A}'$ is a uniformly random tuple then the target encapsulation (c_0^*, c_1^*) contains no information about x. Thus the probability that F or G happens is essentially $(q_H + q_K) \cdot 2^{-\ell}$, where q_H is the number of queries to H made by $\mathcal{A}$ and q_K is the number of queries made to K. So we have

$$\Pr[\mathcal{A}' \text{ wins its game}| \text{ Input is random}] = \left(1 - \frac{q_H + q_K}{2^{\ell}}\right).$$

– If the input to $\mathcal{A}'$ is a pair of LWE samples then $\mathcal{A}$ is running in a perfect simulation of the game $\mathbb{G}_2$, until (and if) event F or G happens. If F or G happens then $\mathcal{A}'$ wins its game, otherwise $\mathcal{A}'$ loses its game. So we have

$$\Pr[\mathcal{A}' \text{ wins its game}| \text{ Input is an LWE sample}] = \Pr[F \vee G \text{ in game } \mathbb{G}_2].$$

Putting this all together we have

$$\begin{aligned}
&\Pr[\mathcal{A}' \text{ wins its game}] \\
&\quad = \Pr[\mathcal{A}' \text{ wins its game}| \text{ Input is random}] \cdot \Pr[\text{Input is random}] \\
&\qquad + \Pr[\mathcal{A}' \text{ wins its game}| \text{ Input is LWE sample}] \\
&\qquad\qquad \cdot \Pr[\text{ Input is LWE sample}] \\
&\quad = \left(1 - \frac{q_H + q_K}{2^{\ell}}\right) \cdot \frac{1}{2} + \Pr[F \vee G \text{ in game } \mathbb{G}_2] \cdot \frac{1}{2}
\end{aligned}$$

Now, combining this with equation 2 we obtain

$$\begin{aligned}\Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2] &\leq \Pr[F \vee G \text{ in game } \mathbb{G}_2] + \frac{1}{2} \\ &= 2 \cdot \Pr[\mathcal{A}' \text{ wins its game}] - \left(1 - \frac{q_H + q_K}{2^\ell}\right) + \frac{1}{2}\end{aligned}$$

Thus we have a bound on the total advantage of $\mathcal{A}$ in game $\mathbb{G}_0$ of

$$\begin{aligned}\mathsf{Adv}^{\mathsf{IND\text{-}CCA}}(\mathcal{A}, t) &\leq 2 \cdot \left| \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_0] - \frac{1}{2} \right| \\ &\leq 2 \cdot \left| \gamma \cdot q_D + \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_1] - \frac{1}{2} \right| \\ &= 2 \cdot \Big| \gamma \cdot q_D + \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_1] \\ &\qquad - \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2] + \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2] - \frac{1}{2} \Big| \\ &\leq 2\gamma \cdot q_D + 2 \cdot \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{B}, 1, t) \\ &\qquad + 2 \cdot \left| \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2] - \frac{1}{2} \right| \\ &\leq 2\gamma \cdot q_D + 2 \cdot \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{B}, 1, t) \\ &\qquad + 2 \cdot \left| 2 \cdot \Pr[\mathcal{A}' \text{ wins its game}] - 1 + \frac{q_H + q_K}{2^\ell} \right| \\ &\leq 2\gamma \cdot q_D + 2 \cdot \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{B}, 1, t) \\ &\qquad + 4 \cdot \left| \Pr[\mathcal{A}' \text{ wins its game}] - \frac{1}{2} \right| + \frac{q_H + q_K}{2^\ell} \\ &\leq 2\gamma \cdot q_D + 2 \cdot \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{B}, 1, t) + 2 \cdot \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{A}', 2, t) + \frac{q_H + q_K}{2^\ell}.\end{aligned}$$

This completes the proof of Theorem 4.

Acknowledgements

This work has been supported in part by ERC Advanced Grant ERC-2015-AdG-IMPaCT, and by EPSRC via grants EP/N021940/1, EP/M012824, EP/M013472/1, EP/L018543/1 and EP/P009417/1.

References

ACPS09. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, Heidelberg, August 2009.

BDF+11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011.

BG14. Shi Bai and Steven D. Galbraith. Lattice decoding attacks on binary LWE. In Willy Susilo and Yi Mu, editors, *ACISP 14*, volume 8544 of *LNCS*, pages 322–337. Springer, Heidelberg, July 2014.

BV11a. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011.

BV11b. Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 505–524. Springer, Heidelberg, August 2011.

CMV+15. Donald Donglong Chen, Nele Mentens, Frederik Vercauteren, Sujoy Sinha Roy, Ray C. C. Cheung, Derek Pao, and Ingrid Verbauwhede. High-speed polynomial multiplication architecture for Ring-LWE and SHE cryptosystems. *IEEE Trans. on Circuits and Systems*, 62-I(1):157–166, 2015.

CS03. Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

DB16. Chaohui Du and Guoqiang Bai. A family of scalable polynomial multiplier architectures for ring-LWE based cryptosystems. Cryptology ePrint Archive, Report 2016/323, 2016. `http://eprint.iacr.org/2016/323`.

Den03. Alexander W. Dent. A designer's guide to KEMs. In Kenneth G. Paterson, editor, *9th IMA International Conference on Cryptography and Coding*, volume 2898 of *LNCS*, pages 133–151. Springer, Heidelberg, December 2003.

DTV15. Alexandre Duc, Florian Tramèr, and Serge Vaudenay. Better algorithms for LWE and LWR. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 173–202. Springer, Heidelberg, April 2015.

FO99a. Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In Hideki Imai and Yuliang Zheng, editors, *PKC'99*, volume 1560 of *LNCS*, pages 53–68. Springer, Heidelberg, March 1999.

FO99b. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554. Springer, Heidelberg, August 1999.

FO13. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, January 2013.

GH15. Tim Güneysu and Helena Handschuh, editors. *CHES 2015*, volume 9293 of *LNCS*. Springer, Heidelberg, September 2015.

GMMV03. David Galindo, Sebastià Martín Molleví, Paz Morillo, and Jorge Luis Villar. Easy verifiable primitives and practical public key cryptosystems. In Colin Boyd and Wenbo Mao, editors, *ISC 2003*, volume 2851 of *LNCS*, pages 69–83. Springer, Heidelberg, October 2003.

KF15. Paul Kirchner and Pierre-Alain Fouque. An improved BKW algorithm for LWE with applications to cryptography and lattices. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 43–62. Springer, Heidelberg, August 2015.

KMW16. Elena Kirshanova, Alexander May, and Friedrich Wiemer. Parallel implementation of BDD enumeration for LWE. In Mark Manulis, Ahmad-Reza Sadeghi, and Steve Schneider, editors, *ACNS 16*, volume 9696 of *LNCS*, pages 580–591. Springer, Heidelberg, June 2016.

LP11. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 319–339. Springer, Heidelberg, February 2011.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May 2010.

LPR13. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 35–54. Springer, Heidelberg, May 2013.

LSR+15. Zhe Liu, Hwajeong Seo, Sujoy Sinha Roy, Johann Großschädl, Howon Kim, and Ingrid Verbauwhede. Efficient ring-LWE encryption on 8-bit AVR processors. In Güneysu and Handschuh [GH15], pages 663–682.

MR09. Daniele Micciancio and Oded Regev. Lattice-based cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 147–191, Berlin, Heidelberg, New York, 2009. Springer, Heidelberg.

NIS17. NIST National Institute for Standards and Technology. Post-quantum crypto project, 2017. `http://csrc.nist.gov/groups/ST/post-quantum-crypto/`.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

RRVV15. Oscar Reparaz, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. A masked ring-LWE implementation. In Güneysu and Handschuh [GH15], pages 683–702.

RVM+14. Sujoy Sinha Roy, Frederik Vercauteren, Nele Mentens, Donald Donglong Chen, and Ingrid Verbauwhede. Compact ring-LWE cryptoprocessor. In Lejla Batina and Matthew Robshaw, editors, *CHES 2014*, volume 8731 of *LNCS*, pages 371–391. Springer, Heidelberg, September 2014.

TU16. Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the fujisaki-okamoto and OAEP transforms. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 192–216. Springer, Heidelberg, October / November 2016.

A Proof of Theorem 1

Proof. This is essentially the "standard" proof of security for Ring-LWE based encryption [LPR10]. We proceed by a series of Games, where we model the PRG as a random oracle, and thus can select the randomness used from the required distributions at will without using the PRG. In Game $\mathbb{G}_0$, given in Figure 5, we are playing the usual IND-CPA game for our encryption scheme: We see that there are three Ring-LWE samples here, one given by $(a, b = a \cdot s + e')$ where e'

Game $\mathbb{G}_0$: IND-CPA Security of our Encryption Scheme

1. $a \leftarrow R_q$
2. $s, e' \leftarrow \chi_\sigma$
3. $b \leftarrow a \cdot s + e'$.
4. $(\mathbf{m}_0, \mathbf{m}_1, \mathsf{state}) \leftarrow \mathcal{A}_1(a, b)$, we insist that $|\mathbf{m}_0| = |\mathbf{m}_1| < N$.
5. $\beta \leftarrow \{0, 1\}$.
6. $v, e, d \leftarrow \chi_\sigma$.
7. $\mu \leftarrow \mathsf{BV\text{-}2\text{-}RE}(\mathbf{m}_b)$.
8. $a' \leftarrow a \cdot v + e$.
9. $b' \leftarrow b \cdot v + d$.
10. $t \leftarrow b' + \Delta_q \cdot \mu$.
11. $c_0 \leftarrow \mathsf{Trunc}(t, |\mathbf{m}_0|)$.
12. $c_1 \leftarrow a'$.
13. $\beta' \leftarrow \mathcal{A}_2(c_0, c_1, \mathsf{state})$.
14. Output one if and only if $\beta = \beta'$.

Figure 5. Game $\mathbb{G}_0$: IND-CPA Security of our Encryption Scheme

is selected from the distribution χ_σ, one given by $(a, a' = a \cdot v + e)$ and one given by $(b, b' = a \cdot v + e)$, where the last two are for the same secret v. It is clear that we have

$$\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) = 2 \cdot \left| \Pr[\beta = \beta'] - \frac{1}{2} \right| = 2 \cdot \left| \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_0] - \frac{1}{2} \right|.$$

We now make a game hop to the game, defined in Figure 6 in which instead of picking $b = a \cdot s + e'$ we select $b \in R_q$ uniformly at random. It is then clear that if the adversary can distinguish playing $\mathbb{G}_0$ from $\mathbb{G}_1$ then it can solve the Lima-LWE problem, thus we have, for some adversary $\mathcal{B}$,

$$\left| \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_0] - \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_1] \right| = \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{B}, 1, t).$$

We now make another game hop in which we replace a' and b' by elements selected uniformly at random. We call this Game $\mathbb{G}_2$ and give it in Figure 7. Again it is clear that if $\mathcal{A}$ can distinguish between playing $\mathbb{G}_1$ and $\mathbb{G}_2$ then it can solve the Lima-LWE problem. Thus we have, for some adversary $\mathcal{A}'$,

$$\left| \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_1] - \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2] \right| = \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{A}', 2, t).$$

Finally we see that in $\mathbb{G}_2$ the message (thought of as $\Delta_q \cdot \mu$) is encrypted via a one-time pad with the key b', which is chosen uniformly at random. Thus we have

$$\Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2] = \frac{1}{2}.$$

Game $\mathbb{G}_1$

1. $a, b \leftarrow R_q$
2. $(\mathbf{m}_0, \mathbf{m}_1, \mathsf{state}) \leftarrow \mathcal{A}_1(a, b)$, we insist that $|\mathbf{m}_0| = |\mathbf{m}_1| < N$.
3. $\beta \leftarrow \{0, 1\}$.
4. $v, e, d \leftarrow \chi_\sigma$.
5. $\mu \leftarrow \mathsf{BV\text{-}2\text{-}RE}(\mathbf{m}_b)$.
6. $a' \leftarrow a \cdot v + e$.
7. $b' \leftarrow b \cdot v + d$.
8. $t \leftarrow b' + \Delta_q \cdot \mu$.
9. $c_0 \leftarrow \mathsf{Trunc}(t, |\mathbf{m}_0|)$.
10. $c_1 \leftarrow a'$.
11. $\beta' \leftarrow \mathcal{A}_2(c_0, c_1, \mathsf{state})$.
12. Output one if and only if $\beta = \beta'$.

Figure 6. Game $\mathbb{G}_1$

Game $\mathbb{G}_2$

1. $a, b \leftarrow R_q$
2. $(\mathbf{m}_0, \mathbf{m}_1, \mathsf{state}) \leftarrow \mathcal{A}_1(a, b)$, we insist that $|\mathbf{m}_0| = |\mathbf{m}_1| < N$.
3. $\beta \leftarrow \{0, 1\}$.
4. $a', b' \leftarrow R_q$.
5. $\mu \leftarrow \mathsf{BV\text{-}2\text{-}RE}(\mathbf{m}_b)$.
6. $t \leftarrow b' + \Delta_q \cdot \mu$.
7. $c_0 \leftarrow \mathsf{Trunc}(t, |\mathbf{m}_0|)$.
8. $c_1 \leftarrow a'$.
9. $\beta' \leftarrow \mathcal{A}_2(c_0, c_1, \mathsf{state})$.
10. Output one if and only if $\beta = \beta'$.

Figure 7. Game $\mathbb{G}_2$

Putting all these results together we obtain

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{IND\text{-}CPA}}(\mathcal{A}) &= 2 \cdot \left| \Pr[\beta = \beta'] - \frac{1}{2} \right| \\
&= 2 \cdot \left| \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_0] - \frac{1}{2} \right| \\
&\leq 2 \cdot \left| \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_0] - \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_1] \right| \\
&\qquad + 2 \cdot \left| \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_1] - \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2] \right| \\
&\qquad + 2 \cdot \left| \Pr[\mathcal{A} \text{ wins game } \mathbb{G}_2] - \frac{1}{2} \right| \\
&\leq 2 \cdot \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{B}, 1, t) + 2 \cdot \mathsf{Adv}^{\mathsf{LWE}}(\mathcal{A}', 2, t).
\end{aligned}$$