100912024
Joshua Limbrey

Cryptanalysis of Lattice Based Post-Quantum Cryptosystems

Preliminary Literature Review
March 2022

Supervisor: Dr Rachel Player

Submitted as part of the requirements for the award of the
MSc in Information Security
at Royal Holloway, University of London.

Contents

1 List of Definitions, Notation, Abbreviations and Acronyms

Note: All security properties discussed will be for public-key encryption (PKE) schemes.

Def 1.1 (Indistinguishable)**.**

$$(pk, sk) \leftarrow \Sigma.KeyGen$$

An adversary $\mathcal{A}$ produces two messages $m_0, m_1 \in \mathcal{M}$ (of equal length). We choose a bit $b \in \{0, 1\}$.

$$c = Enc_{pk}(m_b)$$

Give the adversary (c, pk), and allow them to generate a bit $a \in \{0, 1\}$. If $a = b$, then the adversary has succeeded.
We say an encryption scheme Σ is indistinguishable if the following holds:

$$\mathbb{P}(\mathcal{A} \text{ succeeds}) = \frac{1}{2} + \varepsilon \qquad \text{where } \varepsilon \text{ is negligible.}$$

Intuitively, an encryption scheme has indistinguishability if an adversary is given a challenge ciphertext c, they cannot tell if it is from m_0 or m_1.

Def 1.2 (IND-CPA or CPA security)**.** "Indistinguishability of ciphertexts under chosen plaintext attack" for an encryption scheme Σ.

$$(pk, sk) \leftarrow \Sigma.KeyGen$$

The adversary $\mathcal{A}$ is given pk and outputs two messages m_0, m_1 (of equal length), and is also given a challenge ciphertext:

$$c = Enc_{pk}(m_b) \qquad \text{for a chosen } b \in \{0,1\}.$$

$\mathcal{A}$ now generates a bit $a \in \{0,1\}$, and if $a = b$ then $\mathcal{A}$ has succeeded.
The encryption scheme has CPA security if:

$$\mathbb{P}(\mathcal{A} \text{ succeeds}) = \frac{1}{2} + \varepsilon \qquad \text{where } \varepsilon \text{ is negligible.}$$

This can intuitively be thought of as if an attacker is given access to the public key (therefore able to encrypt plaintext's of their choice **but not decrypt**), then if given the encryption of one of two plaintexts, the attacker has negligible advantage over guessing.

Def 1.3 (IND-CCA or CCA security)**.** "Indistinguishability of ciphertexts under a chosen ciphertext attack" for an encryption scheme Σ.

$$(pk, sk) \leftarrow \Sigma.KeyGen$$

The adversary $\mathcal{A}$ is given pk and a decryption oracle $\mathcal{O}_{Dec_{sk}}$, and outputs m_0, m_1 (of equal length). The adversary is only able to query this oracle up until it receives the challenge ciphertext,

$$c = Enc_{pk}(m_b) \qquad \text{for a chosen } b \in \{0,1\}.$$

$\mathcal{A}$ then generates a bit $a \in \{0,1\}$, and if $a = b$ then $\mathcal{A}$ has succeeded. The encryption scheme has CCA security if:

$$\mathbb{P}(\mathcal{A} \text{ succeeds}) = \frac{1}{2} + \varepsilon \qquad \text{where } \varepsilon \text{ is negligible.}$$

Intuitively, this is if an adversary is able to ask for decryptions before given the challenge, once given the challenge ciphertext they have negligible advantage over guessing.

Def 1.4 (IND-CCA2 or CCA2 security)**.** "Indistinguishability of ciphertexts under an adaptive chosen ciphertext attack" for an encryption scheme Σ.

$$(pk, sk) \leftarrow \Sigma.KeyGen$$

The adversary $\mathcal{A}$ is given pk and a decryption oracle $\mathcal{O}_{Dec_{sk}}$, and outputs m_0, m_1 (of equal length). The adversary then receives the challenge ciphertext,

$$c = Enc_{pk}(m_b) \qquad \text{for a chosen } b \in \{0,1\}.$$

but may continue to query $\mathcal{O}_{Dec_{sk}}$ provided the requested decryption is not of c $\mathcal{A}$ then generates a bit $a \in \{0,1\}$, and if $a = b$ then $\mathcal{A}$ has succeeded. The encryption scheme has CCA security if:

$$\mathbb{P}(\mathcal{A} \text{ succeeds}) = \frac{1}{2} + \varepsilon \qquad \text{where } \varepsilon \text{ is negligible.}$$

Intuitively, this can be thought of as if an adversary has the ability to decrypt any ciphertext other than the challenge, can they decrypt the challenge.

Note: We shall use $\mathcal{B}$ to denote the set of 8-bit unsigned integers (or bytes), ie. the set $\{0, ..., 255\}$

2 Introduction

In this PLR, we shall discuss two lattice-based cryptosystems submitted as part of the National Institute of Standards and Technology (NIST) post-quantum call for proposals round 3. These are Saber[1], an indistinguishability of ciphertexts (IND)-chosen ciphertext attack (CCA) Mod-learning with rounding (LWR) based key encapsulation mechanism (KEM) and Kyber[3], an IND-adaptive chosen ciphertext attack (CCA2) Mod-learning with errors (LWE) based KEM. In this preliminary literature review we will lay the foundation for my full dissertation, ensuring that both myself and the reader fully understand the cryptosystems being discussed.

This will be broken down into:

1. Algorithm specification of Kyber.PKE
2. Demonstration of security properties
3. Transformation of Kyber.PKE to Kyber.KEM
4. Algorithm specification of Saber.PKE
5. Demonstration of security properties
6. Transformation of Saber.PKE to Saber.KEM

This will lay the foundation that will allow us to conduct cryptanalysis on both cryptosystems.

3 Understanding the LWE/LWR problem

KYBER.PKE is a module LWE based encryption scheme, and Saber.PKE a module LWR based encryption scheme; both relying on the hardness of the of the LWE/LWR problem(s) - both believed to be hard for both classical and quantum computers. Below is an informal mod q set-up for the LWE problem for a prime q:

1. Let us chose an n dimensional vector $\mathbf{s} \in \mathbb{F}_q^n$. This is our secret.
2. Let us randomly and uniformly generate an $m \times n$ matrix $\mathbf{A}$ over $\mathbb{F}_q$ from elements in $\mathbb{F}_q$.
3. Let us generate an m dimensional vector, $\mathbf{e}$, s.t. $\mathbf{e}_i \sim \chi \forall i \in 1, ..., m$ independently for the distribution χ on $\mathbb{F}_q$ centred on 0.
4. Let $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ over $\mathbb{F}_q$

Now, given $(\mathbf{A}, \mathbf{b})$, find $\mathbf{s}$. This problem can be reduced to solving the lattice problems shortest vector problem (SVP) or closest vector problem (CVP). First we consider the q-ary lattice

$$\mathcal{L}_{Im(\mathbf{A})} = \{y \in \mathbb{Z}^m | y = \mathbf{A}z \mod q \text{ for some } z \in \mathbb{Z}^n\}$$

which is generated by the column vectors of our matrix $\mathbf{A} \mod q$. The LWR problem instead uses deterministic rounding instead of adding the small random error $\mathbf{e}$[2].

4 Algorithm specification of Kyber.PKE

KYBER.PKE is defined over the ring $R \equiv \mathbb{Z}/(X^n + 1)$ and $R_q \equiv \mathbb{Z}_q[X]/(X^n + 1)$ where $n = 2^{n'-1}$ s.t. $X^n + 1$ is the $2^{n'\text{th}}$ cyclotomic polynomial[3]. In relation to our earlier explained LWE problem, R is $\mathbb{F}$ and R_q is $\mathbb{F}_q$. We begin, by generating our matrix $\mathbf{A}$ using the following algorithm:

Algorithm 1 Generate keys.

$N := 0$
$(\rho, \sigma) := G(d)$ ▷ Where G is a hash function s.t. $G : \mathcal{B}^* \to \mathcal{B}^{32} \times \mathcal{B}^{32}$
for $i \leftarrow 0, k-1$ **do**
  for $j \leftarrow 0, k-1$ **do**
    $\mathbf{A} := Parse(XOF(\rho, j, i))$ ▷ This essentially generates a random $k \times k$ matrix over R_q as ρ is pseudorandom from the hash function $G()$.
  end for
end for
for $i \leftarrow 0, k-1$ **do**
  $\mathbf{s} := CBD(PRF(\sigma, N))$ ▷ Where CBD is a function outputting a polynomial in R_q with the coefficients distributed central-binomially. PRF is a pseudorandom function, $PRF : \mathcal{B}^{32} \times \mathcal{B} \to \mathcal{B}^*$.
  $N := N + 1$
end for
for $i \leftarrow 0, k-1$ **do**
  $\mathbf{e} := CBD(PRF(\sigma, N))$
  $N := N + 1$
end for
$\mathbf{s} := NTT(\mathbf{s})$ ▷ Where NTT is a bijection mapping $f \in R_q$ to a polynomial with the coefficient vector.
$\mathbf{e} := NTT(\mathbf{e})$
$\mathbf{b} := \mathbf{As} + \mathbf{e}$
return $\mathbf{A}$, $\mathbf{s}$, $\mathbf{b}$, $\mathbf{e}$

From this, we have our public and private keys, $pk := (\mathbf{b} \mod q) \| \rho$ and $sk := \mathbf{s} \mod q$ (both encoded).

Algorithm 2 Encryption

Input: pk, $m \in \mathcal{B}^{32}$
first we must extract $\mathbf{A}$ and $mathbfb$ from pk.
$\rho := pk + 12 \cdot k \cdot \frac{n}{8}$ ▷ ρ was simply appended to the end of $\mathbf{b}$ so we can extract it simply. As we now have ρ we can re-construct $\mathbf{A}$ like we did in key generation.
for $i \leftarrow 0, k-1$ **do**
  for $j \leftarrow 0, k-1$ **do**
    $\mathbf{A}^T := Parse(XOF(\rho, i, j))$
  end for
end for
for $i \leftarrow 0, k-1$ **do**
  $\mathbf{r} := CBD(PRF(r, N))$ ▷ Where $r \in \mathcal{B}^{32}$ is a random coin.
  $N := N + 1$
end for
for $i \leftarrow 0, k-1$ **do**
  $\mathbf{e}_1 := CBD(PRF(r, N))$
  $N := N + 1$
end for
$e_2 := CBD(PRF(r, N))$
$\mathbf{r} := NTT((r))$
$\mathbf{u} := NTT^{-1}(\mathbf{A}^T \circ \mathbf{r}) + \mathbf{e}_1$
$v := NTT^{-1}(\mathbf{b}^T \circ \mathbf{r}) + e_2 + m$
return $(\mathbf{u} \| v)$

It is important that the ciphertext composes of two parts, only one of which is dependent on the message, so that the receiver has enough information in order to decrypt correctly.

Algorithm 3 Decryption

Input: sk, c
$m := v - NTT^{-1}(\mathbf{s}^T \circ NTT(\mathbf{u}))$
return m

It may not be readily obvious why it is this decryption works:

$$\begin{aligned} v - NTT^{-1}(\mathbf{s}^T \circ NTT(\mathbf{u})) &= NTT^{-1}(\mathbf{b}^T \circ \mathbf{r} + e_2 + m) - NTT^{-1}(\mathbf{s}^T \circ NTT(\mathbf{u})) \\ &= NTT^{-1}(\mathbf{b}^T \circ \mathbf{r} + e_2 + m) - NTT^{-1}(\mathbf{s}^T \circ \mathbf{A}^T \circ \mathbf{r} + \mathbf{e}_1) \\ &= NTT^{-1}(\mathbf{b}^T \circ \mathbf{r} + e_2 + m - \mathbf{s}^T \circ \mathbf{A}^T \circ \mathbf{r} + \mathbf{e}_1) \\ &= NTT^{-1}((\mathbf{A} \circ \mathbf{s})^T \circ \mathbf{r} - (\mathbf{s}^T \circ \mathbf{A}^T) \circ \mathbf{r} + \mathbf{e}^T + \mathbf{e}_1 + e_2 + m) \\ &= NTT^{-1}(m + \mathbf{e}^T + \mathbf{e}_1 + e_2) \\ &= m \end{aligned}$$

As we can assume the errors are small.[1]

[1] I'm not 100% sure on the last step as I have only seen an LWE setup encrypting a single bit and so you can disregard errors by seeing if it is approx. 0 or $\frac{q-1}{2}$.

References

[1] A. Basso et al. *SABER: Mod-LWR based KEM*. URL: https://www.esat.kuleuven.be/cosic/pqcrypto/saber/files/saberspecround3.pdf.

[2] J. Alwen et al. *Learning with Rounding, Revisited*. URL: https://link.springer.com/content/pdf/10.1007%2F978-3-642-40041-4_4.pdf.

[3] R. Avanzi et al. *CRYSTLS-KYBER, Algorithm Specifications and Supporting Documents*. URL: https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf.

[4] J. Katz and Y. Lindell. *Introduction to Modern Cryptography*. Second edition. Chapman & hall/crc cryptography and network security series. Boca Raton : CRC Press/Taylor & Francis, 2015. ISBN: 9781466570269.

[5] M. Rosulek. *The Joy of Cryptography*. URL: https://joyofcryptography.com.