

# Improving Secure Channels without Random Oracles based on Ring-LWE

Master's Thesis of

Robert Brede

At the KIT Department of Informatics  
KASTEL – Institute of Information Security and Dependability

First examiner: Prof. Jörn Müller-Quade

Second examiner: Prof. Thorsten Strufe

First advisor: M.Sc. Laurin Benz

Second advisor: M.Sc. Wasilij Beskorovajnov

15. May 2024 – 15. November 2024

Karlsruher Institut für Technologie  
Fakultät für Informatik  
Postfach 6980  
76128 Karlsruhe



# Abstract

When communicating over the internet, the current state of the art is to use a secure channel like TLS to protect confidentiality and authenticity. This is because the internet is a public network, which facilitates eavesdropping and manipulation of the data during transmission. These secure channels can be realized by a public key encryption scheme (PKE) combined with an authenticated channel. One way to realize the encryption is the KEM-DEM framework, which combines a key encapsulation mechanism (KEM) with a data encapsulation mechanism (DEM) to realize a PKE. Scalable quantum computers can break many current used key encapsulation mechanisms (KEMs). Thus, new KEMs base their security on other problems, which we suspect to be difficult to solve even for a quantum computer. One of them is learning with errors (LWE) and its ring variant, ring learning with errors (RLWE). There have been multiple new schemes like the newly standardized Kyber that base their security on variants of these problems. However, most of them are proven secure in the random oracle model (ROM) and those that are not either do not propose parameters or have key sizes that are impractically large. As the commonly used security notion of indistinguishability under chosen ciphertext attack (IND-CCA) is stronger than necessary for realizing secure channels, weaker notions such as the sender-binding KEM (SB-KEM) have been introduced together with the security notion of indistinguishability under sender-binding chosen plaintext attack (IND-SB-CPA<sub>SB-KEM</sub>).

This work improves the current state of secure channels that do not rely on the ROM by constructing a new IND-SB-CPA<sub>SB-KEM</sub> secure SB-KEM based on RLWE with concrete parameters. While the key and cipher sizes are worse than the ROM based Kyber, they improve upon all known IND-CCA secure KEMs without the ROM. The SB-KEM is then used in the KEM-DEM framework and combined with SPHINCS256 to realize a secure channel in the universal composability (UC) framework. For 87 bits of security it requires a bandwidth of 69.2 kilobytes (KB) and 97.3 KB for 128 bits. The second parameter set for the SB-KEM actually provides 164 bits of security but SPHINCS256 only 128. To confirm the theoretical results, a proof of concept for the SB-KEM is implemented in python.



# Zusammenfassung

Bei der Kommunikation über das Internet ist der aktuelle Stand der Technik, sichere Kanäle wie zum Beispiel TLS zu benutzen, um Vertraulichkeit und Authentizität zu schützen. Hintergrund dafür ist, dass das Internet ein öffentliches Netzwerk ist, welches einfaches Abhören und Manipulieren der versendeten Daten ermöglicht. Diese sicheren Kanäle können durch die Kombination eines Public Key Encryption Schemes (PKE) und eines authentifizierten Kanals realisiert werden. Eine Methode für die Konstruktion des PKEs ist das KEM-DEM Framework, welches einen Key Encapsulation Mechanism (KEM) mit einem Data Encapsulation Mechanism (DEM) kombiniert. Da skalierbare Quantum Computer viele aktuell genutzten KEMs brechen können, basieren neue KEMs ihre Sicherheit auf Probleme, von denen wir ausgehen, dass sie auch für Quantum Computer schwer zu lösen sind. Eines davon ist Learning with Errors (LWE) und dessen Ringvariante, Ring Learning with Errors (RLWE). Es gab mehrere neue Entwürfe von KEMs, die ihre Sicherheit auf diese Probleme basieren, wie zum Beispiel der kürzlich standardisierte Kyber. Jedoch ist die Sicherheit der meisten Verfahren mithilfe des Random Oracle Models (ROM) bewiesen und für die, die es nicht benutzen, werden entweder keine Parameter angegeben oder deren Schlüsselgrößen sind unpraktikabel groß. Da die häufig verwendete Notation von Indistinguishability under Chosen Ciphertext Attack (IND-CCA) stärker ist als nötig, um sichere Kanäle zu realisieren, wurden schwächere Begriffe vorgeschlagen. Ein Beispiel dafür ist der Begriff Sender-Binding KEM (SB-KEM) zusammen mit der Sicherheitsstufe Indistinguishability under Sender-Binding Chosen Plaintext Attack (IND-SB-CPA<sub>SB-KEM</sub>).

Diese Arbeit verbessert den Stand der Technik von sicheren Kanälen, die nicht das ROM benötigen, indem ein neuer IND-SB-CPA<sub>SB-KEM</sub> sicherer SB-KEM basierend auf RLWE konstruiert sowie konkrete Parameter bestimmt werden. Obwohl die Schlüssel- und Chiffregrößen schlechter als der ROM basierte Kyber sind, stellen sie eine Verbesserung gegenüber allen bekannten IND-CCA sicheren KEMs ohne ROM da. Der SB-KEM wird anschließend im KEM-DEM Framework benutzt und mit SPHINCS256 kombiniert, um einen sicheren Kanal im Universal Composability (UC) Framework zu realisieren. Für 87 Bit Sicherheit haben die Nachrichten eine Größe von 69.2 Kilobytes (KB) und 97.3 KB für 128 Bit. In der zweiten Einstellung bietet der SB-KEM sogar 164 Bit Sicherheit, jedoch SPHINCS256 nur 128. Um die theoretischen Ergebnisse zu bestätigen, ist ein Prototyp für den SB-KEM in Python implementiert.



# Contents

<b>Abstract</b>	<b>i</b>
<b>Zusammenfassung</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Related Work . . . . .	2
1.2 Our Contribution . . . . .	3
<b>2 Fundamentals</b>	<b>5</b>
2.1 Linear Algebra . . . . .	5
2.2 Lattices . . . . .	6
2.3 Algebraic Number Theory . . . . .	6
2.3.1 The Space $H$ . . . . .	6
2.3.2 Canonical and Coefficient embedding . . . . .	8
2.3.3 The Lattice of the Ring of Integers in $H$ . . . . .	10
2.3.4 Cyclotomic Number Fields . . . . .	10
2.4 Probability . . . . .	13
2.5 Gaussian Random Variables . . . . .	13
2.6 Semantic Security . . . . .	18
2.6.1 The KEM-DEM Framework . . . . .	19
2.6.2 Sender-Binding . . . . .	20
2.6.3 Signature Schemes . . . . .	21
2.7 Learning With Errors . . . . .	23
2.8 Trapdoor . . . . .	24
2.8.1 Inverting LWE . . . . .	25
2.9 Full-Rank Difference Encoding . . . . .	26
2.10 Universal Composability . . . . .	27
<b>3 Construction</b>	<b>33</b>
3.1 Concrete Parameters . . . . .	41
3.1.1 Concrete Ring-LWE hardness estimation . . . . .	43
3.1.2 Parameter Sets . . . . .	43
3.1.3 Comparison to existing KEMs . . . . .	44
<b>4 Secure Channels</b>	<b>47</b>
4.1 Concrete Instantiation . . . . .	50
4.2 Analysis . . . . .	50



<b>5</b>	<b>Implementation</b>	<b>53</b>
5.1	Decomposition of $f(X) = X^n + 1$ . . . . .	53
5.2	Ring operations . . . . .	54
5.3	Gaussian Sampling . . . . .	54
5.4	Inverting RLWE . . . . .	56
<b>6</b>	<b>Conclusion</b>	<b>57</b>
	<b>Acronyms</b>	<b>59</b>
	<b>List of Figures</b>	<b>61</b>
	<b>Bibliography</b>	<b>63</b>

# 1 Introduction

Modern communication over the internet uses secure channels like TLS to protect confidentiality and authenticity. The reason for this is the accessibility to the messages during transmission as the internet is a public network. This allows for eavesdropping and manipulation of the data before they reach their target. One way to realize these secure channels is the combination of a public key encryption scheme (PKE) with an authenticated channel [14]. The encryption on the other hand can be realized using the KEM-DEM framework [20]. The idea of the KEM-DEM framework is to have an asymmetric component, which generates and encrypts a symmetric key, which the symmetric component uses to encrypt the data. Because of these roles, the former is called a key encryption mechanism (KEM), whereas the latter is called data encryption mechanism (DEM). This split enables more flexibility regarding the length of the messages.

Previously, KEMs such as RSA [41] were based on the intractability assumptions of factorization and discrete logarithm. However, Shor’s algorithm [42] can solve these problems in polynomial time and therefore break current KEMs, given that a scalable quantum computer exists. Hence, new problems are chosen as underlying assumptions for these so called post-quantum cryptographic schemes to ensure security against quantum computers. A prominent problem is learning with errors (LWE), introduced by Regev [40]. It asks to find a secret  $s \in \mathbb{Z}_q^n$ , given samples of the form  $\langle a, s \rangle + e$ , where  $a \in \mathbb{Z}_q^n$  is uniformly random and  $e \in \mathbb{Z}_q$  is a small error term. The authors reduced LWE to problems in ideal lattices, which are assumed to be hard even for quantum computers. To improve performance, LWE has been adopted to the ring setting, which is called ring learning with errors (RLWE) [31].

However, most quantum secure KEM designs like the newly standardized Kyber [43] are proven indistinguishability under chosen ciphertext attack (IND-CCA) secure in the random oracle model (ROM). The ROM assumes a perfect hash function that is truly random. Although it cannot be implemented in practice as all systems on the world would need to request the random oracle for every hash function, it is broadly used in theoretic proofs. On the flip side, there have not been any practical attacks against constructions that have been proven secure using the ROM [28]. A major contribution to the heavy use of the ROM is the Fujisaki-Okamoto transformation [23], which transforms an one-way under chosen plaintext attack (OW-CPA) KEM into an IND-CCA one in the ROM.

There are only a hand full of post-quantum IND-CCA secure PKEs or KEMs that do not use the ROM. Additionally, those either do not propose concrete parameters, which means they only provide asymptotical growth of their parameters, or have large key sizes in the orders of megabytes (MB), which is highly unpractical. To ease the creation of KEMs, Benz et al. [7] introduced the sender-binding KEM (SB-KEM) together with the security notion

of indistinguishability under sender-binding chosen plaintext attack ( $\text{IND-SB-CPA}_{\text{SB-KEM}}$ ). The authors proved that they are sufficient for secure channels, when used in the KEM-DEM framework and combined with an authenticated channel. This work aims to improve secure channels that do not rely on the ROM by filling the lack of KEMs.

## 1.1 Related Work

There are two ways of improving secure channels. One way is the creation of new security notions that are easier to achieve but still sufficient to realize secure message transfer (SMT). They are either proven to realize SMT in the UC framework or shown to have stronger notions when combined with other primitives. The other way is to construct a PKE or KEM and prove that it fulfills a certain security notion. This also includes finding parameters for the scheme.

Starting with the security notions for PKEs, the most common one is IND-CCA, which Canetti [14] proved to be sufficient to realize SMT in the UC framework. However, it is also shown to be unnecessarily strong [16]. Therefore, weaker security notions have been introduced. Canetti, Krawczyk, and Nielsen [16] built SMT using a PKE, which is replayable CCA (RCCA) secure. MacKenzie, Reiter, and Yang [33] introduced the tag based encryption (TBE) with the security notion of indistinguishability under adaptive-tag weakly chosen ciphertext attack ( $\text{IND-aTAG-wCCA}$ ) and showed how to realize SMT with it. Kiltz [27] proved that a weaker notion known as indistinguishability under selective-tag weakly chosen ciphertext attack ( $\text{IND-sTAG-wCCA}$ ) can be combined with a signature scheme to achieve an IND-CCA secure KEM.

Beskorovajnov et al. [10] analyzed post-quantum secure channels that do not rely on the ROM and came to the conclusion, that there is a lack of schemes that have practical parameter sizes. The authors then introduced the new security notion of sender-binding encryption and proved secure channels based on them. Thereby, they eased the creation of secure channels, as their new security notion of indistinguishability under sender-binding chosen plaintext attack ( $\text{IND-SB-CPA}$ ) is a weaker security notion than  $\text{IND-sTAG-wCCA}$ .

To realize these security notions in the KEM-DEM framework, there have been multiple notions for the KEMs and DEMs. Cramer and Shoup [20] showed that an IND-CCA secure PKE can be realized from an IND-CCA secure KEM and an one-time IND-CCA secure DEM. Abe et al. [1] introduced tag-KEMs with the associated security notion of  $\text{CCA}_{\text{tag-KEM}}$  and showed that the combination with an one-time secure DEM is enough for an IND-CCA secure PKE.

Benz et al. [7] built upon the  $\text{IND-SB-CPA}$  notion, by introducing the SB-KEM, the KEM variant of the sender-binding setting together with its associated  $\text{IND-SB-CPA}_{\text{SB-KEM}}$  security notion. The authors proved that this security notion is sufficient to construct an  $\text{IND-SB-CPA}$  secure PKE by combining the SB-KEM with a DEM that is indistinguishable under one-time attack ( $\text{IND-OT}$ ). This concludes our overview of the security notions.

For the constructions of KEMs, we will not consider constructions that rely their security on the Diffie-Hellman assumption, as it is broken on a quantum computer by Shor’s algorithm [42].

There are many KEMs that are proven secure in the ROM, such as the newly standardized Kyber [43], NTRU-based [24] or BIKE [3]. On the other hand, there are not many post-quantum IND-CCA secure KEMs or PKEs without the ROM. Often, the authors do not provide concrete key sizes but only their asymptotic growth. Yu and Zhang [47] described an IND-sTAG-wCCA secure TBE based on the learning parity with noise. However, the authors did not provide concrete parameters. Benhamouda et al. [6] constructed an IND-CCA tag-based PKE but neither provide concrete parameters. Based on that PKE, Blazy, Chevalier, and Vu [11] build an oblivious transfer and provided concrete parameters. Although, the authors did not provide key sizes for the underlying PKE, these can be estimated to be above 100MB.

Boyen, Izabachène, and Li [13] introduced a hybrid encryption scheme without the ROM based on LWE using the trapdoor by Micciancio and Peikert [35]. The authors stated that their scheme can be adopted to the ring setting, but did not provide concrete parameters for neither their proven LWE variant nor the RLWE variant. Benz et al. [7] constructed an LWE-based SB-KEM drawing from the works of Boyen, Izabachène, and Li [13]. However, its key sizes are still not practical being in the order of megabytes. This work adopts their SB-KEM to the ring setting.

To the best of our knowledge, the best post-quantum secure KEMs that do not rely on the random oracle and provide concrete key sizes are the works of Xu and Li [45] and Yang, Ma, and Zhang [46]. However, their key sizes are in the order of megabytes.

## 1.2 Our Contribution

The main contribution of this work is a new SB-KEM. Its design is based on the LWE-based SB-KEM of Benz et al. [7]. It is proven to be IND-SB-CPA<sub>SB-KEM</sub> secure and correct based on RLWE without the ROM. Afterwards, concrete parameter sets are proposed based on the security estimations of the lattice estimator [2]. The SB-KEM reaches about 87 and 164 bit security with key sizes of 51.2 KB and 102 KB respectively, which is worse than Kyber, but a vast improvement compared to other designs that do not use the ROM.

The SB-KEM is then combined with a signature scheme to realize a secure channel in the universal composability (UC) framework. Concretely, the ideal functionality  $\mathcal{F}_{M-SMT}$  [10] is realized with the assumption of certificate authorities described by the ideal functionality  $\mathcal{F}_{CA}$  [15]. SPHINCS256 [8] is hereby used as signature scheme and a one-time pad as DEM. This yields a secure channel with message sizes of 69.2 KB or 97.3 KB for 87 or 128 bits of security, respectively. Lastly, a proof of concept is implemented in python to confirm the theoretical results.

The rest of the work is structured as follows. First, Chapter 2 introduces the fundamentals needed for this work. Then, Chapter 3 describes the new SB-KEM and proves its security and correctness alongside the parameter sets. This new construction is used in Chapter 4 to realize a secure channel. Lastly, Chapter 5 describes details for the proof of concept before we summarize and give an outlook in Chapter 6.

## 2 Fundamentals

For a function  $f$  and a set  $S$  we define  $f(S) = \{f(s) \mid s \in S\}$ . Analogous for a matrix  $A$  and a set  $S$ , we write  $AS = \{Ax \mid x \in S\}$ . Vectors are column vectors and denoted as lower case letters. For a vector  $x$ ,  $x_i$  denotes the  $i$ -th entry of  $x$  with the first being  $x_1$ . Matrices are uppercase letters and elements of a number field are written as lowercase bold like  $\mathbf{x}$ . The element-wise multiplication is denoted as  $\odot$ . For Vectors in  $\mathbb{C}^n$  or  $\mathbb{R}^n$  we have the norms  $\|a\| := \|a\|_2 = (\sum_{i=1}^n |a_i|^2)^{1/2}$  and  $\|a\|_\infty = \max_i |a_i|$ . For vectors  $a \in K^n, b \in K^m$  of a field or ring  $K$ , we write  $(a, b) \in K^{m+n}$  for appending the vectors. Similarly, for matrices  $A \in K^{n \times m}, B \in K^{n \times \omega}$ , appending the matrices is written as  $(A, B) \in K^{n \times (m+\omega)}$ . Throughout this work,  $I_n$  is the identity matrix of dimension  $n$ . For a matrix  $S$  and  $p \in \{2, \infty\}$ , its  $p$ -norm is  $\|S\|_p = \max_{\|x\|_p=1} \|Ax\|_p$ . For a matrix  $S \in \mathbb{R}^{n \times n}$ , its origin-centered parallelepiped is  $\mathcal{P}_{1/2}(S) = \{Sx \mid x \in \mathbb{R}^n, \|x\|_\infty \leq 1/2\}$ . The finite ring, which results from taking integers mod  $q$  is denoted as  $\mathbb{Z}_q = \mathbb{Z} \bmod q$ . The Gaussian distribution is the distribution proportional to  $\rho(x) = e^{-\pi \langle x, x \rangle}$ , where  $\langle \cdot, \cdot \rangle$  denotes the scalar product. For  $x \in \mathbb{R}$ ,  $\lfloor x \rfloor$  denotes rounding to the next integer. For  $c \in \mathbb{C}$ ,  $\bar{c}$  denotes its complex conjugated. For sets, fields or rings  $K_1, K_2$ , we write  $K_1 \cong K_2$  iff  $K_1$  is isomorph to  $K_2$ . The tensor product of two fields  $K_1, K_2$  is denoted as  $K_1 \times K_2$ . The logarithm with basis 2 is denoted as  $\log := \log_2$  whereas  $\ln := \log_e$ . A function  $f : \mathbb{R} \rightarrow \mathbb{R}$  is said to be *negligible* iff for every  $i \in \mathbb{N}$ , there is a  $x_0 \in \mathbb{R}$  such that  $|f(x)| < x^{-i}$  for every  $x > x_0$ . A function  $g$  is *overwhelming* if  $1 - g$  is negligible.

### 2.1 Linear Algebra

A matrix  $S \in \mathbb{R}^{n \times n}$  is *positive semidefinite* if  $x^T S x \geq 0$  for every  $x \in \mathbb{R}^n$ . A matrix  $\Sigma \in \mathbb{R}^{n \times n}$  is positive semidefinite if and only if there is a matrix  $S \in \mathbb{R}^{n \times n}$  such that  $\Sigma = S S^T$ . Upon positive semidefinite matrices there is a partial order denoted as  $\Sigma \preceq \Sigma'$  iff  $\Sigma' - \Sigma$  is positive semidefinite. We write  $S \leq T$  for matrices  $S, T$  if  $S S^T \preceq T T^T$ .

For a set  $S = \{s_1, \dots, s_n\} \subset V$  of linear independent vectors from a vector space  $V$  with scalar product  $\langle \cdot, \cdot \rangle$ , the Gram-Schmidt orthogonalization  $\tilde{S} = \{\tilde{s}_1, \dots, \tilde{s}_n\}$  is defined as

$$\begin{aligned} \tilde{s}_1 &= s_1 \\ \tilde{s}_i &= s_i - \sum_{v=1}^{i-1} \frac{\langle s_i, \tilde{s}_v \rangle}{\langle \tilde{s}_v, \tilde{s}_v \rangle} \tilde{s}_v. \end{aligned}$$

Note that  $\tilde{S}$  depends on the order, in which the vectors are orthogonalized.

## 2.2 Lattices

For a vector space  $V$ , a lattice  $\Lambda \subset V$  is a discrete subspace. A basis of  $\Lambda$  is a set of linear independent vectors  $B = \{b_1, \dots, b_i\} \subset V$  such that there is a unique linear combination with integer coefficients of  $B$  for each vector of the lattice. For a given lattice  $\Lambda$ , the basis is not unique, but each has the same amount of vectors. The rank of  $\Lambda$  is defined as  $|B|$ , the number of vectors in the basis. A lattice is called *full rank* if its rank is equal to the dimension of  $V$ . We will abuse notation and view the basis also as matrix, which contains all basis vectors as columns. A coset of a lattice  $\Lambda \subset V$  is the set  $\{x + c \mid x \in \Lambda\}$ , where  $c \in V$  is arbitrary but fixed. The dual of a Lattice  $\Lambda \subseteq V$  is defined as  $\Lambda^\vee = \{x \in V, \forall y \in \Lambda : \langle x, y \rangle \in \mathbb{Z}\}$ . The determinant of a lattice  $\Lambda$  is  $\det(\Lambda) = |\det(B)|$ , where  $B$  is any basis of  $\Lambda$ . For a lattice  $\Lambda$ ,  $\lambda_1(\Lambda)$  is the norm of the shortest vector in  $\Lambda$ . For  $i \leq n$ ,  $\lambda_i(\Lambda)$  denotes the length of the  $i$ -th shortest vector considering only linear independent vectors.

## 2.3 Algebraic Number Theory

A more detailed introduction can be found in textbooks like the works of Oggier and Viterbo [37]. A number field  $K$  is a finite field extension of  $\mathbb{Q}$ . It can be shown, that each number field is the field that arises from adding an algebraic number  $\zeta$  alongside all multiples and powers of it to  $\mathbb{Q}$ . This is denoted as  $\mathbb{Q}[\zeta]$ . The number field  $K = \mathbb{Q}[\zeta]$  is isomorph to the residue class field of the polynomial ring  $\mathbb{Q}[X]$  modulo  $f(X)$ , where  $f$  is the monic irreducible polynomial with  $f(\zeta) = 0$ , which is unique [25]. This residue class field is denoted as  $\mathbb{Q}[X]/f(X)$ . The polynomial  $f$  is called the minimal polynomial of  $\zeta$ . The order of  $K$  is equal to the degree of  $f$  and is denoted by  $n$ . Throughout this work, we will consider a number field in the  $\mathbb{Q}[X]/f(X)$  representation with representatives denoted as polynomials of degree  $n - 1$ . Calculations are often performed on the algebraic integers of a number field. They are defined as

**Definition 1.** (*Algebraic Integer*) An algebraic integer is a complex root of some monic polynomial whose coefficients are integers.

For any number field, its set of algebraic integers forms a ring with addition and multiplication. It is called the *ring of integers*. For a number field  $K = \mathbb{Q}[X]/f(X)$ , its ring of integers is  $R = \mathbb{Z}[X]/f(X)$ . For an integer  $q$ , the finite ring  $R_q$  is defined as  $R_q := R \bmod q = \mathbb{Z}_q[X]/f(X)$ . For integers  $q, k > 0$  and  $a \in R_q^k$ , the lattice  $\Lambda_q^\perp(a)$  is defined as

$$\Lambda_q^\perp(a) = \{x \in R^k : a^T x = 0 \pmod{q}\}.$$

### 2.3.1 The Space H

The space  $H$  is a special subspace of the complex vector space  $\mathbb{C}^n$  and is used to embed elements of a number field into the complex vector space, which allowed the famous

Minkowski's Theorem. For integers  $s_1, s_2$  with  $n = s_1 + 2s_2$ , the idea is to accommodate  $s_1$  real embeddings and  $2s_2$  complex ones. Hence,  $H \subset \mathbb{C}^n$  is defined as

$$H = \{x \in \mathbb{R}^{s_1} \times \mathbb{C}^{2s_2}, x_{2i-1+s_1} = \overline{x_{2i+s_1}}, 1 \leq i \leq s_2\}.$$

Note that in this definition, the indices are in a different order compared to definitions found in the literature [30]. We put the pairs of conjugated complex numbers next to each other to have a nicer structure in the matrices following. With the inner product induced on it by  $\mathbb{C}^n$ ,  $H$  is an inner product space that is isomorphic to  $\mathbb{R}^n$  [31]. The main idea is to store each complex number as two real numbers. Normally, this doubles the dimension, but as each complex number comes with its complex conjugated instead of storing the two complex numbers, the real and imaginary part is stored separately. Throughout this work, the isomorphism  $\psi$  is defined as

$$\psi : H \rightarrow \mathbb{R}^n, \psi(x) = \begin{pmatrix} x_1 \\ \vdots \\ x_{s_1} \\ \sqrt{2} \cdot \operatorname{Re}(x_{s_1+1}) \\ \sqrt{2} \cdot \operatorname{Im}(x_{s_1+1}) \\ \sqrt{2} \cdot \operatorname{Re}(x_{s_1+3}) \\ \sqrt{2} \cdot \operatorname{Im}(x_{s_1+3}) \\ \vdots \\ \sqrt{2} \cdot \operatorname{Re}(x_{s_1+2s_2}) \\ \sqrt{2} \cdot \operatorname{Im}(x_{s_1+2s_2}) \end{pmatrix}. \quad (2.1)$$

Only half of the complex numbers are used because the other half are the complex conjugated ones. For the complex numbers, a factor of  $\sqrt{2}$  is inserted to have  $\|x\|_{\mathbb{C}} = \|\psi(x)\|_{\mathbb{R}}$  and  $\langle x, y \rangle_{\mathbb{C}} = \langle \psi(x), \psi(y) \rangle_{\mathbb{R}}$ . We also define the embedding of one pair of complex numbers in  $\mathbb{R}^2$  as

$$\psi_2 : \{x \in \mathbb{C}^2 \mid x_1 = \overline{x_2}\} \rightarrow \mathbb{R}^2, \psi_2(a + bi, a - bi) = (\sqrt{2}a, \sqrt{2}b).$$

To see that  $\psi$  is an isomorphism, consider  $\psi_2$ , which embeds one pair of complex conjugated numbers in the same matter that  $\psi$  does. There are no two  $c_1, c_2 \in \mathbb{C}$  with  $\psi_2((c_1, \overline{c_1})) = \psi_2((c_2, \overline{c_2})) = (a, b) \wedge c_1 \neq c_2$  as the former would imply  $c_1 = \sqrt{2}a + \sqrt{2}bi = c_2$ , a contradiction to the latter. On the other side, for each  $(\sqrt{2}a, \sqrt{2}b) \in \mathbb{R}^2$ , there is a  $c = a + bi$  with  $\psi_2((c, \overline{c})) = (\sqrt{2}a, \sqrt{2}b)$ . As the real numbers and the pairs of complex numbers of  $H$  are independent under  $\psi$ , this analysis extends to  $\psi$  on  $H$ .

With element-wise multiplication and addition,  $H$  forms a field. The isomorphism  $\psi$  is also a field isomorphism from  $H$  to  $\mathbb{R}^n$ . Although addition in  $H$  is simply addition in  $\mathbb{R}^n$ , multiplication is not as simple. The following lemma shows how multiplication is translated to  $\mathbb{R}^n$ .





where  $\zeta_i$  is the  $i$ -th root of  $f$ . An embedding  $\sigma_i$  is said to be real if  $\zeta_i$  is a real number. Otherwise it is complex. For every complex root, its complex conjugate is also a root of  $f$ . Combining these embeddings yields the canonical embedding of  $K$  into  $H$ :

$$\sigma : K \rightarrow H, \mathbf{x} \mapsto (\sigma_1(\mathbf{x}), \dots, \sigma_n(\mathbf{x}))$$

Hereby, the embeddings are sorted in such a way that the first  $s_1$  are the real embeddings and the complex conjugated are next to each other. The canonical embedding is a field homomorphism with element-wise multiplication and addition in  $H$ . As operations are element-wise in  $H$ , it is enough to show that each embedding is homomorph, meaning that  $\sigma_i(\mathbf{a}) + \sigma_i(\mathbf{b}) = \sigma_i(\mathbf{a} + \mathbf{b})$  and analog for multiplication. For addition we have

$$\sigma_i(\mathbf{a} + \mathbf{b}) = \sum_{j=1}^n (a_j + b_j) \zeta_i^{j-1} = \sum_{j=1}^n a_j \zeta_i^{j-1} + \sum_{j=1}^n b_j \zeta_i^{j-1} = \sigma_i(\mathbf{a}) + \sigma_i(\mathbf{b}).$$

For the multiplication, interpret  $\mathbf{a}, \mathbf{b} \in K$  as polynomials  $a(X), b(X)$  and  $\sigma_i(\mathbf{a}) = a(\zeta_i)$ . Let  $c(X) = a(X) \cdot b(X) \bmod f(X)$  and  $a(X) \cdot b(X) = c(X) + g(X)f(X)$  for some polynomial  $g(X)$ . Then it holds that

$$\sigma_i(\mathbf{a}) \cdot \sigma_i(\mathbf{b}) = a(\zeta_i) \cdot b(\zeta_i) = c(\zeta_i) + g(\zeta_i)f(\zeta_i) = c(\zeta_i) = \sigma_i(\mathbf{a} \cdot \mathbf{b}).$$

For  $p \in \{2, \infty\}$ , the canonical embedding defines a  $p$ -norm for an element  $\mathbf{v} \in K$  as  $\|\mathbf{v}\|_p := \|\sigma(\mathbf{v})\|_p$ .

Another embedding is the coefficient embedding, which is to represent  $\mathbf{x} \in K$  by its coefficients relative to the power basis  $\{1, X, \dots, X^{n-1}\}$ . This also induces a norm. By default we use the norm induced by the canonical embedding. To specify which embedding is used for  $\mathbf{x} \in K$ , we use  $[\mathbf{x}]_c := \sigma(\mathbf{x})$  and  $\|\mathbf{x}\|_{c,p} := \|[\mathbf{x}]_c\|_p$  to denote the canonical embedding and its induced norm whereas  $[\mathbf{x}]_k \in \mathbb{Q}^n$  and  $\|\mathbf{x}\|_{k,p} := \|[\mathbf{x}]_k\|_p$  denotes the coefficient embedding with its norm. Note that if  $\mathbf{x} \in R$ , then  $[\mathbf{x}]_k \in \mathbb{Z}^n$ . We extend the notion to  $x \in H$  as  $[x]_k := [\sigma^{-1}(x)]_k$  and  $\|x\|_k := \|\sigma^{-1}(x)\|_k$ .

For a vector over a number field  $K$ , its norm depends on three variables. The norm used for the outer vector and the norm and embedding used for each element of  $K$ . For integer  $m$ ,  $x \in K^m$ ,  $p, q \in \{2, \infty\}$  and embedding  $e \in \{c, k\}$  we denote  $\|[x]_{e,p}\|_q$  as

$$\|[x]_{e,p}\|_q := \|v\|_q, v = \begin{pmatrix} \|\mathbf{x}_1\|_{e,p} \\ \vdots \\ \|\mathbf{x}_m\|_{e,p} \end{pmatrix}$$

That means  $q$  defines the norm on the outer vector, whereas  $e$  and  $p$  determine the embedding and norm of each element of  $K$ .

### 2.3.3 The Lattice of the Ring of Integers in $H$

As calculations of the SB-KEM are performed in the ring of integers  $R$  of a number field  $K = \mathbb{Q}[X]/f(X)$  with order  $n$ , this chapter analyzes the image of it under the canonical embedding. As the set of all coefficient embeddings  $\{[\mathbf{x}]_k \mid \mathbf{x} \in R\}$  is equal to  $\mathbb{Z}^n$ , it is useful to look on the connection of the coefficient embedding and the canonical embedding. A representative of the former can be embedded into the latter by the matrix multiplication

$$\sigma(\mathbf{x}) = B_K[\mathbf{x}]_k, \quad (\mathbf{x} \in K)$$

where

$$B_K = \begin{pmatrix} 1 & \zeta_1 & \zeta_1^2 & \dots & \zeta_1^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta_n & \zeta_n^2 & \dots & \zeta_n^{n-1} \end{pmatrix} \in \mathbb{C}^{n \times n}. \quad (2.2)$$

Hereby,  $\zeta_i = \sigma_i(X)$  is the  $i$ -th root of  $f$ . The matrix  $B_K$  is a Vandermonde matrix, which is invertible, if all  $\zeta_i$  are distinct [26]. It can be shown that this is the case [18].

The set  $\sigma(\mathbb{Z}[X]/f(X)) = \{B_K x \mid x \in \mathbb{Z}^n\}$  forms a ring with element-wise multiplication and addition. We will denote this ring as  $H_K$ . The set also forms a lattice with element-wise addition and scalar multiplication, which we will denote as  $\Lambda_K$ . By definition,  $B_K$  is a  $\mathbb{Z}$ -basis of  $H_K$  and a basis of  $\Lambda_K$ . We will sometimes abuse notation and perform ring operations on a lattice element. As the underlying sets are the same, it is performed by embedding the lattice element as its corresponding ring element with the trivial identity embedding, performing the operation and embedding the result back into the lattice.

An important property of  $H_K$  is that every non-zero element does not contain any zero entries as the following lemma shows.

**Lemma 2.** *Let  $K$  be a number field,  $H_K$  its canonically embedded ring of integers and  $x = (x_1, \dots, x_n) \in H_K$ . If  $x_i = 0$  for any  $i$ , then  $x = 0$ .*

*Proof.* As the embeddings  $\sigma_i$  are injective [31],  $\sigma_i(\mathbf{x}) = 0$  only for  $\mathbf{x} = 0$ . Therefore, for  $x \in H_K$  if any index is 0,  $x$  must be zero.  $\square$

### 2.3.4 Cyclotomic Number Fields

Cyclotomic number fields or cyclotomics for short are a subclass of number fields. The  $m$ -th cyclotomic number field is  $\mathbb{Q}[\zeta_m] \cong \mathbb{Q}[X]/\Phi_m(X)$ , where  $\zeta_m$  is the  $m$ -th root of unity and  $\Phi_m(X)$  its minimal polynomial. They are often used as they have nicer properties than number fields in general, especially, if  $m$  is a power of two. For these so called power of two cyclotomics, it can be shown that the reducing polynomial is  $\Phi_m(X) = X^{m/2} + 1$  [21].

Lyubashevsky, Peikert, and Regev [30] analyzed the fraction of invertible elements in the ring of integers of a cyclotomic after applying a modulo  $q$ . Their result is that the fraction is at least polynomial in  $n$  and  $q$ , as described in the following lemma.

**Lemma 3.** ([30], Claim 2.25) Consider the  $m$ -th cyclotomic number field of degree  $n$  for some  $m \geq 2$ . Then for any  $q \geq 2$ , the fraction of invertible elements in  $R_q$  is at least  $1/\text{poly}(n, \log q)$ .

For any number field  $K = \mathbb{Q}[X]/f(X)$ , it can be checked in polynomial time whether  $\mathbf{g} \in R_q = \mathbb{Z}_q[X]/f(X)$  is invertible with the extended euclidean algorithm adjusted to polynomials. Given the two polynomials  $f$  and  $g$  the algorithm yields polynomials  $a, b \in \mathbb{Q}[X]/f(X)$  such that

$$af + bg = 1.$$

The element  $\mathbf{g}$  is invertible iff  $b \pmod q$  exists in  $R_q$ . More concretely,  $b$  has the form  $b = b_n^{-1}(b_0 + b_1X + \dots + b_{n-1}X^{n-1})$  where  $b_i \in \mathbb{Z}$ . If  $b_n$  is invertible in  $\mathbb{Z}_q$ , then  $b$  exists in  $R_q$  and  $\mathbf{g}$  is invertible.

There is another approach, if the factorization of  $f$  into irreducible polynomials in  $\mathbb{Z}_q$  is known. If  $r$  monic polynomials  $f_i$  that are irreducible in  $\mathbb{Z}_q$  are known such that  $f = \prod_i f_i \pmod q$  and  $f_i \neq f_j (i \neq j)$ , then  $R_q$  is isomorph to  $\mathbb{Z}_q[X]/f_1(X) \times \dots \times \mathbb{Z}_q[X]/f_r$  by the Chinese Remainder Theorem (CRT)[38]. Therefore,  $\mathbf{g} \in R_q$  is invertible if and only if it is not zero modulo any  $f_i$ . This property can be used to check whether a polynomial is invertible or draw uniformly distributed invertible polynomials by drawing uniform nonzero polynomials modulo  $f_i$  for each  $i$ .

Another important statement from Lyubashevsky, Peikert, and Regev [30] is the regularity lemma. It analyzes the distribution of  $Ax$ , where  $A$  is a uniformly random matrix and  $x$  is a vector drawn from a discrete Gaussian distribution, which essentially is a Gaussian that is restricted to a discrete space like a lattice. It is defined more formally in Section 2.5. The result of the regularity lemma is, that  $Ax$  is close to uniformly random, under certain constraints.

**Lemma 4.** ([30], Corollary 7.5) Let  $R$  be the ring of integers in the  $m$ -th cyclotomic number field  $K$  of degree  $n$ ,  $q \geq 2$  an integer. Let  $k \leq l \leq \text{poly}(n)$  be positive integers. Assume  $A = (I_k, \bar{A}) \in (R_q)^{k \times l}$ , where  $I_k \in (R_q)^{k \times k}$  is the identity matrix and  $\bar{A} \in (R_q)^{k \times (l-k)}$  is uniformly random. Then with probability  $1 - 2^{-\Omega(n)}$  over the choices of  $\bar{A}$ , the distribution of  $Ax \in R_q^k$ , where each coordinate of  $x \in R_q^l$  is chosen from a discrete Gaussian distribution of parameter  $r > 2n \cdot q^{k/l+2/(nl)}$  over  $R$ , satisfies that the probability of each of the  $q^{nk}$  possible outcomes is in the interval  $(1 \pm 2^{-\Omega(n)})q^{-nk}$  (and in particular is within statistical distance  $2^{-\Omega(n)}$  of the uniform distribution over  $R_q^k$ ).

For a cyclotomic number field  $K$ , the length of the  $n$ -th shortest vector of  $\Lambda_K$  can be bound by the structure of the Basis  $B_K$  as shown in the following lemma.

**Lemma 5.** Let  $K$  be a cyclotomic number field. For the canonical embedded lattice  $\Lambda_K \subset H$  we have  $\lambda_n(\psi(\Lambda_K)) = \lambda_n(\Lambda_K) \leq \sqrt{n}$ .

*Proof.* Each vector in the basis  $B_K$  of  $\Lambda_K$  consists of roots of unity and thus has length  $\sqrt{n}$ . As all its  $n$  vectors are linear independent,  $\lambda_n(\Lambda) \leq \sqrt{n}$ . As  $\|\psi(x)\|_2 = \|x\|_2$  for any  $x \in H$  it holds that  $\lambda_n(\Lambda_K) = \lambda_n(\psi(\Lambda_K))$ .  $\square$

The next lemma proves the nice properties of power of two cyclotomics by analyzing the basis  $B_K$  and its inverse in more detail.

**Lemma 6.** *Let  $m$  be a power of two,  $n := m/2$  and  $K_m = \mathbb{Q}[X]/(X^n + 1)$  be the  $m$ -th cyclotomic. For the Basis  $B_{K_m}$  as in Eq. (2.2) it holds that*

$$B_{K_m}^{-1} = \frac{1}{n} B_{K_m}^H$$

and

$$\|B_{K_m}\|_2 = \sqrt{n}; \quad \|B_{K_m}^{-1}\|_2 = \frac{1}{\sqrt{n}}.$$

*Proof.* Let  $b_i = (1, \zeta_i, \dots, \zeta_i^{n-1})$ , where  $\zeta_i = \sigma_i(X)$  is the  $i$ -th root of  $X^n + 1$ . Then, for  $i \neq j$  using the geometric sum and the fact that  $\zeta_i^n = \bar{\zeta}_j^n = -1$

$$(B_{K_m} \cdot B_{K_m}^H)_{ij} = \sum_{v=0}^{n-1} (\zeta_i \bar{\zeta}_j)^v = \frac{1 - (\zeta_i \bar{\zeta}_j)^n}{1 - (\zeta_i \bar{\zeta}_j)} = \frac{1 - (-1 \cdot -1)}{1 - (\zeta_i \bar{\zeta}_j)} = 0.$$

Note that  $(\zeta_i \bar{\zeta}_j) \neq 1$  as  $\zeta_i^{-1} = \bar{\zeta}_i$  and  $\zeta_i \neq \zeta_j$ . For the diagonal entries we have

$$(B_{K_m} \cdot B_{K_m}^H)_{ij} = \|b_i\|_2^2 = n$$

Thus,  $B_{K_m} \cdot B_{K_m}^H = nI_n$ , which yields  $B_{K_m}^{-1} = \frac{1}{n} B_{K_m}^H$ .

Therefore,  $\frac{1}{\sqrt{n}} B_{K_m}$  is a unitary matrix, and

$$\begin{aligned} \|B_{K_m}\|_2 &= \sqrt{n} \cdot \left\| \frac{1}{\sqrt{n}} B_{K_m} \right\|_2 = \sqrt{n}, \\ \|B_{K_m}^{-1}\|_2 &= \frac{1}{n} \cdot \|B_{K_m}^H\|_2 = \frac{1}{n} \cdot \|B_{K_m}\|_2 = \frac{1}{\sqrt{n}}. \end{aligned}$$

$\square$

## 2.4 Probability

Distributions are denoted in calligraphic letters like  $\mathcal{X}, \mathcal{Y}$ . The uniform distribution over the finite set  $X$  is written as  $\mathcal{U}(X)$ . If the set  $X$  is clear from the context, we only write  $\mathcal{U}$ . If a variable  $x$  is distributed as distribution  $\mathcal{X}$ , we write  $x \sim \mathcal{X}$ . We write  $x \stackrel{\$}{\leftarrow} X$  if  $x$  is drawn uniformly from  $X$ . For a distribution  $\mathcal{X}$  over a set  $S$ ,  $a \in S$  and  $Y \subseteq S$ , we define  $\mathcal{X}(a) := \Pr [x = a \mid x \leftarrow \mathcal{X}]$  and  $\mathcal{X}(Y) := \Pr [x \in Y \mid x \leftarrow \mathcal{X}]$ . Regarding statistical distance, we adopt the notation of Genise et al. [22]. For two probability distributions  $\mathcal{X}, \mathcal{Y}$  over the same set, the *statistical distance*  $\Delta(\mathcal{X}, \mathcal{Y})$  is:

$$\Delta(\mathcal{X}, \mathcal{Y}) = \sup_A |\mathcal{X}(A) - \mathcal{Y}(A)|,$$

where  $A$  ranges over all measurable sets. For two real numbers  $x, y$  and  $\epsilon \geq 0$ ,  $x$  *approximates*  $y$  within relative error  $\epsilon$  (written  $x \approx_\epsilon y$ ) if  $x \in [1 - \epsilon, 1 + \epsilon] \cdot y$ . The symmetric relation  $(x \approx_\epsilon y) \wedge (y \approx_\epsilon x)$  is abbreviated as  $x \stackrel{\epsilon}{\approx} y$ . We write  $\mathcal{X} \approx_\epsilon \mathcal{Y}$  iff  $\mathcal{X}(z) \approx_\epsilon \mathcal{Y}(z)$  for every  $z$ . Again, the symmetric relation  $(\mathcal{X} \approx_\epsilon \mathcal{Y}) \wedge (\mathcal{Y} \approx_\epsilon \mathcal{X})$  is abbreviated as  $(\mathcal{X} \stackrel{\epsilon}{\approx} \mathcal{Y})$ .

Two distribution ensembles  $\mathcal{X}_n, \mathcal{Y}_n (n \in \mathbb{N})$  are called *statistically close* if the statistical distance  $\Delta(\mathcal{X}_n, \mathcal{Y}_n)$  is negligible in  $n$ .

## 2.5 Gaussian Random Variables

For the Gaussian random variables, we adopt the notation by Genise et al. [22]. The Gaussian function  $\rho$  is defined as  $\rho : \mathbb{R}^n \rightarrow \mathbb{R}, \rho(x) = \exp(-\pi \langle x, x \rangle)$ . By normalizing, we obtain the continuous Gaussian distribution  $\mathcal{D}$ . For a matrix  $S \in \mathbb{R}^{n \times n}$  the Gaussian distribution  $\mathcal{D}_S$  is defined as:

$$\mathcal{D}_S := \sqrt{2\pi} \cdot S \cdot \mathcal{D}.$$

For  $s \in \mathbb{R}$ , we write  $\mathcal{D}_s$  as short for  $\mathcal{D}_{sI_n}$ . We inserted the constant of  $\sqrt{2\pi}$ , such that  $\mathcal{D}_s$  has variance  $s^2$ .

For a lattice  $\Lambda \subset \mathbb{R}^n$ , the discrete Gaussian  $\mathcal{D}_{\Lambda, S}$  is the Gaussian distribution restricted to the lattice. This means, the probability of  $x$  is proportional to  $\mathcal{D}_S(x)$  for  $x \in \Lambda$  and zero otherwise. It is defined as:

$$\mathcal{D}_{\Lambda, S}(x) = \frac{\mathcal{D}_S(x)}{\mathcal{D}_S(\Lambda)} \quad (x \in \Lambda).$$

Note that the denominator is the normalization. To sample a discrete Gaussian for a lattice  $\Lambda \subset H$ , the samples are drawn from  $\mathcal{D}_{\psi(\Lambda), S}$  and embedded into  $H$  via the inverse of  $\psi$ .

For working with discrete Gaussians on a lattice  $\Lambda \subset \mathbb{R}^n$ , Micciancio and Regev [36] introduced the *smoothing parameter*  $\eta_\epsilon(\Lambda)$ . It is defined as follows.

**Definition 2.** (*Smoothing Parameter*) For a lattice  $\Lambda \subset \mathbb{R}^n$  and positive real  $\epsilon > 0$ , the smoothing parameter  $\eta_\epsilon(\Lambda)$  is the smallest real, such that  $\rho(s \cdot \Lambda^\vee) \leq 1 + \epsilon$ .

For an invertible matrix  $S \in \mathbb{R}^{n \times n}$ , we write  $\eta_\epsilon(\Lambda) \leq S$  if  $\eta_\epsilon(S^{-1}\Lambda) \leq 1$ . The name of the smoothing parameter comes from its fundamental property proved by Regev [40] and Micciancio and Regev [36]. Essentially, a Gaussian distribution above the smoothing parameter is large enough, such that the discrete structure of the lattice is "smoothed". This implies two facts. First, the Gaussian measure over the whole lattice does not change under translation of the lattice. Second, when choosing a uniformly random lattice point and adding a continuous Gaussian noise with width above the smoothing parameter, the resulting distribution is close to uniform over  $\mathbb{R}^n$ . These two facts are summarized in the following lemma. As other definitions, the version stated here is the one from Genise et al. [22].

**Lemma 7.** ([22], Lemma 3) *For any full rank lattice  $\Lambda \subset \mathbb{R}^n$  and  $\epsilon \geq 0$  where  $\eta_\epsilon(\Lambda) \leq 1$ , we have  $\rho(\Lambda + c) \approx_\epsilon 1/\det(\Lambda)$  for any  $c \in \mathbb{R}^n$ ; equivalently,  $(\mathcal{D}_{1/(2\pi)} \bmod \Lambda) \approx_\epsilon \mathcal{U} := \mathcal{U}(\mathbb{R}^n/\Lambda)$ .*

For a lattice in  $\mathbb{R}^n$ , the smoothing parameter can be bounded as follows.

**Lemma 8.** (Regev [40], Lemma 2.12) *For any  $n$ -dimensional lattice  $\Lambda \subset \mathbb{R}^n$  and  $\epsilon > 0$ ,*

$$\eta_\epsilon(\Lambda) \leq \lambda_n(\Lambda) \cdot \sqrt{\frac{\ln(2n(1+1/\epsilon))}{\pi}}.$$

*In particular, for any superlogarithmic function  $\omega(\log n)$ ,  $\eta_\epsilon(\Lambda) \leq \lambda_n(\Lambda) \cdot \sqrt{\omega(\log n)}$  for some negligible function  $\epsilon(n)$ .*

From the definition of the smoothing parameter follows:

**Lemma 9.** ([22], Lemma 2) *For any lattice  $\Lambda \subset \mathbb{R}^n$ ,  $\epsilon \geq 0$  and matrices  $S, T$  of full rank, we have  $\eta_\epsilon(\Lambda) \leq S$  if and only if  $\eta_\epsilon(T\Lambda) \leq TS$ .*

The following lemmas analyze the distribution after applying transformations to a Gaussian distribution. The goal is to analyze the distribution of  $Te + f$ , where  $T$  is a matrix and  $e$  and  $f$  are vectors. Hereby, all entries are elements of  $H_K$  for a number field  $K$  and are drawn from a discrete Gaussian. Starting with normal operations in  $\mathbb{R}^n$ , the first lemma deals with multiplication of a Gaussian distribution with a matrix.

**Lemma 10.** ([22], Lemma 1) *For any lattice  $\Lambda \subseteq \mathbb{R}^n$  and matrices  $S, T \in \mathbb{R}^{n \times n}$  representing linear functions where  $T$  is injective on  $\Lambda$ , we have*

$$T \cdot \mathcal{D}_{\Lambda, S} = \mathcal{D}_{T\Lambda, TS}.$$

The proof is in essence, that when  $T$  is injective, for each sample  $T \cdot \sqrt{2\pi}Sx, x \leftarrow \mathcal{D}$ , there is exactly one preimage  $\sqrt{2\pi}Sx$ . Therefore, the probability of drawing  $T \cdot \sqrt{2\pi}Sx$  is exactly the same as drawing  $\sqrt{2\pi}Sx$ , namely the probability of drawing  $x$  from  $\mathcal{D}$ .

The sum of two discrete Gaussian distributions is also a discrete Gaussian if certain properties about the smoothing parameter hold as shown by the following theorem, which is adapted to the scaling of  $\sqrt{2\pi}$  in our definition of the Gaussian distribution.

**Theorem 1.** ([22], Theorem 3) *Let  $\epsilon \in (0, 1)$ , define  $\epsilon' = 4\epsilon/(1 - \epsilon)^2$ . Let  $A_1, A_2 \subset \mathbb{R}^n$  be cosets of full-rank lattices  $\Lambda_1, \Lambda_2 \subseteq \mathbb{R}^n$  (respectively), let  $\Sigma_1, \Sigma_2$  be positive definite matrices where  $\eta_\epsilon(\Lambda_2) \leq \sqrt{2\pi} \cdot \sqrt{\Sigma_2}$  and let*

$$\mathcal{X} = \left[ (x_1, x_2) \mid x_1 \leftarrow \mathcal{D}_{A_1, \sqrt{\Sigma_1}}, x_2 \leftarrow x_1 + \mathcal{D}_{A_2 - x_1, \sqrt{\Sigma_2}} \right].$$

*If  $\eta_\epsilon(\Lambda_1) \leq \sqrt{2\pi} \cdot \sqrt{\Sigma_3}$  where  $\Sigma_3^{-1} = \Sigma_1^{-1} + \Sigma_2^{-1}$ , then the marginal distribution  $\mathcal{X}_2$  of  $x_2$  in  $\mathcal{X}$  satisfies*

$$\mathcal{X}_2 \stackrel{\epsilon'}{\approx} \mathcal{D}_{A_2, \sqrt{\Sigma_1 + \Sigma_2}}.$$

The next lemma deals with multiplication in  $H_K$ . As the underlying lemmas and theorems are proven over  $\mathbb{R}^n$ , we will analyze it in  $\psi(H_K)$ .

**Lemma 11.** *Let  $H_K$  be the embedded ring of integers for a number field  $K$ ,  $\Lambda_K$  its corresponding lattice and  $s_T$  be a real. For  $e \in H_K$  and  $t \leftarrow \mathcal{D}_{\Lambda_K, s_T}$ ,  $\psi(e \odot t)$  is distributed as  $\mathcal{D}_{A_e \psi(\Lambda_K), s_T A_e}$ .*

*Proof.* For  $e = 0$ , the statement is trivial. Otherwise, by Lemma 1, the matrix  $A_e$  is injective as linear function and fulfills  $\psi(e \odot t) = A_e \psi(t)$ . Therefore, Lemma 10 applies and  $\psi(e \odot t) = A_e \psi(t) \sim \mathcal{D}_{A_e \psi(\Lambda_K), s_T A_e}$ .  $\square$

Two important facts about the distribution of  $\psi(e \odot t)$  will be useful later. First, its variance  $s_T^2 A_e A_e^T = s_T^2 \Sigma_e$  is a diagonal matrix by definition of  $A_e$ . Second,  $A_e \psi(\Lambda_K) \subseteq \psi(\Lambda_K)$ . As  $H_K$  is a ring,  $e \odot t \in H_K$  for every  $e, t \in H_K$ . Thus, for any  $\psi(t) \in \psi(H_K) = \psi(\Lambda_K)$ ,  $A \cdot \psi(t) = \psi(e \odot t) \in \psi(\Lambda_K)$ . With that, the next lemma analyzes the sum  $Te + f$ .

**Lemma 12.** *Let  $K$  be a number field of order  $n$ ,  $H_K$  its embedded ring of integers and  $\Lambda_K$  the corresponding lattice. Let  $\Sigma_f$  be a positive semi-definite diagonal matrix. For  $s_T > 0$ , an integer  $m$  and  $1 \leq i \leq m$ , let  $t_i \leftarrow \mathcal{D}_{\Lambda_K, s_T}$ ,  $e_i \in H_K$  and  $f \leftarrow \mathcal{D}_{\Lambda_K, \sqrt{\Sigma_f}}$ . Let  $\epsilon := \epsilon(n) > 0$  be*

*negligible in  $n$ . If  $\eta_\epsilon(\psi(\Lambda_K)) \leq \sqrt{2\pi} \cdot A_{e_i}^{-1} \sqrt{\left( \Sigma_f^{-1} + (s_T^2 \Sigma_{e_i})^{-1} \right)^{-1}}$  for every  $i$ , where  $e_i \neq 0$  and  $\eta_\epsilon(\psi(\Lambda_K)) \leq \sqrt{2\pi} \cdot \sqrt{\Sigma_f}$ , then the distribution of*

$$f + \sum_{i=1}^m e_i \odot t_i$$

*is statistically close to  $\mathcal{D}_{\Lambda_K, \sqrt{\Sigma}}$ , where  $\Sigma = \Sigma_f + s_T^2 \sum_{i=1}^m \Sigma_{e_i}$ .*



*Proof.* We show this lemma by repeatedly using Theorem 1. Let  $\Lambda := \psi(\Lambda_K)$  be the embedding of  $\Lambda_K$  into  $\mathbb{R}^n$ . As stated in Lemma 11,  $\psi(e_i \odot t_i)$  is statistically close to  $\mathcal{D}_{A_{e_i}\Lambda, s_T A_{e_i}}$ . Each summand with  $e_i = 0$  can obviously be ignored and therefore we assume  $e_i \neq 0$  for each  $i$ .

The rest is shown by an induction over  $m$ . For the  $m$ -th addition, it is shown that  $\psi(f + \sum_{i=1}^m e_i \odot t_i)$  is statistically close to  $\mathcal{D}_{\Lambda, \sqrt{\Sigma_m}}$ , where  $\Sigma_m = \Sigma_f + s_T^2 \sum_{i=0}^{k-1} \Sigma_{e_k}$ . For  $m = 0$  this is simply  $\psi(f) \sim \mathcal{D}_{\Lambda, \sqrt{\Sigma_f}}$ , which follows from the definition of  $\mathcal{D}_{\Lambda_K, \sqrt{\Sigma_f}}$ .

For  $m + 1$ , we use Theorem 1 with the following instantiation:

$$\begin{aligned} A_1 &= A_{e_m}\Lambda, \quad \Sigma_1 = s_T^2 \Sigma_{e_m}, \\ A_2 &= \Lambda, \quad \Sigma_2 = \Sigma_m. \end{aligned}$$

The Theorem has two requirements:

$$\begin{aligned} \eta_\epsilon(\Lambda) &\leq \sqrt{2\pi} \cdot \sqrt{\Sigma_m}, \\ \eta_\epsilon(A_{e_m}\Lambda) &\leq \sqrt{2\pi} \cdot \sqrt{\Sigma_3} \end{aligned}$$

where  $\Sigma_3^{-1} = \Sigma_1^{-1} + \Sigma_2^{-1}$ . The first equation holds as

$$\eta_\epsilon(\Lambda) \leq \sqrt{2\pi} \cdot \sqrt{\Sigma_f} \leq \sqrt{2\pi} \cdot \sqrt{\Sigma_f + s_T^2 \sum_{i=1}^m \Sigma_{e_m}} = \sqrt{2\pi} \cdot \sqrt{\Sigma_m}.$$

As  $e_m \neq 0$ ,  $A_{e_m}^{-1}$  exists and is injective. Therefore, by Lemma 9 the second inequality is equal to

$$\eta_\epsilon(\Lambda) \leq \sqrt{2\pi} \cdot A_{e_m}^{-1} \sqrt{\Sigma_3}.$$

As  $\Sigma_f$  and  $\Sigma_{e_m}$  are diagonal matrices, it holds that

$$\left( \Sigma_f^{-1} + (s_T^2 \Sigma_{e_m})^{-1} \right)^{-1} \preceq \left( \left( \Sigma_f + s_T^2 \sum_{i=0}^{k-1} \Sigma_{e_i} \right)^{-1} + (s_T^2 \Sigma_{e_m})^{-1} \right)^{-1} = \Sigma_3$$

and therefore

$$\eta_\epsilon(\Lambda) \leq \sqrt{2\pi} \cdot A_{e_m}^{-1} \sqrt{\left( \Sigma_f^{-1} + (s_T^2 \Sigma_{e_m})^{-1} \right)^{-1}} \leq \sqrt{2\pi} \cdot A_{e_m}^{-1} \sqrt{\Sigma_3}.$$

Therefore,  $\psi(\sum_{i=1}^m e_i \odot t_i + f)$  is statistically close to  $\mathcal{D}_{\Lambda, \sqrt{\Sigma}}$ . The statement follows from this by definition of the Gaussian distribution on elements in  $H_K$ .

□

The next lemma concerns bounding the norm of a vector drawn from a Gaussian distribution. We start with the bounds by Banaszczyk [5].

**Lemma 13.** ([5], Lemma 1.5(i)) Let  $\Lambda \subset \mathbb{R}^n$  be lattice and  $B_n \subset \mathbb{R}^n$  be the open sphere of unity. For each  $c \geq (2\pi)^{-1/2}$  one has

$$\rho(\Lambda \setminus c\sqrt{n}B_n) < \left( e^{-\pi c^2} c\sqrt{2\pi e} \right)^n \rho(\Lambda).$$

From this follows the bounds we need.

**Lemma 14.** Let  $a > 1$  and  $c = \frac{a}{\sqrt{2\pi}}$ . Let  $\Lambda \subset \mathbb{R}^n$  be a lattice. Let  $\Sigma$  be a diagonal matrix and  $s = \|\sqrt{\Sigma}\|_2$ . Then, the probability

$$\Pr \left[ \|x\|_2 > sa\sqrt{n} \mid x \leftarrow \mathcal{D}_{\Lambda, \sqrt{\Sigma}} \right] < \left( e^{-\pi c^2} c\sqrt{2\pi e} \right)^n$$

is negligible in  $n$ .

*Proof.* As the Gaussian distribution only depends on  $\Sigma$ , not on the choice of  $\sqrt{\Sigma}$ , choose  $S := \sqrt{\Sigma}$  invertible. Such a matrix exists because  $\Sigma$  is a diagonal matrix. Let  $y \leftarrow \mathcal{D}_{\Lambda, I_n}$  and  $B_n \subset \mathbb{R}^n$  the open sphere of unity. Let  $\Lambda' = (\sqrt{2\pi}S)^{-1}\Lambda$ . Using the definition of  $\mathcal{D}_{\Lambda, \sqrt{\Sigma}}$ , we have

$$\begin{aligned} \Pr[\|x\|_2 \geq sa\sqrt{n} \mid x \leftarrow \mathcal{D}_{\Lambda, S}] &= \Pr[\|\sqrt{2\pi}Sy\|_2 \geq sa\sqrt{n} \mid y \leftarrow \mathcal{D}_{\Lambda', I_n}] \\ &\leq \Pr[\|\sqrt{2\pi}S\|_2 \cdot \|y\|_2 \geq sa\sqrt{n} \mid y \leftarrow \mathcal{D}_{\Lambda', I_n}] \\ &= \Pr[\|y\|_2 \geq c\sqrt{n} \mid y \leftarrow \mathcal{D}_{\Lambda', I_n}]. \end{aligned}$$

By Lemma 13 it holds that

$$\Pr[\|y\|_2 \geq c\sqrt{n} \mid y \leftarrow \mathcal{D}_{\Lambda', I_n}] = \frac{\rho(\Lambda' \setminus c\sqrt{n}B_n)}{\rho(\Lambda')} < \left( e^{-\pi c^2} c\sqrt{2\pi e} \right)^n.$$

For  $a, c$  as defined, we have

$$\left( e^{-\pi c^2} c\sqrt{2\pi e} \right) < 1$$

and therefore, the probability is negligible in  $n$ .  $\square$

The next lemma shows that for a power of two cyclotomic, a canonically embedded Gaussian distribution translates nicely into the coefficient embedding.

**Lemma 15.** Let  $m$  be a power of two,  $n = m/2$  and  $K = \mathbb{Q}[X]/(X^n + 1)$  be the  $m$ -th cyclotomic. Let  $R = \mathbb{Z}[X]/(X^n + 1)$  be its ring of integers and  $\Lambda = \sigma(R)$  its canonically embedded lattice with basis  $B_K$ . For a positive definite matrix  $\Sigma$  and  $x \leftarrow \mathcal{D}_{\Lambda, \sqrt{\Sigma}}$  we have

$$[x]_k \sim \mathcal{D}_{\mathbb{Z}^n, \sqrt{\Sigma_2}}$$

where  $\Sigma_2 = B_K^{-1}\Sigma B_K^{-H}$ . In particular, if  $\Sigma = vI_n$  for some  $v > 0$ , then  $\Sigma_2 = \frac{1}{n}\Sigma$ .

*Proof.* By definition of  $[x]_k$  and Lemma 10, we have

$$[x]_k = B_K^{-1} \cdot x \sim \mathcal{D}_{B_K^{-1}\Lambda, B_K^{-1}S}.$$

As  $\Lambda = \{B_K \cdot x \mid x \in \mathbb{Z}^n\}$ , it holds that  $B_K^{-1}\Lambda = \mathbb{Z}^n$ . The variance is

$$\Sigma_2 = B_K^{-1}S(B_K^{-1}S)^H = B_K^{-1}\Sigma B_K^{-H}.$$

For the second part, by Lemma 6,  $B_U := \sqrt{n}B_K^{-1}$  is a unitary matrix and thus

$$\Sigma_2 = B_K^{-1}\Sigma B_K^{-H} = B_K^{-1}vI_n B_K^{-H} = vB_K^{-1}B_K^H = v \cdot \frac{1}{n}I_n = \frac{1}{n} \cdot \Sigma.$$

□

## 2.6 Semantic Security

This chapter covers the notions of semantic security, which are used throughout this work. These include the formal definitions of cryptographic schemes used to provide confidentiality and authenticity as well the different security notions that assess their security.

A probabilistic polynomial timed (PPT) attacker or algorithm is an algorithm, which can use polynomial amount of randomness but its runtime is bounded by a polynomial on the length of the input.

Section 2.4 defined the statistically closeness of distribution ensembles. In cryptography, there is an equivalent definition of it, which states that no attacker can distinguish samples from the two distributions with non-negligible probability. A relaxation of this is *computational indistinguishability*, in which the attacker is PPT. The formal definition is as follows.

**Definition 3.** (*Computationally Indistinguishable*) Two distributions ensembles  $(\mathcal{X})_\kappa, (\mathcal{Y})_\kappa$  are computationally indistinguishable iff for every PPT attacker  $\mathcal{A}$  the advantage  $adv_{\mathcal{A}, \mathcal{X}, \mathcal{Y}}^{IND}$  defined as

$$adv_{\mathcal{A}, \mathcal{X}, \mathcal{Y}}^{IND}(\kappa) = |\Pr [\mathcal{A}(z) = 1 \mid z \leftarrow \mathcal{X}_\kappa] - \Pr [\mathcal{A}(z) = 1 \mid z \leftarrow \mathcal{Y}_\kappa]|$$

is negligible in  $\kappa$ .

A key derivation function (KDF) is a function  $KDF : SEED \rightarrow K$ , which maps a seed to a key, such that its output for a random seed is indistinguishable from a random key. The attacker hereby does not know the chosen seed. The advantage of any attacker is the chance of distinguishing the KDF from random. More formally it is defined as:

**Definition 4.** (*Key Derivation Function*) Let  $SEED$  and  $K$  be sets. A key derivation function (KDF) is a function  $KDF : SEED \rightarrow K$ , such that for any PPT attacker the advantage

$$adv_{\mathcal{A}, KDF}^{KDF}(\kappa) = |\Pr [\mathcal{A}(z) = 1 \mid z \leftarrow \mathcal{U}(K)] - \Pr [\mathcal{A}(KDF(z)) = 1 \mid z \leftarrow \mathcal{U}(SEED)]|$$

is negligible in  $\kappa$ .

### 2.6.1 The KEM-DEM Framework

Part of modern asymmetric cryptography is the KEM-DEM framework introduced by Cramer and Shoup [20]. It consists of a key encapsulation mechanism (KEM), which generates symmetric keys and encapsulates them and a data encapsulation mechanism (DEM), which uses the symmetric key to encapsulate the data. The combination of the two yield a PKE, which is used to provide confidentiality. More formally, a KEM is defined as:

**Definition 5.** (KEM, [7]) A key encapsulation mechanism (KEM) is given by a set of three PPT algorithms ( $gen, enc, dec$ ) with

$$\begin{aligned} gen &: 1^\kappa \mapsto (sk, pk) \\ enc &: pk \mapsto (K, C) \\ dec &: (sk, C) \mapsto K \end{aligned}$$

such that the correctness property holds. That means  $K = dec(sk, C)$  whenever  $(sk, pk) \leftarrow gen(1^\kappa)$  and  $(K, C) \leftarrow enc(pk)$ .

Whereas the DEM is defined as:

**Definition 6.** (DEM, [7]) A data encapsulation mechanism (DEM) is given by a set of two PPT algorithms ( $DEM.Enc, DEM.Dec$ ) with

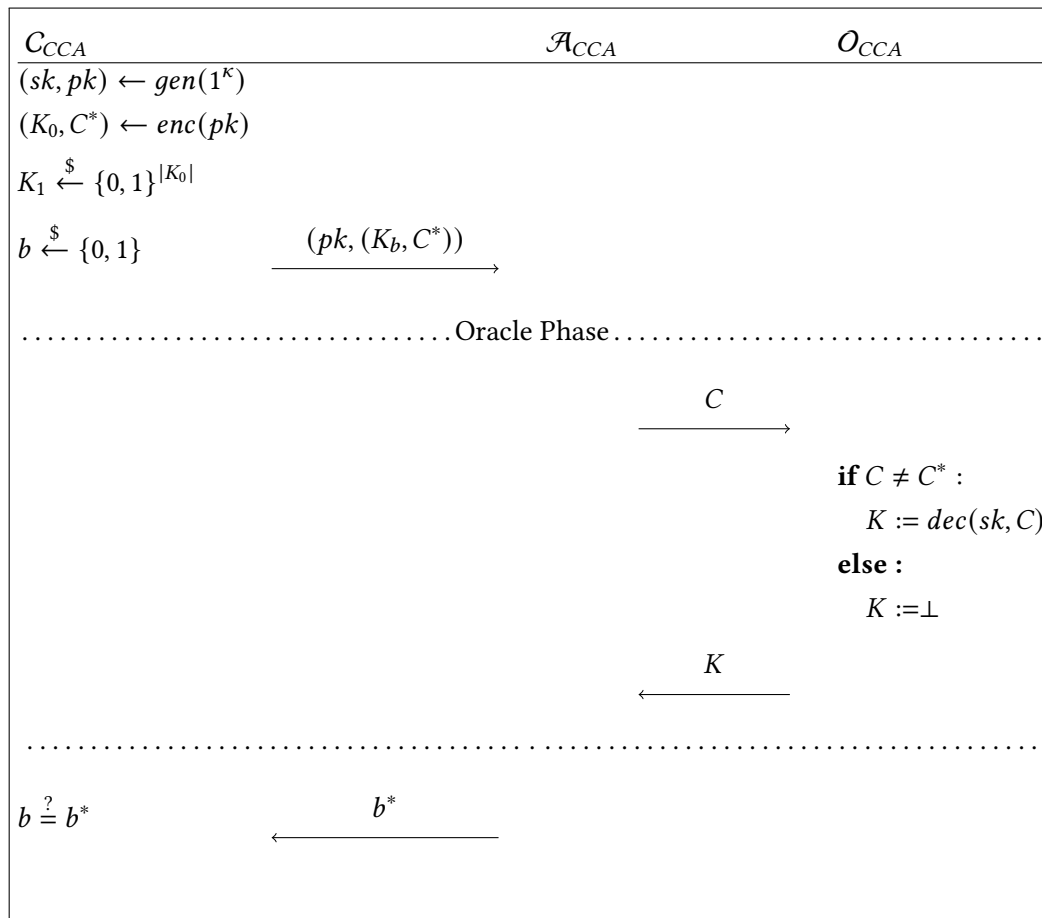
$$\begin{aligned} DEM.enc &: (K, m) \mapsto c \\ DEM.dec &: (K, c) \mapsto m \end{aligned}$$

such that  $m = DEM.dec(K, c)$  whenever  $c \leftarrow DEM.enc(K, m)$  (correctness).

These definitions however do not make any assessment about the security of the KEM or the DEM. A common definition to fill this gap for KEMs is IND-CCA, in which the attacker has to distinguish a key  $K$  generated by  $enc$  from a random one, given the corresponding cipher of  $K$  and the public key. As additional help, the attacker has access to a decryption oracle, which decrypts any ciphers other than the challenge cipher. This oracle provides the attacker with some information about the secret key. In the real world, this corresponds to a target that reacts predicably to messages from the attacker. The formal definition is as follows.

**Definition 7.** (IND-CCA) A KEM  $\Gamma = (gen, enc, dec)$  is indistinguishable under chosen ciphertext attack (IND-CCA), iff for every PPT attacker  $\mathcal{A}$ , the advantage  $Adv_{\mathcal{A}, \Gamma}^{IND-CCA}(\kappa) = \left| \Pr [b = b^*] - \frac{1}{2} \right|$  of winning the IND-CCA game depicted in Fig. 2.1 is negligible in  $\kappa$ .

More generally,  $K_1$  is from the key space  $K$ , instead of  $\{0, 1\}^{|K_0|}$ . But in our work, it is always  $K = \{0, 1\}^{|K_0|}$ . IND-CCA is also defined for DEMs, but for a secure channel, the DEM only needs to fulfill a weaker security notion named indistinguishable under one time attack (IND-OT). In this notion, the attacker receives only one encryption and does not have access to neither an encryption nor decryption oracle. Formally, it is defined as:

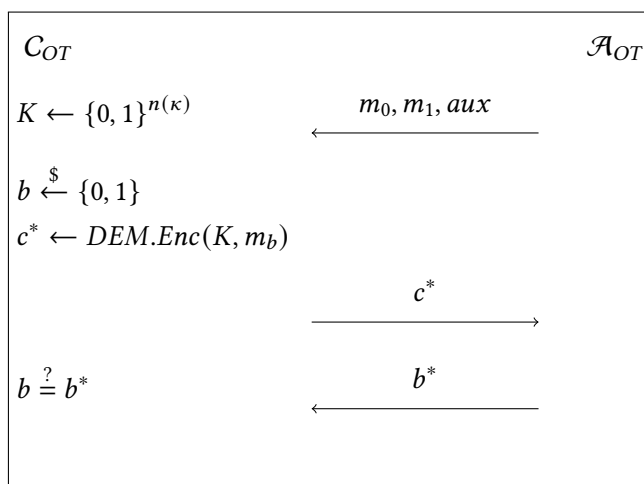


**Figure 2.1:** The IND-CCA Game for KEMs

**Definition 8.** (IND-OT, [7]) A DEM  $\Sigma = (DEM.enc, DEM.dec)$  satisfies indistinguishability under one-time attack (IND-OT), iff for any PPT adversary  $\mathcal{A}$ , the advantage  $Adv_{\mathcal{A}, \Sigma}^{IND-OT}(\kappa) = |Pr[b = b^*] - \frac{1}{2}|$  of winning the game depicted in Fig. 2.2 is negligible in  $\kappa$ .

### 2.6.2 Sender-Binding

Benz et al. [7] introduced a variation of the KEM, which uses a sender identity (ID) in encryption and decryption. This sender ID serves two purposes. On the one hand, it stops replay attacks by binding the ciphertext to the sender ID. On the other hand, it allows for a weaker notion than IND-CCA, in which the cipher is malleable but bound to the sender. The adapted KEM is called sender-binding KEM (SB-KEM) and is as follows.



**Figure 2.2:** The IND-OT Game for DEMs

**Definition 9.** (SB-KEM, [7]) A sender-binding key encapsulation mechanism (SB-KEM) is given by a set of three PPT algorithms ( $gen$ ,  $enc$ ,  $dec$ ) with

$$\begin{aligned}
 gen &: 1^\kappa \mapsto (sk, pk) \\
 enc &: (pk, S) \mapsto (K, C) \\
 dec &: (sk, S, C) \mapsto K
 \end{aligned}$$

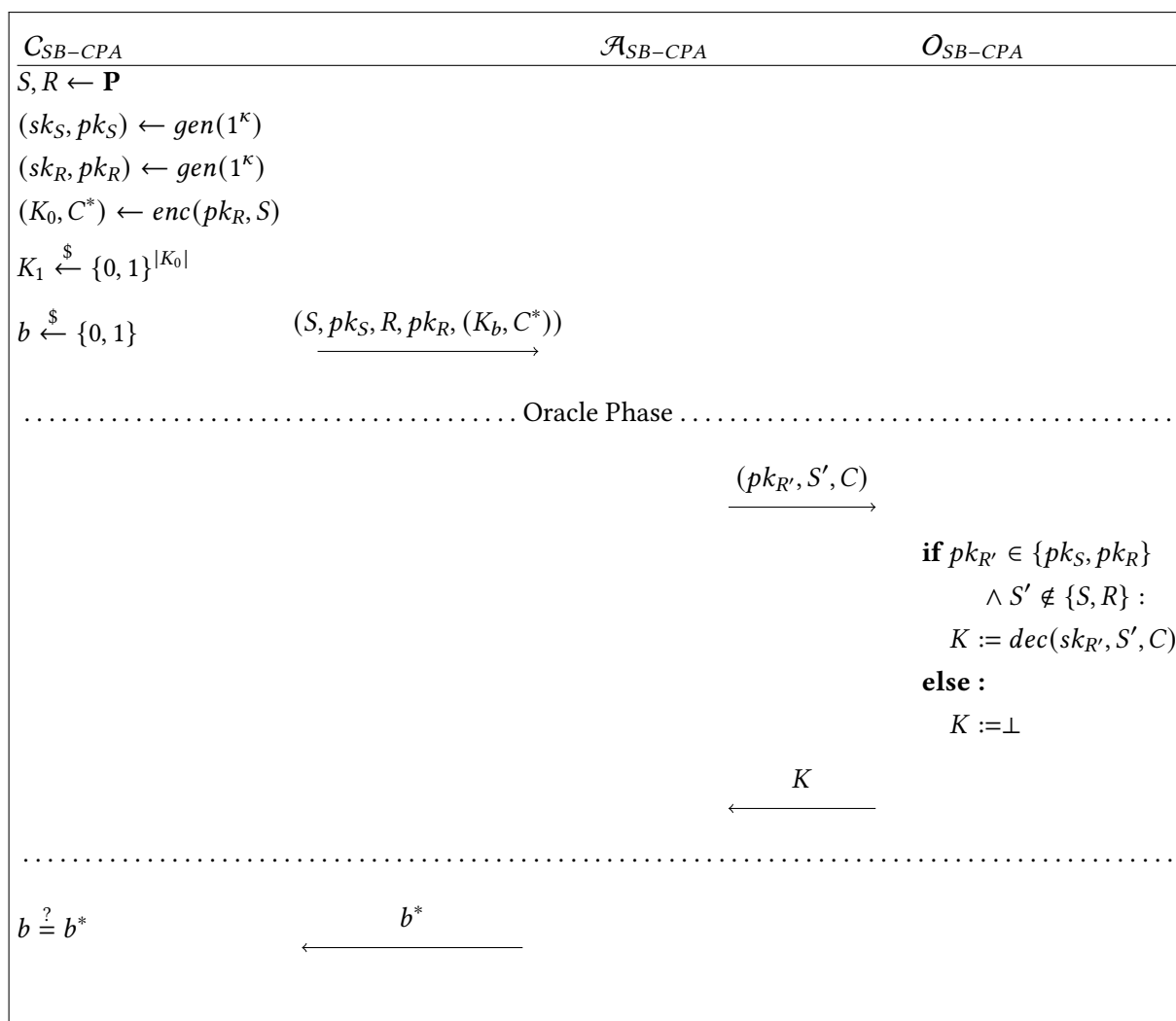
such that correctness holds, which means  $K = dec(sk, S, C)$  for  $(sk, pk) \leftarrow gen(1^\kappa)$  and  $(K, C) \leftarrow enc(pk, S)$ .

Note that  $K$  only needs to be the same if the same sender ID is used. In terms of security, the authors also introduced the security notion  $IND-SB-CPA_{SB-KEM}$ . It is similar to IND-CCA, however the oracle does not decrypt messages, where the sender is the sender or receiver of the challenge cipher. Formally it is defined as:

**Definition 10.** ( $IND-SB-CPA_{SB-KEM}$ , [7]) An SB-KEM  $\Gamma = (gen, enc, dec)$  with the set of Party IDs  $\mathcal{P}$  satisfies indistinguishability under sender-binding chosen plaintext attack ( $IND-SB-CPA$ ) security, iff for any PPT adversary  $\mathcal{A}$  the advantage  $adv_{\mathcal{A}, \Gamma}^{IND-SB-CPA}(\kappa) = \left| Pr [b = b^*] - \frac{1}{2} \right|$  to win the IND-SB-CPA game shown in Fig. 2.3 is negligible in the security parameter  $\kappa$ .

### 2.6.3 Signature Schemes

Signature schemes are used to provide authenticity. They are defined as follows.



**Figure 2.3:** The IND-SB-CPA<sub>SB-KEM</sub> Game

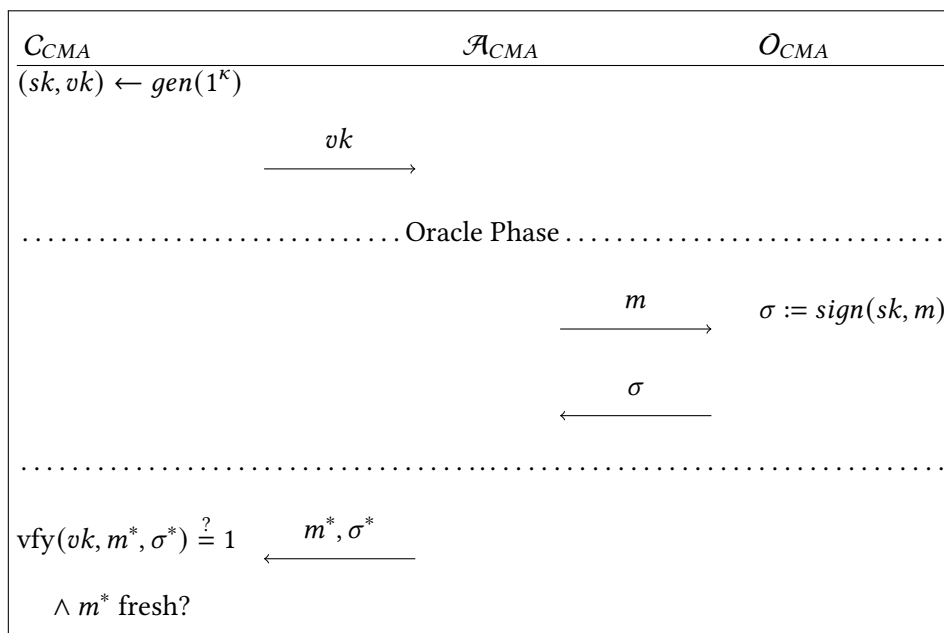
**Definition 11.** (Signature Scheme) A signature schemes  $\Sigma = (gen, sign, vfy)$  is given by a set of three probabilistic polynomial time (PPT) algorithms  $(gen, sign, vfy)$  with

$$\begin{aligned}
 gen &: 1^\kappa \mapsto (sk, vk) \\
 sign &: (sk, m) \mapsto \sigma \\
 vfy &: (vk, m, \sigma) \mapsto \{0, 1\}
 \end{aligned}$$

such that  $vfy(vk, m, sign(sk, m)) = 1$  for  $(sk, vk) \leftarrow gen(1^\kappa)$  (correctness).

The security demand is that an attacker cannot forge a valid signature without a secret key, given many valid signatures. The new signature has to be for a *fresh* message, which means that no valid signature for that message was already revealed. The security notion is denoted as EUF-CMA and is formally defined as:

**Definition 12.** (EUF-CMA) A signature scheme  $\Sigma = (\text{gen}, \text{sign}, \text{vfy})$  is existential unforgeable under chosen message attack (EUF-CMA), iff for every PPT attacker  $\mathcal{A}$ , the advantage  $\text{adv}_{\mathcal{A}, \Sigma}^{\text{EUF-CMA}}(\kappa) = |\Pr[\text{vfy}(vk, m^*, \sigma^*) = 1]|$  of winning the EUF-CMA game depicted in Fig. 2.4 is negligible in  $\kappa$ .



**Figure 2.4:** The EUF-CMA Game

## 2.7 Learning With Errors

Learning with errors (LWE) proposed by Regev [40] is a problem used in cryptography as it is assumed to be hard to solve even for quantum computers. The problem asks to find a secret  $s \in \mathbb{Z}^n$  given samples of the form  $as + e$ , where  $a \in \mathbb{Z}^n$  is uniform and the error  $e \in \mathbb{Z}^n$  is drawn from an error distribution  $\chi$ , for example a Gaussian distribution. This work uses the ring variation called ring LWE (RLWE) in which the ring of integers over a number field is used instead of  $\mathbb{Z}^n$ . More specifically, we use the non-dual version, in which the secret is sampled in  $R$  instead of its dual  $R^\vee$ . The RLWE distribution in the canonical embedding is defined as follows.

**Definition 13.** (RLWE Distribution) Let  $H_K$  be the canonical embedded ring of integers for a number field  $K$ . For a modulo  $q$ , distribution  $\chi$  on  $H_q := H_K/qH_K$ , and a secret  $s \in H_q$ , the RLWE distribution  $A_{s,q,\chi}$  is defined as sampling the value  $a$  uniformly from  $H_q$ ,  $e \leftarrow \chi$  and outputting  $(a, as + e) \in H_q \times H_q$ .

Based on this distribution, there are two versions of the problem, decision-RLWE and search-RLWE. The decision-RLWE $_{q,\chi}$  asks to distinguish between a uniform distribution



on  $H_q \times H_q$  and  $A_{s,q,\chi}$ , whereas the search-LWE $_{q,\chi}$  asks to find  $s$ , given samples from  $A_{s,q,\chi}$ . In its standard form,  $s$  is drawn from a uniform distribution and then fixed. In contrast, in the normal RLWE $_{q,\chi}$  (NRLWE $_{q,\chi}$ ),  $s$  is chosen from  $\chi$  and then fixed. The advantage of an attacker  $\mathcal{A}$  against the decision-NRLWE (d-NRLWE) is its advantage of distinguishing, formally

$$\text{adv}_{\mathcal{A}}^{d\text{-NRLWE}_{q,\chi}}(\kappa) = |\Pr [\mathcal{A}(a, b) = 1 \mid (a, b) \leftarrow \mathcal{U}(H_q \times H_q)] - \Pr [\mathcal{A}(a, b) = 1 \mid (a, b) \leftarrow A_{s,q,\chi}, s \leftarrow \chi] |.$$

On the contrary, in the search-NRLWE (s-NRLWE), the advantage of an attacker  $\mathcal{A}$  is its probability of finding  $s$ , formally

$$\text{adv}_{\mathcal{A}}^{s\text{-NRLWE}_{q,\chi}}(\kappa) = \Pr [\mathcal{A}(a, b) = s \mid (a, b) \leftarrow A_{s,q,\chi}, s \leftarrow \chi].$$

The d-NRLWE $_{q,\chi}$  (or s-NRLWE $_{q,\chi}$ ) assumption states that for every PPT attacker  $\mathcal{A}$ , the advantage  $\text{adv}_{\mathcal{A}}^{d\text{-NRLWE}_{q,\chi}}(\kappa)$  (or  $\text{adv}_{\mathcal{A}}^{s\text{-NRLWE}_{q,\chi}}(\kappa)$ ) is negligible in  $\kappa$ . If the lattice  $\Lambda$  is clear from the context, we write d-NRLWE $_{q,r}$  for  $r \in \mathbb{R}$  as shorthand for d-NRLWE $_{q,\chi}$  with  $\chi = \mathcal{D}_{\Lambda,r}$ .

## 2.8 Trapdoor

Micciancio and Peikert [35] introduced improved trapdoors for the LWE function. A trapdoor is a secret information that helps inverting a one-way function. In this case, that function is  $f_{a,e}(s) = as + e \pmod q$ , the function of the LWE samples. The authors defined trapdoors over  $\mathbb{Z}^n$ . As this work uses them over a ring of integers  $R$  of a number field  $K$ , we adapt the definition to this setting.

**Definition 14.** (*g-trapdoor, adapted from [35]*) Let  $R$  be the ring of integers of a number field  $K$  and  $q$  a modulo. Let  $m > \omega \geq 1$  be integers,  $a \in R_q^m$  and  $g \in R_q^\omega$ . A *g-trapdoor*  $T \in R^{\omega \times (m-\omega)}$  is a matrix for which  $(T, I)a = gh$  for some invertible  $h \in R_q$ . We refer to  $h$  as the tag or label of the trapdoor.

The vector  $g$  is called *gadget vector*. It is chosen in such a way that inverting  $gs + e$  is very easy as is shown in Section 2.8.1. Let

$$g = \begin{pmatrix} 1 \\ 2 \\ 4 \\ \vdots \\ 2^{k-1} \end{pmatrix} \in R_q^k, k = \lceil \log q \rceil.$$

Note that the polynomials in  $g$  are all just constant polynomials.

For a prime  $q = \sum_{i=1}^k q_i 2^{i-1}$ , the following matrix  $B_g$  is a basis of the lattice  $\Lambda_q^\perp(g)$  [35].

$$B_g = \begin{pmatrix} 2 & & & & & q_1 \\ -1 & 2 & & & & q_2 \\ & -1 & & & & q_3 \\ & & \ddots & \ddots & & \vdots \\ & & & -1 & 2 & q_{k-1} \\ & & & & -1 & q_k \end{pmatrix} \in R_q^{k \times k} \quad (2.3)$$

Hereby, each  $q_i \in \{0, 1\}$  is interpreted as constant polynomial.

### 2.8.1 Inverting LWE

Inverting the LWE function  $f_{g,e}(s) = gs + e \pmod q$  works, if  $e \in \mathcal{P}_{1/2}(q \cdot B^{-T})$ , where  $B$  is either  $B_g$  or its Gram-Schmidt orthogonalization [35, 29].

We describe inverting for a power of two modulo  $q$  as it is more intuitive. By working in the power basis  $1, x, \dots, x^{n-1}$ , each coefficient can be considered independently of each other, as each entry of  $g$  is a constant polynomial and  $e$  is added. Therefore, we only describe it for one coefficient  $d$ . Let  $d_{e_j}$  be the corresponding coefficient of  $e_j$ . The idea is to determine each bit of  $d$  individually, starting at the least significant one. Let  $d = d_k \dots d_1$  be the binary representation of  $d$ . The last sample is equal to

$$2^{k-1}d + d_{e_{k-1}} \pmod q = d_1 \cdot 2^{k-1} + d_{e_{k-1}}.$$

As  $d_{e_{k-1}} < 2^{k-1}$  by assumption on  $e$ , the most significant bit of the last sample is equal to  $d_1$ . For the next bit  $d_2$  consider the previous sample

$$2^{k-2}d + d_{e_{k-2}} \pmod q = d_2 \cdot 2^{k-1} + d_1 \cdot 2^{k-2} + d_{e_{k-2}}.$$

As  $d_1$  is known by the previous calculation, subtracting it yields the same equation as before, and  $d_2$  can be extracted. Continuing this for the other samples yields  $d$ . Performing this for every coefficient yields  $s$ .

This inversion can be adapted for a modulo which is not a power of two [35]. Lai, Cheung, and Chow [29] described additional details in the case of power of two cyclotomics. Based on the inversion of  $gs + e$ , the *Invert* function recovers  $s$  from  $c = as + e$  with the help of a  $g$ -trapdoor  $T$  with tag  $h$ . It is defined as follows.

*Invert*( $c, T, h$ ):

- $b = (T, I) \cdot c$
- Calculate  $\hat{s}$  with  $b = g\hat{s} + \hat{e}$  for some small  $\hat{e}$  as described above.
- $s = h^{-1}\hat{s}$
- Return  $s$

The *Invert* function returns the correct  $s$ , if the error  $\hat{e}$  is suitable small as stated in the following lemma.

**Lemma 16.** *Let  $R$  be the ring of integers of a number field  $K$  and  $q$  a modulo. Let  $m > \omega$  be integers,  $a_0 \in R_q^m$  and  $g \in R_q^k$  the gadget vector above. Let  $T$  be a  $g$ -trapdoor for  $a_0$  with tag  $h \in R_q$ . Let  $s \in R_q$  be a secret and  $c = a_0s + e$  for some error  $e \in R_q^m$ . Let  $B$  be  $B_g$  as in Eq. (2.3) or its Gram-Schmidt orthogonalization. If  $(T, I_\omega)e \in \mathcal{P}_{1/2}(q \cdot B^{-T})$ , then  $\text{Invert}(c, T, h)$  returns the correct  $s$ .*

*Proof.* The correctness of the *Invert* function relies on the inversion of  $g\hat{s} + \hat{e}$  in the second step. More specifically, we have  $\hat{e} = (T, I_\omega)e$  as

$$b = (T, I_\omega)c = (T, I_\omega)a_0s + (T, I_\omega)e = ghs + (T, I_\omega)e.$$

If  $\hat{e} \in \mathcal{P}_{1/2}(q \cdot B^{-T})$ , then the correct  $\hat{s} = hs$  is calculated [35, 29]. As  $h$  is invertible,  $s = h^{-1}\hat{s}$  is uniquely determined by  $\hat{s}$ .  $\square$

Note that the *Invert* function only calculates the secret  $s$ . The error  $e$  can then be calculated as  $e = c - as$ . To generate an  $a$  with a corresponding  $g$ -trapdoor  $T$  with tag  $h$ , choose  $a' \in R_q^\omega$ , a tag  $h$  and a trapdoor  $T$  and set  $a = (a', hg - Ta')$ .

## 2.9 Full-Rank Difference Encoding

For the security, the construction requires an embedding of the sender IDs into the ring of integers that fulfills a notion that is stronger than injective. Two images are not only different, but invertible and even the difference of two images is invertible. This mapping is known as a full-rank difference encoding (FRD). The formal definition is as follows.

**Definition 15.** *(Full-rank difference encoding, [9] Definition 3) Let  $S$  be a sender identity space,  $R$  the ring of integers of a number field  $K$  and  $q$  a modulo. A function  $H : S \rightarrow R_q$  is a full-rank difference encoding (FRD), iff it fulfills the following three properties:*

1. *For all  $u \in S$ ,  $H(u)$  is invertible*
2. *For all  $u, v \in S$  with  $u \neq v$ , the element  $H(u) - H(v)$  is invertible*
3.  *$H$  is computable in polynomial time*

Bert et al. [9] showed a FRD onto cyclotomics. The underlying idea is to find a set of invertible elements, which is closed under subtraction. For this, they used the following theorem proven by Lyubashevsky and Seiler [32].

**Theorem 2.** ([9], Theorem 2) Let  $n \geq r > 1$  be powers of 2, and  $q$  a prime such that  $q = 2r + 1 \pmod{4r}$ . Then the cyclotomic polynomial  $X^n + 1$  factors in  $\mathbb{Z}_q[X]$  as  $X^n + 1 = \prod_{i=1}^r (X^{n/r} - s_i)$ , for some distinct  $s_i \in \mathbb{Z}_q^*$  such that the  $(X^{n/r} - s_i)$  are irreducible in  $\mathbb{Z}_q[X]$ . Moreover, any  $f \in R_q$  such that  $0 < \|f\|_{k,\infty} < q^{1/r}/\sqrt{r}$  or  $0 < \|f\|_k < q^{1/r}$  is invertible.

The theorem gives sufficient conditions for invertible polynomials, given that the modulo  $q$  fulfills  $q = 2r + 1 \pmod{4r}$  for  $r$ , a power of 2 smaller than  $n$ . The authors listed two methods to realize the set of invertible polynomials. Either choose small coefficients or polynomials of small degree. They opted for the latter option, leading to the following FRD.

**Lemma 17.** ([9], Proposition 1) Let  $n \geq r > 1$  be powers of 2,  $q$  be a prime such that  $q = 2r + 1 \pmod{4r}$  and  $S = \mathbb{Z}_q^{n/r} \setminus \{0\}$ . Then the following map  $H : S \rightarrow R_q$  is an FRD.

$$(m_1, \dots, m_{n/r}) \mapsto \sum_{i=1}^{n/r} m_i X^{i-1}$$

The idea of the proof is to consider the polynomial  $H(s)$  in the CRT basis. By Theorem 2,  $f = X^n + 1$  factors into  $f_i = X^{n/r} - s_i$ . Therefore,  $\mathbb{Z}_q[X]/f$  is isomorph to  $\mathbb{Z}_q[X]/f_1 \times \dots \times \mathbb{Z}_q[X]/f_r$ , where  $\times$  denotes the tensor product. Thus, each element of  $\mathbb{Z}_q[X]/(X^n + 1)$  can be uniquely represented by the residues modulo  $f_i$ . Because of the isomorphism, each element in  $\mathbb{Z}_q[X]/f$  is invertible iff each residue mod  $f_i$  is. As  $\mathbb{Z}_q[X]/f_i$  is a field for every  $i$ , elements in it are invertible if and only if they are not zero. As  $H(s)$  has degree smaller than  $n/r$  and is not zero, the modulo  $f_i$  does not change anything and therefore the residues are all not zero, making  $H(s)$  invertible. For two senders  $u, v \in S$ , the difference  $H(u) - H(v)$  is again a non-zero polynomial of degree smaller than  $n/r$  and therefore invertible.

## 2.10 Universal Composability

Universal composability (UC) is a framework introduced by Canetti [14], which aims to decompose large protocols into modular components. The underlying setting is multi-party, which means that there are multiple parties that follow a protocol to achieve a certain functionality. Some of these parties are corrupted by an adversary  $\mathcal{A}$ , which aims to disrupt the functionality intended by the protocol. In the case of secure channels this means gaining information about messages that are sent between two honest parties or send messages in the name of honest parties. In this work, we consider an active adversary with static corruption. The former means that the adversary fully controls the corrupted parties and can deviate from the protocol. The latter states that the adversary chooses the corrupted parties before the protocol starts and may not adapt the set of corrupted parties throughout the execution. To simulate the real world application, there is an environment  $\mathcal{Z}$  that instructs each party what it wants to achieve. In the case of the secure channel, it instructs parties which messages should be sent to which party.

The basic idea of UC is to define *ideal functionalities*, which represent a trusted third party in an ideal setting. Other protocols then use these ideal functionalities to achieve more complex functionalities. When instantiating the more complex protocols, the used ideal functionalities can be instantiated with any protocol that UC-realizes them. UC-realize intuitively means that no matter what the adversary does, the complete interaction between all parties can be simulated with only publicly available information. That implies that the adversary does not learn anything it is not intended to. More formally it is defined as follows.

**Definition 16.** (UC-realize) A protocol  $\pi$  UC-realizes an ideal functionality  $\mathcal{F}$ , iff for every adversary  $\mathcal{A}$ , there is a simulator  $\mathcal{S}$ , such that no environment  $\mathcal{Z}$  can distinguish, whether it is interacting with  $\pi$  and  $\mathcal{A}$  or  $\mathcal{F}$  and  $\mathcal{S}$ . It is written as  $\pi \geq_{UC} \mathcal{F}$ .

A common ideal functionality is  $\mathcal{F}_{AUTH}$  shown in Fig. 2.5. It defines authenticated communication, meaning that parties can send messages and the receiving party knows the sender ID. This stops an adversary from sending messages in the name of other parties. However, anyone and especially the adversary can read any sent messages. The adversary can block any message and sent it in his own name.

#### Functionality $\mathcal{F}_{AUTH}$

**Provides:**

Single-receiver single-message single-sender authenticated message transfer with constant message size.

**Behavior:**

- Upon invocation with input (**send**,  $sid$ ,  $R$ ,  $m$ ) from some party  $S$ , send backdoor message (**send**,  $sid$ ,  $S$ ,  $R$ ,  $m$ ) to the adversary  $\mathcal{A}$ .
- Upon receiving (**send ok**,  $sid$ ) from adversary  $\mathcal{A}$ : If not yet generated output, then output (**sent**,  $sid$ ,  $S$ ,  $R$ ,  $m$ ) to  $R$
- Ignore all further inputs

**Figure 2.5:** The Ideal  $\mathcal{F}_{AUTH}$  Functionality

Canetti [15] showed how to realize  $\mathcal{F}_{AUTH}$  with a signature scheme and a certificate authority (CA). For this, they introduced the ideal functionality  $\mathcal{F}_{CERT}$ , which provides signatures that are bound to the sender. It is depicted in Fig. 2.6. Note that the notation is adapted to be consistent throughout this work. The Protocol  $\pi_{AUTH}^{\mathcal{F}_{CERT}}$ , which realizes  $\mathcal{F}_{AUTH}$  using  $\mathcal{F}_{CERT}$  is depicted in Fig. 2.7. Intuitively, it uses  $\mathcal{F}_{CERT}$  to sign each message when sending and verify the authenticity upon receiving. This puts the main task of verifying the authenticity of the signature onto the protocol realizing  $\mathcal{F}_{CERT}$ . For this, the authors constructed the protocol  $\pi_{CERT}^{\mathcal{F}_{CA}}$ . In their work, this protocol uses the ideal functionalities  $\mathcal{F}_{SIG}$ , which realizes signatures and  $\mathcal{F}_{CA}$  for the CA. The former is replaced with an EUF-CMA secure signature

**Functionality  $\mathcal{F}_{CERT}$** **Provides:**

Signatures that are bound to the parties.

**Behavior:**

- Upon receiving (**Sign**,  $sid = (S, sid')$ ,  $m$ ) from some party S, send (**Sign**,  $sid$ ,  $m$ ) to the adversary. Upon receiving (**Signature**,  $sid$ ,  $m$ ,  $\sigma$ ) from the adversary, verify that no entry  $(m, \sigma, 0)$  is recorded. If it is, output an error message to S and halt. Else, output (**Signature**,  $sid$ ,  $m$ ,  $\sigma$ ) to S and record entry  $(m, \sigma, 1)$ .
- Upon receiving a value (**Verify**,  $sid$ ,  $m$ ,  $\sigma$ ) from some Party P, hand (**Verify**,  $sid$ ,  $m$ ,  $\sigma$ ) to the adversary. Upon receiving (**Verified**,  $sid$ ,  $m$ ,  $\phi$ ) from the adversary, do:
  1. If  $(m, \sigma, 1)$  is recorded then set  $f = 1$ .
  2. Else, if the signer is not corrupted and no entry  $(m, \sigma', 1)$  for any  $\sigma'$  is recorded, then set  $f = 0$  and record the entry  $(m, \sigma, 0)$ .
  3. Else, if there is an entry  $(m, \sigma, f')$  is recorded, then set  $f = f'$ .
  4. Else, set  $f = \phi$  and record the entry  $(m, \sigma', \phi)$ .

Output (**Verified**,  $sid$ ,  $m$ ,  $f$ ) to P.

**Figure 2.6:** The Ideal  $\mathcal{F}_{CERT}$  Functionality [15]

**Protocol  $\pi_{AUTH}^{\mathcal{F}_{CERT}}$** **Provides:**

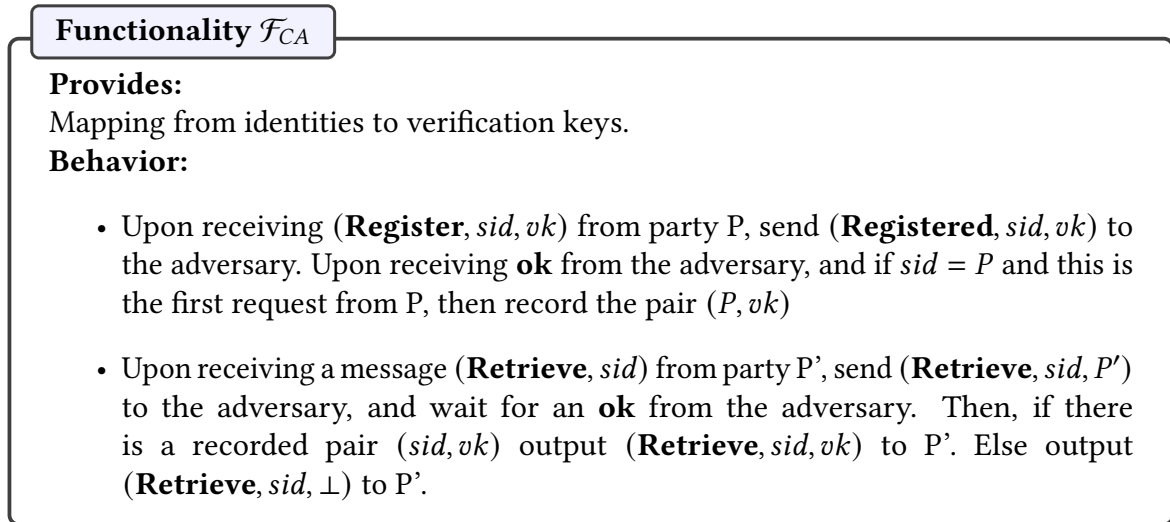
Single-receiver single-message single-sender authenticated message transfer with constant message size.

**Behavior of Party P:**

- Upon receiving (**Send**,  $sid$ ,  $B$ ,  $m$ ), set  $sid' = (P, sid)$ ,  $m' = (m, B)$  and send (**Sign**,  $sid'$ ,  $m'$ ) to  $\mathcal{F}_{CERT}$ . Upon receiving (**Signed**,  $sid'$ ,  $m'$ ,  $s$ ), send  $(sid, P, m, s)$  to B.
- Upon receiving  $(sid, B, m, s)$ , set  $sid' = (B, sid)$ ,  $m' = (m, P)$  and send (**Verify**,  $sid'$ ,  $m'$ ,  $s$ ) to  $\mathcal{F}_{CERT}$  to obtain (**Verified**,  $sid'$ ,  $m'$ ,  $s$ ,  $f$ ). If  $f = 1$ , then output (**Sent**,  $sid$ ,  $P$ ,  $B$ ,  $m$ ) and halt. Else halt with no output.

**Figure 2.7:** The Protocol  $\pi_{AUTH}^{\mathcal{F}_{CERT}}$  Realizing  $\mathcal{F}_{AUTH}$

scheme, which was proven to realize the needed ideal functionality [15]. The latter is depicted in Fig. 2.8 and will remain an ideal functionality throughout this work. It defines the ability to register a public key at a trusted authority and retrieve the public key associated with a party. Fig. 2.9 shows the resulting protocol  $\pi_{CERT}^{\mathcal{F}_{CA}}$ . The idea is to store the verification key of the sending party in the CA. The receiver can then retrieve the verification key of the sender and thus verify that the signature of the message is not only valid but also belongs to the sender ID.



**Figure 2.8:** The Ideal Functionality  $\mathcal{F}_{CA}$

The protocol  $\pi_{SIG}$  realizing  $\mathcal{F}_{SIG}$  is adapted to allow for multiple messages being signed by the same signing key. Without this adaptation, a new signing key would be generated for each message, being highly inefficient. Canetti and Rabin [17] showed that this can be achieved by creating the new protocol  $\hat{\rho}$ , which simulates multiple instances of  $\pi_{SIG}$ . Instead of receiving a single session identifier ( $sid$ ), which changes with each call to  $\pi_{SIG}$ ,  $\hat{\rho}$  receives two session identifiers. The  $sid$ , which determines the instantiation of  $\hat{\rho}$  and the subsession identifier ( $ssid$ ), which indicates which instantiation of  $\pi_{SIG}$  should be called. As  $\hat{\rho}$  is only instantiated once throughout the whole protocol, the  $sid$  is constant and not really used. Therefore, it is omitted throughout this work for better readability and the given  $sid_{AUTH}$ , which changes, is the identifier to indicate which instantiation of  $\pi_{SIG}$  should be called. For multiple  $sid_{AUTH}$ ,  $\hat{\rho}$  does not really instantiate new instances of  $\pi_{SIG}$ , as this would lead to many signing keys. Instead, the same signing key can be used if the  $sid_{AUTH}$  is signed in addition to the message [17]. Of course, each party has its own signing key.

Another important fact is that the protocols using  $\pi_{AUTH}^{\mathcal{F}_{CERT}}$  have to check, that the same  $sid_{AUTH}$  is not used twice to prevent replay attacks. Canetti [15] described two methods to ensure this. One is to keep a list of used sids and compare with the list when receiving a message. The other option is to negotiate the usable sids at the beginning of the protocol. Concretely, the parties draw nonces at beginning and share them with each other. These shared nonces are then used one after the other. This reduces the amount of stored data but introduces a

**Protocol**  $\pi_{CERT}^{\mathcal{F}_{CA}}$ **Provides:**

Signatures that are bound to the parties.

**Parameters:**

- EUF-CMA secure signature scheme  $\Sigma = (Gen, Sign, Vfy)$

**State of Party P:**

- Keypair  $(vk, sk) \in (PK, SK)$  of own credentials
- Function  $f_{PK} : P \rightarrow PK$  of known public keys

**Behavior of Party P:**

- Upon receiving (**Sign**,  $sid = (P, sid_{AUTH}), m$ )
  1. If  $(vk, sk)$  does not exist, draw  $(sk, vk) \leftarrow Gen(1^\kappa)$ , and send (**Register**,  $P, vk$ ) to  $\mathcal{F}_{CA}$ .
  2. Create  $\sigma \leftarrow Sign(sk, (m, sid_{AUTH}))$ . Output (**Signature**,  $sid, m, \sigma$ )
- Upon receiving (**Verify**,  $sid = (P', sid_{AUTH}), m, \sigma$ ) check if  $f_{PK}(P')$  exists. If it does not send (**Retrieve**,  $P'$ ) to  $\mathcal{F}_{CA}$  to obtain response (**Retrieve**,  $P', vk$ ). If  $vk = \perp$  output (**Verified**,  $sid, m, 0$ ). Else set  $f_{PK}(P') = vk$  and output (**Verified**,  $sid, m, \sigma, Vfy(pk, (m, sid_{AUTH}), \sigma)$ ).

**Figure 2.9:** The Protocol  $\pi_{CERT}^{\mathcal{F}_{CA}}$  Realizing  $\mathcal{F}_{CERT}$  using an EUF-CMA Secure Signature Scheme

chance of error [15]. Throughout this work, we will assume there is a mechanism in place that checks for duplicates and rejects them, when we write "fresh  $sid_{AUTH}$ ".





### 3 Construction

This chapter describes the new construction of a SB-KEM and proves its correctness and security in Theorem 3 and Theorem 4 respectively. The SB-KEM is an adaptation of the one proposed by Benz et al. [7] to the ring setting. This reduces cipher and key sizes. The security proof is based on decision-NRLWE but does not use the ROM. Thus, this SB-KEM contributes to improving secure channels that do not rely on the ROM when used in the KEM-DEM framework.

There are two variants of the construction: the *computational variant* and the *statistical variant*. Both variants are secure against computational attackers, but the public key is either computationally or statistically indistinguishable from uniform, depending on the variant. This yields different bounds on the advantage of attackers on the scheme.

Let  $K = \mathbb{Z}[X]/f(X)$  be a cyclotomic number field with dimension  $n$ . Denote its ring of integers under the canonical embedding as  $H_K \subset H$  and the corresponding lattice as  $\Lambda$ . Let  $B_\Lambda$  be the basis of  $\Lambda$  as in Eq. (2.2), which is also the change of basis matrix from the coefficient embedding to the canonical embedding. Let  $q$  be a modulo,  $k = \lceil \log q \rceil$  and  $H_q = H_K/qH_K$ . Let  $H_q^*$  be the set of invertible elements of  $H_q$ . For a coefficient vector  $x \in \mathbb{Z}_q^n$ , we denote with  $\lfloor x \rfloor_{q/2}$  rounding each coefficient to 0 or  $q/2$ , whichever is closest mod  $q$ .

Additional building blocks are the gadget vector  $g$  with its *Invert* function and Basis  $B_g$  as well as its Gram-Schmidt orthogonalization  $\tilde{B}_g$ , a full-rank difference encoding function FRD, which translates sender IDs to  $H_q$  and a key derivation function KDF. Using the canonical embedding, the gadget vector and FRD described in Section 2.8 and Section 2.9 can be adapted to  $H_q$ .

Let  $t \in \mathbb{R}^+$ , such that for any diagonal matrix  $\Sigma$ ,

$$\delta := \Pr \left[ \|x\|_2 > \|\sqrt{\Sigma}\|_2 \cdot \frac{1}{t} \cdot \sqrt{n} \mid x \leftarrow \mathcal{D}_{\Lambda, \sqrt{\Sigma}} \right] \leq \epsilon(n) \quad (3.1)$$

where  $\epsilon(n)$  is negligible in  $n$ . For  $t \in (0, 1)$ ,  $\delta$  is negligible in  $n$  for any cyclotomic number field as stated in Lemma 14.

For the statistical variant choose an integer  $m \geq 2$ ,  $\beta_T > 2n \cdot q^{1/m+2/(mn)}$  and  $\alpha$  with  $\frac{1}{\alpha} \geq 2\beta_T \cdot \|\tilde{B}_g^T\|_{k, \infty} \cdot \|B_\Lambda^{-1}\|_2 \cdot \sqrt{n(2n+1)m}$  such that  $q \geq \frac{1}{\sqrt{2\pi \cdot \alpha t \beta_T \sqrt{m}}} \cdot \sqrt{n} \cdot \omega(\sqrt{\log n})$ . For the computational variant instead choose  $m = 2$ ,  $\beta_T \in \mathbb{R}$ , the bounds on  $\alpha$  and  $q$  as in the statistical variant, such that decision-NRLWE $_{q, \chi}$  with  $\chi = \mathcal{D}_{\Lambda, t\beta_T}$  is hard.

Recall, that for  $e \in H$ ,  $\Sigma_e$  is the matrix with the squared norms on the diagonal. We define the following SB-KEM  $\Gamma := (\text{gen}, \text{enc}, \text{dec})$ :

gen( $1^\kappa$ ):

- $a_0 \xleftarrow{\$} H_q^*$ ,  $a' \xleftarrow{\$} H_q^{m-1}$ ,  $T \leftarrow \mathcal{D}_{\Lambda, t\beta_T}^{k \times m}$
- $a = (a_0, a') \in H_q^m$
- $a_1 = Ta \in H_q^k$
- Return  $(sk, pk) = (T, (a, a_1))$

enc( $pk = (a, a_1), S$ ):

- $\tilde{e} \leftarrow \mathcal{D}_{\Lambda, tq\alpha}$ ,  $e \leftarrow \mathcal{D}_{\Lambda, tq\alpha}^m$ ,  $seed \xleftarrow{\$} \{0, 1\}^n$
- $\Sigma_f = t^2\beta_T^2(\sum_{i=1}^m \Sigma_{e_i}) + t^2\beta_T^2 m(q\alpha)^2 I_n$
- $\bar{f} \leftarrow \mathcal{D}_{\Lambda, \sqrt{\Sigma_f}}^k$
- $s = B_\Lambda(q/2 \cdot seed) + \tilde{e} \in H_q$
- $c_0 = as + e \in H_q^m$
- $c_1 = (a_1 + FRD(S)g)s + \bar{f} \in H_q^k$
- $K_0 = KDF(seed)$
- Return  $((c_0, c_1), K_0)$

dec( $c = (c_0, c_1), S, sk = T$ ):

- $s = Invert((c_0, c_1), -T, FRD(S))$
- $e = c_0 - as$ ,  $\bar{f} = c_1 - (a_1 + FRD(S)g)s$
- check  $\|e_i\|_2 \leq q\alpha\sqrt{n}$  ( $1 \leq i \leq m$ ), else return  $\perp$
- check  $\|\bar{f}_i\|_2 \leq q\alpha\beta_T n\sqrt{(n+1)m}$  ( $1 \leq i \leq k$ ), else return  $\perp$
- $seed = \frac{2}{q} \cdot \lfloor B_\Lambda^{-1}s \rfloor_{q/2}$
- check  $\|s - \frac{q}{2}B_\Lambda \cdot seed\|_2 \leq q\alpha\sqrt{n}$ , else return  $\perp$
- $K_0 = KDF(seed)$
- Return  $K_0$

To uniformly draw invertible elements in the gen algorithm, there are two approaches. One is rejection sampling, which means to draw an element in  $H_q$ , check whether it is invertible and draw again if it is not. As described in Section 2.3.4, it can be checked whether a polynomial is invertible in  $R_q$  in polynomial time. Elements in  $H_q$  can be checked as well by embedding them with the inverse of  $\sigma$ . An important fact for the runtime is that the fraction of invertible elements in  $H_q$  is polynomial in  $n$  and  $\log q$  as stated in Lemma 3. The other approach requires knowledge of polynomials  $f_i$  that are irreducible in  $\mathbb{Z}_q$  with  $f = \prod_i f_i \pmod q$  and  $f_i \neq f_j (i \neq j)$ . Then, one can sample non-zero polynomials in  $\mathbb{Z}_q[X]/f_i(X)$  for

each  $i$  and use the CRT to map back from  $\mathbb{Z}_q[X]/f_1 \times \dots \times \mathbb{Z}_q[X]/f_r$  to  $R_q$ . The resulting polynomial is invertible by construction. In addition, this distribution is uniform on all invertible polynomials as the set of all invertible polynomials is exactly the combinations of non-zero polynomials in the smaller fields  $\mathbb{Z}_q[X]/f_i$ . The polynomials are embedded into  $H_q$  via  $\sigma$ .

Before starting with correctness and security, we first show some bounds on the smoothing parameter of  $\Lambda$  to use Lemma 12 later.

**Lemma 18.** *Let  $\psi$  be as in Eq. (2.1). For the parameters of  $\Gamma$  and  $\epsilon > 0$ , it holds that  $\eta_\epsilon(\psi(\Lambda)) \leq \sqrt{2\pi} \cdot \sqrt{\Sigma_f}$ ,  $\eta_\epsilon(\psi(\Lambda)) \leq \sqrt{2\pi} \cdot t\beta_T \sqrt{mq\alpha}$  and  $\eta_\epsilon(\psi(\Lambda)) \leq \sqrt{2\pi} \cdot A_{e_v}^{-1} \sqrt{(\Sigma_f^{-1} + (t^2\beta_T^2\Sigma_{e_v})^{-1})^{-1}}$  for  $1 \leq v \leq m$  where  $e_v \neq 0$ .*

*Proof.* First we prove that  $\eta_\epsilon(\psi(\Lambda)) \leq \sqrt{2\pi} \cdot t\beta_T \sqrt{mq\alpha}$ . Using the definition of  $q$  combined with Lemma 8 and Lemma 5 and the fact that  $\Gamma$  uses a cyclotomic number field yields

$$\eta_\epsilon(\psi(\Lambda)) \leq \lambda_n(\psi(\Lambda)) \cdot \omega(\sqrt{\log n}) \leq \sqrt{n} \cdot \omega(\sqrt{\log n}) \leq \sqrt{2\pi} \cdot t\beta_T \sqrt{mq\alpha}.$$

By definition of  $\Sigma_f$  we have

$$\Sigma_f = t^2\beta_T^2 \left( \sum_{i=1}^m \Sigma_{e_i} \right) + t^2\beta_T^2 m(q\alpha)^2 I \succeq t^2\beta_T^2 m(q\alpha)^2 I$$

and therefore

$$\eta_\epsilon(\psi(\Lambda)) \leq \sqrt{2\pi} \cdot t\beta_T \sqrt{mq\alpha} I_n \leq \sqrt{2\pi} \cdot \sqrt{\Sigma_f}.$$

For the last inequality let  $S := A_{e_v}^{-1} \sqrt{(\Sigma_f^{-1} + (t^2\beta_T^2\Sigma_{e_v})^{-1})^{-1}}$ . Let  $1 \leq v \leq m$  be arbitrary but fixed and  $e_v = (r_1, \dots, r_{s_1}, c_1, \bar{c}_1, \dots, c_{s_2}, \bar{c}_{s_2})^T$ ,  $r_j \in \mathbb{R}$ ,  $c_j = a_j + b_j i \in \mathbb{C}$ . Recall that  $A_{e_v}$  is defined as

$$A_{e_v} = \begin{pmatrix} r_1 & & & & & \\ & \ddots & & & & \\ & & r_{s_1} & & & \\ & & & Rot_1 & & \\ & & & & \ddots & \\ & & & & & Rot_{s_2} \end{pmatrix}, Rot_j = \begin{pmatrix} a_j & -b_j \\ b_j & a_j \end{pmatrix},$$

and therefore

$$A_{e_v}^{-1} = \begin{pmatrix} r_1^{-1} & & & & & \\ & \ddots & & & & \\ & & r_{s_1}^{-1} & & & \\ & & & \frac{1}{a_1^2 + b_1^2} Rot_1^T & & \\ & & & & \ddots & \\ & & & & & \frac{1}{a_{s_2}^2 + b_{s_2}^2} Rot_{s_2}^T \end{pmatrix}.$$

Let  $y_j^2 = (\sum_{v=1}^m \Sigma_{e_v})_{jj} + m(q\alpha)^2$ . For  $1 \leq j \leq s_1$  let

$$s_j^r := r_j^{-1} \sqrt{\left( \frac{1}{t^2 \beta_T^2 y_j^2} + \frac{1}{t^2 \beta_T^2 r_j^2} \right)^{-1}}.$$

For  $1 \leq j \leq s_2$ , define the block matrix  $S_j^c$  as

$$S_j^c := \frac{1}{a_j^2 + b_j^2} \cdot \sqrt{\left( \frac{1}{t^2 \beta_T^2 y_j^2} + \frac{1}{t^2 \beta_T^2 (a_j^2 + b_j^2)} \right)^{-1}} \cdot Rot_j^T \in \mathbb{R}^{2 \times 2}.$$

Using that definitions,  $S$  can be written as

$$S = \begin{pmatrix} s_1^r & & & & & & \\ & \ddots & & & & & \\ & & s_{s_1}^r & & & & \\ & & & S_1^c & & & \\ & & & & \ddots & & \\ & & & & & & S_{s_2}^c \end{pmatrix}.$$

With that,  $SS^T$  comes down to  $(s_j^r)^2$  and  $S_j^c(S_j^c)^T$ , which is

$$(s_j^r)^2 = r_j^{-2} \cdot \left( \frac{1}{t^2 \beta_T^2 y_j^2} + \frac{1}{t^2 \beta_T^2 r_j^2} \right)^{-1} = \left( \frac{r_j^2}{t^2 \beta_T^2 y_j^2} + \frac{1}{t^2 \beta_T^2} \right)^{-1} = \frac{t^2 \beta_T^2 y_j^2}{r_j^2 + y_j^2} \stackrel{(*)}{\geq} \frac{t^2 \beta_T^2}{2}$$

and

$$S_j^c(S_j^c)^T = \frac{1}{a_j^2 + b_j^2} \cdot \left( \frac{1}{t^2 \beta_T^2 r_j^2} + \frac{1}{t^2 \beta_T^2 (a_j^2 + b_j^2)} \right)^{-1} \cdot I_2 = \left( \frac{a_j^2 + b_j^2}{t^2 \beta_T^2 r_j^2} + \frac{1}{t^2 \beta_T^2} \right)^{-1} \cdot I_2 \stackrel{(**)}{\geq} \frac{t^2 \beta_T^2}{2} \cdot I_2.$$

The inequalities (\*) and (\*\*) hold because  $r_j^2 \leq y_j^2$  and  $a_j^2 + b_j^2 \leq y_j^2$ , respectively. With that, it holds that

$$SS^T \succeq \frac{t^2 \beta_T^2}{2} I_n.$$

Using Lemma 5 and Lemma 8 with  $t\sqrt{n}$  as  $\omega(\sqrt{\log n})$  function yields

$$\begin{aligned} \eta_\epsilon(\psi(\Lambda)) &\leq \sqrt{n} \cdot t\sqrt{n} I_n \leq t\sqrt{2n} \cdot q^{1/m+2/(mn)} I_n \leq \frac{t\beta_T}{\sqrt{2}} I_n \\ &\leq A_{e_v}^{-1} \sqrt{\left( \Sigma_f^{-1} + (t^2 \beta_T^2 \Sigma_{e_v})^{-1} \right)^{-1}} \leq \sqrt{2\pi} \cdot A_{e_v}^{-1} \sqrt{\left( \Sigma_f^{-1} + (t^2 \beta_T^2 \Sigma_{e_v})^{-1} \right)^{-1}}. \end{aligned}$$

□

The following theorem shows the correctness of the SB-KEM.

**Theorem 3.** *The SB-Kem  $\Gamma = (\text{gen}, \text{enc}, \text{dec})$  is correct with overwhelming probability over the choices of  $e, \tilde{e}, \tilde{f}$ .*

*Proof.* For correctness, three parts have to apply. First, *Invert* needs to return the correct  $s$ . Second, the errors need to be small enough, such that the checks do not fail. Third, the error  $\tilde{e}$  needs to be small enough so that the seed can be recovered from  $s$ .

As stated in Lemma 16, the *Invert* function returns the correct  $s$  if the following equation holds:

$$Te + \tilde{f} \in q \cdot \mathcal{P}_{1/2}(\tilde{B}_G^{-T}) \quad (3.2)$$

where  $\tilde{B}_G$  is the Gram-Schmidt orthogonalization of the basis of  $\Lambda_q^\perp(g)$ . This is equivalent to

$$\| [\tilde{B}_G^T(Te + \tilde{f})]_{k,\infty} \|_\infty / q \leq \frac{1}{2}.$$

According to Lemma 12, each entry of  $Te + \tilde{f}$  is distributed as  $\mathcal{D}_{\Lambda,\Sigma}$ , where  $\Sigma = \Sigma_f + t^2 \beta_T^2 \sum_{i=1}^m \Sigma_{e_i}$ . The bounds on the smoothing parameter hold by Lemma 18. Using the definitions of  $\Sigma_f$  and  $\Sigma_{e_k}$  and

$$r^2 := \beta_T^2 m (q\alpha)^2 + 2\beta_T^2 \cdot \max_j \sum_{i=1}^m |e_{ij}|^2$$

it holds that

$$\Sigma \leq t^2 r^2 I_n$$

where  $\leq$  is on every entry. As each  $e_i$  is drawn by a discrete Gaussian, Eq. (3.1) yields

$$|e_{ij}| \leq \|e_i\|_{c,2} \leq \sqrt{n} q \alpha$$

with overwhelming probability and thus

$$r^2 \leq m(q\alpha)^2 \beta_T^2 + 2mn(q\alpha)^2 \beta_T^2 = (2n+1)m(q\alpha)^2 \beta_T^2.$$

By Eq. (3.1), each element of  $Te + \tilde{f}$  has norm at most  $r\sqrt{n}$  and thus  $\| [Te + \tilde{f}]_{c,2} \|_\infty \leq r\sqrt{n}$  with overwhelming probability and therefore,

$$\begin{aligned} \| [\tilde{B}_G^T(Te + \tilde{f})]_{k,\infty} \|_\infty / q &\leq \| \tilde{B}_G^T \|_{k,\infty} \cdot \| [Te + \tilde{f}]_{k,\infty} \|_\infty / q \\ &\leq \| \tilde{B}_G^T \|_{k,\infty} \cdot \| [Te + \tilde{f}]_{k,2} \|_\infty / q \\ &\leq \| \tilde{B}_G^T \|_{k,\infty} \cdot \| B_\Lambda^{-1} \|_2 \cdot \| [Te + \tilde{f}]_{c,2} \|_\infty / q \\ &\leq \| \tilde{B}_G^T \|_{k,\infty} \cdot \| B_\Lambda^{-1} \|_2 \cdot r\sqrt{n} / q \\ &\leq \| \tilde{B}_G^T \|_{k,\infty} \cdot \| B_\Lambda^{-1} \|_2 \cdot \alpha \beta_T \sqrt{n(2n+1)m} \stackrel{(*)}{\leq} \frac{1}{2}. \end{aligned}$$

The last inequality (\*) holds because  $\frac{1}{\alpha} \geq 2\beta_T \cdot \| \tilde{B}_G^T \|_{k,\infty} \cdot \| B_\Lambda^{-1} \|_2 \cdot \sqrt{n(2n+1)m}$ . Thus, the errors are small enough with overwhelming probability so that Eq. (3.2) holds and *Invert* returns the correct  $s$ .

The checks test the norm of elements of  $H_K$  drawn by a discrete Gaussian. The overwhelming probability for a correct check for  $e_i$  and  $(s - q/2B_\Lambda \cdot \text{seed})$  follow directly from Eq. (3.1) and the variance of  $e$  and  $\tilde{e}$ , respectively. For the check on  $\tilde{f}_i$ , using  $|e_{ij}| \leq q\alpha\sqrt{n}$ , it holds that

$$\Sigma_f \leq t^2(q\alpha)^2\beta_T^2n(n+1)m \cdot I$$

where  $\leq$  is on each entry. Thus, the check is successful with overwhelming probability by Eq. (3.1).

The seed can be recovered from  $s$  with overwhelming probability as by Eq. (3.1) and the variance of  $\tilde{e}$  it follows

$$\|B_\Lambda^{-1}\tilde{e}\|_\infty \leq \|B_\Lambda^{-1}\tilde{e}\|_2 \leq \|B_\Lambda^{-1}\|_2 \cdot \|\tilde{e}\|_{c,2} \leq \|B_\Lambda^{-1}\|_2 \cdot q\alpha\sqrt{n} \leq \frac{q}{4}.$$

□

Before proving the security, we first show that the public key is indistinguishable from uniform, depending on the variant either statistically or computationally.

**Lemma 19.** *Let  $a_0, a', T, k = \lceil \log q \rceil$  be as in  $\Gamma$ . Then, the distribution  $(a, Ta)$  is statistically close to  $(a, u)$  with uniformly distributed  $u$  if the parameters of  $\Gamma$  are chosen according to the statistical variant. If the parameters of  $\Gamma$  are chosen according to the computational variant,  $(a, Ta)$  is computationally indistinguishable from  $(a, u)$  if the decision-NRLWE $_{q,\chi}$  assumption holds for  $\chi = \mathcal{D}_{\Lambda, t\beta_T}$ . Concretely, for any attacker  $\mathcal{A}$  on the indistinguishability, there is an attacker  $\mathcal{A}_{d\text{-NRLWE}}$  on  $d\text{-NRLWE}_{q,\chi}$  with*

$$\text{adv}_{\mathcal{A}}^{\text{IND}}(\kappa) \leq k \cdot \text{adv}_{\mathcal{A}_{d\text{-NRLWE}}}^{d\text{-NRLWE}_{q,\chi}}(\kappa).$$

*Proof.* As  $a_0$  is invertible, multiplying  $(a, Ta) = ((a_0, a'), T(a_0, a'))$  with  $a_0^{-1}$  yields  $((1, a_0^{-1}a'), T(1, a_0^{-1}a'))$ . As  $a'$  is uniformly random,  $a'' := a_0^{-1}a'$  is too. In the statistical variant, each row is statistically indistinguishable from random by Lemma 4 with  $k = 1, l = m$ . In the computational variant, as  $m = 2$  we have  $a'' \in R_q$  and

$$T(1, a'') = \begin{pmatrix} t_{11} + t_{12}a'' \\ \vdots \\ t_{k1} + t_{k2}a'' \end{pmatrix}.$$

We prove with a hybrid proof that  $(a'', w := T(1, a''))$  is indistinguishable from  $(a'', u)$  with uniformly random  $u$ . In game  $H_i$ , the first  $i$  rows of  $w$  are uniformly random instead of  $t_{j1} + t_{j2}a''$ . In the game  $G_0 := H_0$ ,  $w$  is as in the construction and in  $G_1 := H_k$ , it is completely random. If there is an attacker  $\mathcal{A}_i$ , which distinguishes game  $H_{i-1}$  from  $H_i$ , there is an attacker  $\mathcal{A}_{d\text{-NRLWE}}$  on  $d\text{-NRLWE}_{q,\chi}$  with  $\chi = \mathcal{D}_{\Lambda, t\beta_T}$ . Given one NRLWE sample  $(a_{\text{NRLWE}}, b)$ ,

$\mathcal{A}_{d\text{-NRLWE}}$  embeds this sample in the  $i$ -th row and samples everything else as in game  $H_i$ . Specifically, it sets  $a'' = a_{\text{RLWE}}$  and for  $w$  it chooses the vector

$$w = \begin{pmatrix} u_1 \\ \vdots \\ u_{i-1} \\ b \\ t_{i+1,1} + t_{i+1,2}a_{\text{RLWE}} \\ \vdots \\ t_{k,1} + t_{k,2}a_{\text{RLWE}} \end{pmatrix}, u_i \leftarrow \mathcal{U}, t_{ij} \leftarrow \mathcal{D}_{\Lambda, t\beta_T}.$$

Then,  $\mathcal{A}_i$  is run with input  $(a'', w)$ . If  $b = a_{\text{RLWE}}s + e$  for some secret  $s$  and error  $e$ , this simulates game  $H_{i-1}$ . If  $b$  is random instead, it simulates game  $H_i$ . Therefore, for any attacker  $\mathcal{A}_i$  we have

$$\left| \Pr \left[ \mathcal{A}_i^{H_i}(\kappa) = 1 \right] - \Pr \left[ \mathcal{A}_i^{H_{i-1}}(\kappa) = 1 \right] \right| \leq \text{adv}_{\mathcal{A}_{d\text{-NRLWE}}}^{d\text{-NRLWE}_{q,\chi}}(\kappa).$$

With that, we can bound the advantage of every attacker  $\mathcal{A}$  against distinguishing  $G_0$  from  $G_1$  as

$$\begin{aligned} \text{adv}_{\mathcal{A}, G_0, G_1}^{\text{IND}}(\kappa) &= \left| \Pr \left[ \mathcal{A}^{G_1}(\kappa) = 1 \right] - \Pr \left[ \mathcal{A}^{G_0}(\kappa) = 1 \right] \right| \\ &= \left| \Pr \left[ \mathcal{A}^{H_k}(\kappa) = 1 \right] - \Pr \left[ \mathcal{A}^{H_0}(\kappa) = 1 \right] \right| \\ &= \left| \sum_{i=1}^k \Pr \left[ \mathcal{A}^{H_i}(\kappa) = 1 \right] - \Pr \left[ \mathcal{A}^{H_{i-1}}(\kappa) = 1 \right] \right| \\ &\leq \sum_{i=1}^k \left| \Pr \left[ \mathcal{A}^{H_i}(\kappa) = 1 \right] - \Pr \left[ \mathcal{A}^{H_{i-1}}(\kappa) = 1 \right] \right| \\ &\leq k \cdot \text{adv}_{\mathcal{A}_{d\text{-NRLWE}}}^{d\text{-NRLWE}}(\kappa). \end{aligned}$$

□

With that, the following theorem proves the security of the SB-KEM based on decision-NRLWE.

**Theorem 4.** *The statistical variant of the SB-Kem  $\Gamma = (\text{gen}, \text{enc}, \text{dec})$  with parameters  $(n, q, k, \alpha, \beta_T)$  is  $\text{IND-SB-CPA}_{\text{SB-KEM}}$  secure given the decision- $\text{NRLWE}_{q,\chi}$  assumption with  $\chi = \mathcal{D}_{\Lambda, tq\alpha}$ . Concretely, if there is an attacker  $\mathcal{A}$  on the security of the SB-KEM, there exists an attacker  $\mathcal{A}_{\text{KDF}}$  on the KDF and  $\mathcal{A}_{\text{NRLWE}}$  on the decision- $\text{NRLWE}_{q,\chi}$  assumption with*

$$\text{Adv}_{\mathcal{A}, \Gamma}^{\text{IND-SB-CPA}_{\text{SB-KEM}}}(\kappa) \leq \text{Adv}_{\mathcal{A}_{\text{KDF}}, \text{KDF}}^{\text{KDF}}(\kappa) + \text{poly}(n, \log q) \cdot \text{Adv}_{\mathcal{A}_{\text{NRLWE}}}^{d\text{-NRLWE}_{q,\chi}}(\kappa) + \epsilon$$

where  $\epsilon$  is negligible in  $\kappa$ .



The computational variant is  $IND\text{-}SB\text{-}CPA_{SB\text{-}KEM}$  secure given the decision- $NRLWE_{q,t\beta_T}$  assumption in addition to those of the statistical variant. Concretely, for every attacker  $\mathcal{A}$  on the security of the SB-KEM, there exists an attacker  $\mathcal{A}_{KDF}$  on the KDF,  $\mathcal{A}_{NRLWE}$  on the decision- $NRLWE_{q,\chi}$  assumption and  $\mathcal{A}'_{NRLWE}$  on the decision- $NRLWE_{q,t\beta_T}$  with

$$\begin{aligned} Adv_{\mathcal{A},\Gamma}^{IND\text{-}SB\text{-}CPA_{SB\text{-}KEM}}(\kappa) &\leq Adv_{\mathcal{A}_{KDF},KDF}^{KDF}(\kappa) + poly(n, \log q) \cdot Adv_{\mathcal{A}_{NRLWE}}^{d\text{-}NRLWE_{q,\chi}}(\kappa) \\ &\quad + k \cdot Adv_{\mathcal{A}'_{NRLWE}}^{d\text{-}NRLWE_{q,t\beta_T}}(\kappa) + \epsilon \end{aligned}$$

where  $\epsilon$  is negligible in  $\kappa$ .

*Proof.* We show the security through a series of game hops. The first game is the  $IND\text{-}SB\text{-}CPA_{SB\text{-}KEM}$  game and in the last game, the view of an attacker is independent of the challenge bit. An attacker on the security of the SB-KEM can in particular distinguish between the first and the last game.

**Game 0:** This is the  $IND\text{-}SB\text{-}CPA_{SB\text{-}KEM}$  game as in Fig. 2.3.

**Game 1:** For this game, the generation of the public key is changed. Specifically,  $a_1$  is changed to  $Ta - FRD(S)g$  instead of just  $Ta$ . As  $a_1$  is indistinguishable from uniform by Lemma 19, so too is  $Ta - FRD(S)g$ . Therefore, the two distributions are indistinguishable. For the statistical variant, this indistinguishability is statistically and thus, the advantage of any attacker is negligible. In case of the computational variant, if there is an attacker  $\mathcal{A}$ , which distinguishes game 0 from game 1, there is an attacker  $\mathcal{A}_{d\text{-}NRLWE}$  on decision- $NRLWE_{q,t\beta_T}$  with

$$Adv_{\mathcal{A},g_0,g_1}^{IND}(\kappa) \leq k \cdot Adv_{\mathcal{A}_{d\text{-}NRLWE}}^{d\text{-}NRLWE_{q,t\beta_T}}(\kappa)$$

as stated in Lemma 19. Also, for  $S' \neq S$ ,  $FRD(S) - FRD(S')$  is invertible by definition of the FRD and therefore, decrypting ciphertexts for other senders is still possible. For the sender  $S$  it is not possible, but this request is not allowed by definition of  $IND\text{-}SB\text{-}CPA_{SB\text{-}KEM}$ .

**Game 2:** Here, the challenge cipher is changed. Instead of  $(c_0^*, c_1^*)$  as in game 1, we draw  $\bar{c} \in H_q^m$  uniformly random and set  $c_0^* = \bar{c} + a(B_\Lambda(q/2 \cdot seed))$ . For  $c_1^*$  we draw  $\bar{e} \leftarrow \mathcal{D}_{\Lambda,t\beta_T q \alpha \sqrt{m}}^k$  and set  $c_1^* = (Tc_0^*) + \bar{e}$ . To show that this is indistinguishable from game 1, we use a reduction to decision- $NRLWE_{q,\chi}$  by constructing an attacker  $\mathcal{A}_{NRLWE}$ . Given a vector of  $m$  samples  $(a_{RLWE}, b)$ ,  $\mathcal{A}_{NRLWE}$  checks, whether any  $a_i$  in the samples is invertible. This is possible in polynomial time as described in Section 2.3.4. If non exist, it aborts and guesses the challenge bit resulting in a win probability of one half. Otherwise,  $\mathcal{A}_{NRLWE}$  simulates game 2, inserting the samples as follows. It sets the invertible  $a_i$  as  $a_0$  and the others as  $a'$ . These combined yield the public key  $a = (a_0, a')$  and  $a_1 = Ta - FRD(S)g$ . Instead of drawing  $\bar{c}$  randomly,  $\mathcal{A}_{NRLWE}$  sets it to  $b$ , setting  $b_i$  as first element to match the order of  $a$ .

With the resorted samples  $(a'_{RLWE}, b')$ , if  $b' = a'_{RLWE}x + y$  for some secret  $x$  and error  $y$ , then  $(c_0^*, c_1^*)$  is distributed as in game 1, as

$$c_0^* = b' + a'_{RLWE}B_\Lambda(q/2 \cdot \text{seed}) = a'_{RLWE}(x + B_\Lambda(q/2 \cdot \text{seed})) + y \sim as + e \quad (3.3)$$

and

$$\begin{aligned} c_1^* &= Tc_0^* + \bar{e} \stackrel{3.3}{\sim} T(as + e) + \bar{e} = Tas + Te + \bar{e} \\ &= (a_1 + FRD(S)g)s + Te + \bar{e} \stackrel{(*)}{\sim} (a_1 + FRD(S)g)s + \tilde{e}. \end{aligned}$$

For  $(*)$  to hold, we need that  $Te + \bar{e} \sim \tilde{e}$ . By Lemma 12,  $Te + \bar{e}$  is distributed as  $D_{\Lambda, \sqrt{\Sigma}}^k$  with  $\Sigma = t^2\beta_T^2(\sum_{i=1}^m \Sigma_{e_i}) + t^2m(q\alpha)^2$ , which is the distribution of  $\tilde{e}$ . The bounds on the smoothing parameter hold by Lemma 18.

If  $b$  is random, then  $\bar{c}$  is as well and  $\mathcal{A}_{NRLWE}$  simulates game 2. So if there is an invertible  $a_i$  in the samples,  $\mathcal{A}_{NRLWE}$  perfectly simulates one of the games. As the fraction of invertible elements in cyclotomics is polynomial in  $\log q$  and  $n$  as stated in Lemma 3, the chance that any of the  $m$  samples has an invertible  $a$  is  $1 - (1 - \text{poly}(n, \log q))^m$ . Therefore, the advantage of  $\mathcal{A}_{NRLWE}$  is

$$\text{adv}_{\mathcal{A}_{NRLWE}}^{d-NRLWE}(\kappa) = \frac{1}{1 - (1 - \text{poly}(n, \log q))^m} \text{adv}_{\mathcal{A}_{1/2, g^1, g^2}}^{IND}(\kappa)$$

where  $\mathcal{A}_{1/2}$  distinguishes between game 1 and 2.

**Game 3:** Now,  $c_0^*$  is drawn uniformly random from  $H_q^m$ , independently from the seed. As  $\bar{c}$  acted as one-time pad on  $a(B(q/2 \cdot \text{seed}))$ , the statistical view of the adversary does not change.

**Game 4:** Lastly,  $K_0$  is drawn uniformly random. As an attacker has no information about the seed, an attacker who can distinguish between game 3 and game 4 is an attacker on the KDF.

In game 4, an attacker has to distinguish between two randomly drawn  $K_0$  and  $K_1$ , while the rest of the view is independent of these keys. Therefore, any attacker on game 4 has an advantage of zero. An attacker on the security of the SB-KEM can in particular distinguish between game 0 and game 4. However, the advantage of distinguishing these two is bound by the bounds described in the games.

□

### 3.1 Concrete Parameters

This chapter describes concrete parameter choices for the SB-Kem  $\Gamma$  defined in Chapter 3. For the cyclotomic number field  $K$  we choose the  $(2n)$ -th cyclotomic number field, where  $n$  is a power of two as these have a nice canonical embedding. Therefore, the error distributions

translate nicely into the coefficient embedding as shown in Lemma 15. Additional parameters are the second dimension  $m$ , a modulo  $q$ , scaling  $t$ ,  $\beta_T$  and  $\alpha$ . The scaling factor  $t$  impacts the probability of wrong decryption. It depends on the probability of a ring element drawn by a discrete Gaussian with index  $S$ , to have norm greater than  $1/t \cdot \sqrt{n} \cdot \|S\|_2$ , formally

$$\delta := \Pr \left[ \|x\|_2 > \frac{1}{t} \cdot \|S\|_2 \cdot \sqrt{n} \mid x \leftarrow \mathcal{D}_{\Lambda, S} \right]$$

where  $\Lambda = \sigma(R)$  is the lattice of the canonically embedded ring of integers of  $K$ . By Lemma 14 we have

$$\delta < \left( \frac{1}{t} e^{-\frac{1}{2t^2}} \sqrt{e} \right)^n.$$

For concrete parameters,  $\delta$  is a number instead of a function, so the concept of negligible does not apply. Instead, the goal is to achieve that  $\delta$  is small enough. We choose

$$\delta \leq 2^{-64}.$$

This yields the values for  $t$  depending on the dimension  $n$  summarized in Table 3.1.

n	512	1024	2048
t	0.7216	0.7996	0.8566

**Table 3.1:** The values for  $t$  depending on the dimension  $n$  to achieve  $\delta \leq 2^{-64}$

The modulo  $q$  is the most decisive factor besides  $n$ , whether there are any suitable parameter sets. It should be as small as possible, but if it is too small,  $\alpha$  becomes too small such that the underlying NRLWE becomes too easy. The FRD requires that  $q = 2r + 1 \pmod{4r}$  for some power of two  $r$ . The parameter  $r$  influences the number of available sender IDs, which is  $q^{n/r} - 1$ . This limit comes from the FRD. On the other hand, as  $\phi_{2n} = \prod_{i=1}^r (X^{n/r} - s_i)$  for some  $s_i \in \mathbb{Z}_q^*$  as stated by Theorem 2, the calculations in  $R_q$  can be performed in the fields  $\mathbb{Z}_q[X]/(X^{n/r} - s_i)$ , improving performance for larger values of  $r$  [9]. We choose  $r = 16$ . The value for  $q$  is found by trial and error. It is set to a value and adjusted, depending on whether there are parameter sets.

The parameter  $\alpha$  is chosen as small as possible, which is  $\frac{1}{\alpha} = 2\beta_T \cdot \|\tilde{B}_G^T\|_{k, \infty} \cdot \|B_\Lambda^{-1}\|_2 \cdot \sqrt{n(2n+1)m}$ . The norm  $\|\tilde{B}_G^T\|_{k, \infty}$  was determined numerically and yielded  $\|\tilde{B}_G^T\|_{k, \infty} = 4$  for every  $q$  when the orthogonalization is performed in forward order. For the statistical variant we choose  $\beta_T = 2n \cdot q^{(n+2)/(nm)}$ , the smallest value possible. The parameter  $m$  is balanced with  $q$  such that decision-NRLWE $_{q,tq\alpha}$  is hard. Increasing  $m$  increases  $q\alpha$ , which makes the problem harder, but also worsens the performance by increasing key and cipher sizes. In the computational variant,  $\beta_T$  can be chosen more freely, but needs to be large enough such that decision-NRLWE $_{q,t\beta_T}$  is hard. The parameter  $m$  is set to 2, as increasing it only worsens performance. For both variants the condition

$$q > \frac{1}{\sqrt{2\pi} \cdot \alpha t \beta_T \sqrt{m}} \cdot \sqrt{n} \cdot \sqrt{\log n}$$

needs to be checked after setting the parameters.

### 3.1.1 Concrete Ring-LWE hardness estimation

The main decisive factor for the parameters are the underlying NRLWE assumptions. In case of the statistical variant this is only d-NRLWE $_{q,t\alpha q}$ . For the computational variant, d-NRLWE $_{q,t\beta_T}$  needs to be hard as well. To estimate the hardness of NRLWE, we use the lattice estimator [2], a tool frequently used to estimate the hardness of RLWE instances. Concretely, we use commit "bfd74e" and the cost model "MINZOV" [34]. Although the estimator only estimates hardness of LWE, the assumption is that there are no attacks, which leverage the addition structure of the ring. The lattice estimator needs a distribution  $\chi$  in the coefficient embedding, but our definition of NRLWE is in the canonical embedding. For power of two cyclotomics, the discrete Gaussian can be embedded into the coefficient embedding by scaling the parameter by  $1/\sqrt{n}$  as shown in Lemma 15. Therefore, we define the adjusted parameters  $\alpha_k := \alpha/\sqrt{n}$  and  $\beta_{Tk} := \beta_T/\sqrt{n}$ .

Crockett and Peikert [21] analyzed many different parameter sets for LWE in terms of their difficulty. However, for the relatively small values of  $q$  in the analyzed sets, the width of the Gaussian for the errors would be too small in our construction. Therefore, none of their parameter sets can be used for this construction. Bossuat et al. [12] proposed parameter sets for homomorphic encryption based on Ring LWE. These have larger modulo and are therefore better suited for this construction.

The standard deviation for the error distribution is the most controversial part. From a theoretical perspective, Peikert [39] proved instances to be secure against multiple attacks if the standard deviation  $r$  in the dual ring  $R^\vee$  is greater than two. For the ring of integers of power of two cyclotomics of dimension  $n$ , we have  $R = nR^\vee$  [39] and therefore, this corresponds to an standard deviation of  $2n$ . In practice, the error distribution is chosen much narrower. For example, the guidelines for homomorphic encryption chose 3.19 independently of  $n$  [12]. For parameter sets taken from the guidelines and more optimistic sets we aim the standard deviation to be in the order of 3.19. For completeness, there are two parameter sets with the more theoretical standard deviation of  $2n$  as well.

### 3.1.2 Parameter Sets

Table 3.2 shows the different parameter sets. The values  $t\beta_{Tk}$  and  $tq\alpha_k$  are the width of the discrete Gaussians for T and the errors respectively, when drawing them in the coefficient embedding, meaning drawing from  $\mathcal{D}_{\mathbb{Z}^n,s}$ . The column C/S indicates whether the set is for the computational variant (C) or the statistical variant (S). In the column Seclvl, the number of bits of security is provided. That means, an attacker on d-NRLWE $_{q,\chi}$  requires computational power and time equivalent to a brute force attack on a key with that amount of bits. A security level with an L at the end is only determined by the lattice estimator. Hereby, the metric used is ring operations (rop). Those with a P are for parameter sets that have dimension, modulo and width as proposed by Bossuat et al. [12]. These security levels are also confirmed by the lattice estimator. Note, that in the security proof of the scheme, there is a polynomial advantage gain for an attacker on the scheme compared to an attacker on NRLWE. However, this influences the success probability of the attacker, not its runtime.

$n$	$q$ ( $\log_2 q$ )	$m$	$t\beta_{Tk}$	$tq\alpha_k$	C/S	SecLvl	size(pk) = size(c)	size(sk)
512	0xF3d21 (20)	2	1.67	1.679	C	87 L	28.2 KB	51.2 KB
1024	0xF69A1 (20)	2	1.1	1.1	C	164 L	56.3 KB	102 KB
1024	0x31F7E1 (22)	2	1.996	2.16	C	165 L	67.6 KB	124 KB
1024	0x32C118A19E1 (42)	2	2048	2076	C	146 L	237 KB	452 KB
1024	0x1E36050A42A1 (45)	6	9222	2535	S	130 L	294 KB	1.56 MB
2048	0x1F24D205A1 (37)	5	13056	3.206	S	192 P	397 KB	1.75 MB

**Table 3.2:** Concrete parameters for the SB-Kem  $\Gamma$  with the resulting sizes of the public and secret keys as well as the ciphers. C/S stands for computational or statistical variant. SecLvl indicates the bit-security of the set.

The last two columns contain the sizes of the public and private key and the cipher. These numbers are calculated based on the assumption that an element in  $\mathbb{Z}_q$  requires  $\log q$  bits. As the size of the public key is the same as the one of the cipher, they are combined into one column to save space.

As can be seen, the computational variant has much better key sizes than the statistical variant as  $q$  and  $m$  can be chosen smaller for the same  $n$  and  $q\alpha$  because  $\beta_T$  is much smaller. The fourth and fifth parameters sets have the more theoretical width of  $2n$ . The fourth is computational but significant worse than the other computational variants. Considering the statistical variant, the key sizes are similar, though the estimated security of the fifth is worse due to the larger  $q$ .

### 3.1.3 Comparison to existing KEMs

This section compares the key and cipher sizes of the new SB-KEM to existing KEMs. As this work focuses on secure channels without the ROM, this comparison focuses on KEMs that are not proven secure in the ROM. However, most KEMs, which aim to be secure against quantum computers, use the Fujisaki-Okamoto transformation [23], which is proven in the ROM. As representative for KEMs in the ROM, Kyber [43] is used as reference as it has been standardized recently. To the best of our knowledge, the currently best KEMs without the ROM are the LPN-based proposed by Xu and Li [45] and the RLWE-based proposed by Yang, Ma, and Zhang [46].

Table 3.3 shows the parameters of the different KEMs. As can be seen, our SB-KEM is worse than Kyber. However, Kyber is highly optimized and our SB-KEM is not. On the other side, compared to the KEMs without the ROM our new SB-KEM performs way better. Although the LPN-based reaches smaller ciphertext sizes, its key sizes are significantly worse. Our SB-KEM is better than the RLWE-based in every category but for that the difference is lower. Therefore, the new SB-KEM improved upon KEMs that proven without the ROM. In particular, neither the keys nor the cipher has a size above one MB.

---

Scheme	SecLvl	pk	sk	c
Kyber512 [43]	128	0.8 KB	1.632 KB	0.768 KB
Kyber768 [43]	192	1.184 KB	2.4 KB	1.088 KB
LPN-based [45]	128	50.78 MB	62.50 MB	4.54 KB
RLWE-based [46]	80	1.923 MB	0.96 MB	1.280 MB
Ours	87	28.2 KB	51.5 KB	28.2 KB
Ours	164	56.3 KB	102 KB	56.3 KB

**Table 3.3:** Comparison of the key sizes for our construction with existing KEMs.

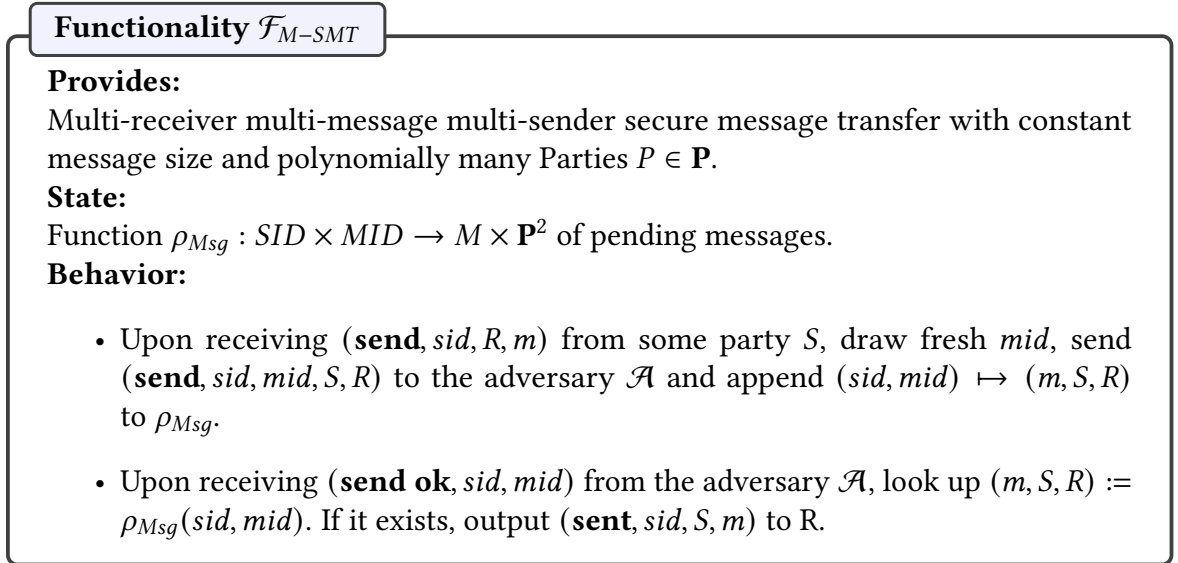


## 4 Secure Channels

This chapter describes secure channels and how the newly constructed SB-KEM is used to realize one. More precisely, the goal is to UC-realize  $\mathcal{F}_{M-SMT}$  [10] depicted in Fig. 4.1. This ideal functionality describes multi-receiver multi-message multi-sender secure message transfer of polynomially many parties. This means that multiple parties can send multiple messages to different receivers and the adversary can neither see the sent messages in plaintext nor change them. However, it can block any messages.

Benz et al. [7] showed that an SB-KEM can be combined with a DEM to realize  $\mathcal{F}_{M-SMT}$  using  $\mathcal{F}_{AUTH}$  by adjusting the protocol defined by Beskorovajnov et al. [10]. The latter uses an PKE, which the SB-KEM and DEM realize. Hereby, the SB-KEM needs to be IND-SB-CPA<sub>SB-KEM</sub> secure and the DEM IND-OT. As described in Section 2.10,  $\mathcal{F}_{AUTH}$  on the other hand can be realized by an EUF-CMA secure signature scheme combined with  $\mathcal{F}_{CA}$ . The protocol  $\pi_{M-SMT}^{\mathcal{F}_{CA}}$  combines the adjusted protocol with the protocols  $\pi_{CERT}^{\mathcal{F}_{CA}}$  and  $\pi_{AUTH}^{\mathcal{F}_{CERT}}$  to realize  $\mathcal{F}_{M-SMT}$  using only the ideal functionality  $\mathcal{F}_{CA}$ . The parameters and the states of each party are described in Fig. 4.2, while Fig. 4.3 depicts the behavior of each party. The protocol is secure under static corruption as the following lemma summarizes.

**Lemma 20.** *Under static corruption, the protocol  $\pi_{M-SMT}^{\mathcal{F}_{CA}}$  UC-realizes  $\mathcal{F}_{M-SMT}$  in the  $\mathcal{F}_{CA}$ -hybrid model.*



**Figure 4.1:** The Ideal Functionality  $\mathcal{F}_{M-SMT}$ .



**Protocol**  $\pi_{M-SMT}^{\mathcal{F}_{CA}}$ **Realizes:**

Multi-receiver multi-message multi-sender secure message transfer with constant message size.

**Parameters:**

- An EUF-CMA secure Signature Scheme  $\Sigma = (\Sigma.gen, \Sigma.enc, \Sigma.vfy)$
- An IND-SB-CPA<sub>SB-KEM</sub> Secure SB-KEM  $\Gamma = (\Gamma.gen, \Gamma.enc, \Gamma.dec)$
- An IND-OT DEM = (DEM.enc, DEM.dec)

**State of party P:**

- Function  $f_{CRED} : SID \rightarrow (\Gamma.PK, \Gamma.SK)$  of own credentials for encryption.
- Keypair  $(sk_{\Sigma}, vk) \in (\Sigma.VK, \Sigma.SK)$  of own credentials for signing.
- Function  $f_{PK} : SID \times P \rightarrow \Gamma.PK$  of known public keys.
- Function  $f_{VK} : SID \times P \rightarrow \Sigma.VK$  of known verification keys.
- Function  $f_{send} : SID \times P \rightarrow M^*$  of pending messages.

**Figure 4.2:** The Setup for the Protocol  $\pi_{M-SMT}^{\mathcal{F}_{CA}}$ , which Realizes  $\mathcal{F}_{M-SMT}$  using  $\mathcal{F}_{CA}$

*Proof.* The lemma follows from (Canetti [15] Claim 3, Claim 4), (Benz et al. [7] Theorem 1) and (Beskorovajnov et al. [10] Theorem 3).  $\square$

This and the next paragraph depict some intuition how the protocol works and protects against multiple attacks. First, the underlying idea is explained. To send a plaintext from A to B, A generates a symmetric key K with its corresponding cipher  $c_1$  with the SB-KEM, uses K to encrypt the plaintext into ciphertext  $c_2$  with the DEM and signs its message consisting of the two ciphers  $(c_1, c_2)$  with the signature scheme. Upon receiving, B checks the signature. If it is valid, it extracts K from  $c_1$  and uses it to decrypt  $c_2$ . If B has not generated public keys for the SB-KEM yet, A can send it a special message to ask it to do it. When generating public keys, they are broadcast to every other party. The certificate authority enables the parties to resolve a party ID into a verification key, so that signatures can verify the sending party.

Second, some intuition how the protocol protects against attacks is depicted, starting with resending a message in its own name. An attacker can send a message in its own name by replacing the sender ID and the signature. However, if the message contains a cipher, it will not decrypt correctly as the wrong sender ID is used in the SB-KEM. If it does not contain a cipher, its information is publicly available for every party. Therefore, the attacker could forge the whole message itself. To protect against forwarding attacks, the receiving party

**Protocol**  $\pi_{M-SMT}^{\mathcal{F}_{CA}}$

**Behavior of Party P:**

- Before signing for the first time: set  $(sk_{\Sigma}, vk) \leftarrow \Sigma.gen(1^{\kappa})$  and send **(Register, P, vk)** to  $\mathcal{F}_{CA}$
- Upon receiving  $(sid_{AUTH}, S, (\mathbf{init}, sid), \sigma)$  with fresh  $sid_{AUTH}$ , if there is no entry  $f_{Cred}(sid)$  yet:
  1. If  $f_{VK}(S)$  does not exist, send **(Retrieve, S)** to  $\mathcal{F}_{CA}$  to obtain **(Retrieve, S, vk)**. If  $vk = \perp$ , ignore the original message. Else set  $f_{VK}(S) = vk$ .
  2. Look up  $vk_S = f_{VK}(S)$ . If  $\Sigma.vfy(vk_S, (\mathbf{init}, sid, P, sid_{AUTH}), \sigma) = 0$ , ignore the original message.
  3.  $(sk, pk) \leftarrow \Gamma.gen(1^{\kappa})$
  4. Append  $sid \mapsto (sk, pk)$  to  $f_{Cred}$
  5. For each party  $P' \neq P$ : Draw a fresh  $sid'_{AUTH}$ , set  $m = (\mathbf{init}, sid, pk)$ ,  $\sigma \leftarrow \Sigma.sign(sk_{\Sigma}, (m, P', sid'_{AUTH}))$  and send  $(sid'_{AUTH}, P, m, \sigma)$  to  $P'$
- Upon receiving  $(sid_{AUTH}, P', (\mathbf{init}, sid, pk_{P'}), \sigma)$  with fresh  $sid_{AUTH}$ , if there is no entry  $f_{pk}(sid, P')$  yet:
  1. If  $f_{VK}(P')$  does not exist, send **(Retrieve, P')** to  $\mathcal{F}_{CA}$  to obtain **(Retrieve, P', vk)**. If  $vk = \perp$ , ignore the original message. Else set  $f_{VK}(P') = vk$ .
  2. Look up  $vk_{P'} = f_{VK}(P')$ . If  $\Sigma.vfy(vk_{P'}, (\mathbf{init}, sid, pk_{P'}, P, sid_{AUTH}), \sigma) = 0$ , ignore the original message.
  3. Append  $(sid, P') \mapsto pk_{P'}$  to  $f_{pk}$
  4. For any  $m \in f_{Send}(sid, P')$ 
    - a) Remove  $m$  from  $f_{Send}(sid, P')$
    - b)  $(K, c_0) \leftarrow \Gamma.enc(pk_{P'}, P)$
    - c)  $c_1 \leftarrow DEM.enc(K, m)$
    - d) Draw fresh  $sid_{AUTH}$ , set  $m' = (sid, (c_0, c_1))$
    - e) Draw  $\sigma \leftarrow \Sigma.sign(sk_{\Sigma}, (m', P', sid_{AUTH}))$  and send  $(sid_{AUTH}, P, m', \sigma)$  to  $P'$
- Upon receiving input **(send, sid, R, m)** with  $m \in \{0, 1\}^n$  from environment  $\mathcal{Z}$ :
  - If  $R = P$  report output **(sent, sid, P, m)** to the environment
  - Else if no entry  $f_{pk}(sid, R)$  exists yet:
    1. Append  $m$  to  $f_{Send}(sid, R)$
    2. Draw fresh  $sid_{AUTH}$
    3. For  $m' = (\mathbf{init}, sid)$  draw  $\sigma \leftarrow \Sigma.sign(sk_{\Sigma}, (m', R, sid_{AUTH}))$  and send  $(sid_{AUTH}, P, m', \sigma)$  to  $R$
  - Else:
    1.  $pk_R = f_{pk}(sid, R)$
    2.  $(K, c_0) \leftarrow \Gamma.enc(pk_R, P)$
    3.  $c_1 \leftarrow KEM.enc(K, m)$
    4. Draw fresh  $sid_{AUTH}$ . For  $m' = (sid, (c_0, c_1))$  draw  $\sigma \leftarrow \Sigma.sign(sk_{\Sigma}, (m', R, sid_{AUTH}))$  and send  $(sid_{AUTH}, P, m', \sigma)$  to  $R$
- Upon receiving  $(sid_{AUTH}, S, (sid, (c_0, c_1)), \sigma)$  with fresh  $sid_{AUTH}$ :
  1. If  $f_{VK}(S)$  does not exist, send **(Retrieve, S)** to  $\mathcal{F}_{CA}$  to obtain **(Retrieve, S, vk)**. If  $vk = \perp$ , ignore the original message. Else set  $f_{VK}(S) = vk$ .
  2. Look up  $vk_S = f_{VK}(S)$ . If  $\Sigma.vfy(vk_S, (sid, (c_0, c_1), P, sid_{AUTH}), \sigma) = 0$ , ignore the original message.
  3. Look up  $sk := f_{CRED}(sid)$  and  $pk := f_{pk}(sid, S)$ . If one of them does not exist, abort.
  4.  $K \leftarrow \Gamma.dec(sk, c_0, S)$
  5.  $m \leftarrow KEM.dec(K, c_1)$
  6. Report output **(sent, sid, S, m)** to the environment  $\mathcal{Z}$

**Figure 4.3:** The Behavior of Each Party in the Protocol  $\pi_{M-SMT}^{\mathcal{F}_{CA}}$ , which realizes  $\mathcal{F}_{M-SMT}$  using  $\mathcal{F}_{CA}$

is signed. Therefore, if an attacker takes a sent message and sends it to someone else, the new receiver will reject the message as the signature does not verify. Replay attacks are prevented by  $sid_{AUTH}$ , which is also signed. As a receiver only accepts a fresh  $sid_{AUTH}$ , a replayed message is ignored. When the attacker changes  $sid_{AUTH}$ , the signature is not valid anymore.

## 4.1 Concrete Instantiation

This section describes a concrete instantiation of the protocol  $\pi_{M-SMT}^{\mathcal{F}_{CA}}$ , which includes the remaining cryptographic schemes as well as party and message spaces. On the side of cryptographic schemes, an EUF-CMA secure signature scheme and an IND-OT DEM are missing. Possible options for the DEM are a one-time pad resulting in messages of  $n$  bits or a pseudo number generator leading to larger messages. For simplicity we opt to use the one-time pad leading to 512, 1024 or 2048 bit messages depending on the parameter set used. Note that the pseudo number generator would also enable variable message size, but the protocol only expects constant message size.

For the signature scheme, to the best of our knowledge, there are only few post-quantum signature schemes, which do not rely on the random oracle model. We choose SPHINCS256 [8], which is a stateless hash-based signature scheme with pk and sk sizes of about 1KB and signature size of 41KB. There is also SPHINCS+ [44], which is standardized and similar to SPHINCS256. However, it is not an option as its security is proven in the ROM. Another option is LMS [19], a stateful hash-based signature. The state requires more care in use and thus, we do not choose LMS.

The following describes the concrete instantiation of the protocol. Let  $(n, q, r, m)$  be the parameters chosen for  $\Gamma$ . The message space is  $M^* = \{0, 1\}^n$ , determined by the one-time pad. The party IDs have to fit into the input of the FRD of  $\Gamma$ , which is  $\mathbb{Z}_q^{n/r} \setminus \{0\}$ . This can encode  $\log q \cdot n/r$  bits. Therefore, the party ID space is  $\mathbf{P} \subseteq \{0, 1\}^{\log q \cdot n/r} \setminus \{0\}$ . In practice, this could be the hash of the email addresses of the parties using a collision resistance hash function. Note that 0 should not be a valid party ID, which can be enforced either by the hash function itself or for example by appending a 1 at the end of the hash.

## 4.2 Analysis

This section analyzes the performance of the protocol, specifically the message size and required storage. For concrete numbers, we use the second parameter set of  $n = 1024, \log q = 20, r = 16, m = 2$  as it provides more than 128 bits of security. Let  $s$  be the number of session identifiers,  $p$  be the number of parties. Each party has to store  $s$  key pairs of  $\Gamma$ , one key pair from SPHINCS256 and  $s \cdot p$  public keys from  $\Gamma$  and verification keys from SPHINCS256. As the plaintexts are only 128 bytes and only stored short term when waiting for the key

generation of the receiving party, we ignore them. This leads to the following formula of required storage.

$$s \cdot (32.6KB + 56.3KB) + 1.088KB + 1.056KB + s \cdot p \cdot (56.3KB + 1.056KB)$$

Depending on the number of parties, the most dominant part is either the key pairs of  $\Gamma$  for each session or the public key of  $\Gamma$  for each party. Next, the analysis of the message size follows. All messages contain  $sid_{AUTH}$ , the sender ID, the payload and a signature. The payload always contains the  $sid$ . The largest payload is the message containing the ciphers. Assuming 64 bit sids and 256 bit sender IDs, the size of the messages is up to

$$8B + 32B + 56.3KB + 32B + 41KB \approx 97.3KB.$$

Using the smallest parameter set instead, the message sizes are up to 69.3 KB, but the secure channel only provides 87 bits of security.



## 5 Implementation

A proof of concept for the SB-KEM described in Chapter 3 was implemented in python<sup>1</sup> version 3.12. This chapter describes implementation details and design choices.

The implementation is specifically for the parameter sets. That means it only supports a power of two cyclotomic as number field. The reason for this is that all calculations are performed in the coefficient embedding. More precisely, the used translation of Gaussian distributions to the coefficient embedding requires a power of two cyclotomic.

### 5.1 Decomposition of $f(X) = X^n + 1$

For the choice of  $q = 2r + 1 \pmod{4r}$  for some power of two  $r$ ,  $f \pmod{q}$  factors into  $r$  polynomials  $f_i$  as stated in Theorem 2. In addition, each  $f_i = X^{n/r} + a_i$  has a very simple structure and is determined by only one scalar  $a_i \in \mathbb{Z}_q^*$ . By the CRT, we have

$$\mathbb{Z}_q[X]/f(X) \cong \mathbb{Z}_q[X]/f_1(X) \times \dots \times \mathbb{Z}_q[X]/f_r(X)$$

and therefore, operations in  $R_q$  could be performed in the smaller fields  $\mathbb{Z}_q[X]/f_r(X)$  instead. However, for this the concrete values of  $a_i$  are needed. This section describes attempts at finding these. The proof of Theorem 2 is a constructive proof [32]. It states, that there are  $r$  values  $a_i \in \mathbb{Z}_q^*$  that have order  $2r$  in  $\mathbb{Z}_q$ , which are the values in the polynomials  $f_i$ . Moreover, if one value  $a_1$  is known, the others can be calculated as  $a_1^k$ , for  $1 < k < 2r$  with  $\gcd(k, 2r) = 1$ . As  $r$  is a power of two, the values for  $k$  are simply all odd numbers smaller than  $2r$ .

In the proof of concept, a method for calculating the decomposition of  $f$  was implemented by brute forcing one value  $a_1$  and then determining the others from this via  $a_1^k$  for odd  $k \in (1, 2r)$ . The brute force has random and ordered mode. In the ordered mode it iterates over all numbers in ascending order and tests whether their order is  $2r$ . The random mode chooses random values instead. The order of  $x$  is checked by the following two equations.

$$x^{2r} = 1 \wedge x^r \neq 1.$$

These two equations suffice as  $r$  is a power of two. If  $x^{2r} = 1$ , then the order has to divide  $2r$  and thus, can only be a power of two. On the other hand,  $x^r \neq 1$  implies that the order is not any smaller power of two than  $2r$ . For smaller values of  $q$  with  $\log q \leq 20$ , either mode

---

<sup>1</sup>[www.python.org](http://www.python.org)

works fine, however for larger values of  $q$  like  $\log q \geq 35$ , the brute force takes too much time independent of random or ordered. However, parallelism would speed up this process, as each value of  $x$  can be tested independent of each other.

## 5.2 Ring operations

The elements of the ring  $R_q = \mathbb{Z}_q[X]/(X^n + 1)$  are represented as polynomials of degree  $n - 1$ . For this, the library `numpy.polynomials`<sup>2</sup> is used as it provides polynomial operations, more specifically addition and multiplication. Division is implemented in the library. However, it is performed within float numbers, instead of  $\mathbb{Z}_q$  as required. This poses a problem as recovering the corresponding value in  $\mathbb{Z}_q$  from the float is unfeasible due to rounding errors. For the reduction modulo  $f = X^n + 1$ , the library can be used as the coefficients of  $f$  are either one or zero. However, for arbitrary polynomials, a custom method is used. The ring operations based on the CRT as in Section 5.1 was not implemented, instead all operations are performed in  $R_q$ .

The coefficients of the polynomials are stored as 64-bit integers. This increases the required storage compared to the calculations in Section 3.1, as these assumed only  $\log q$  bits per coefficient. This increases the required storage for the private key to about 524 KB instead of 51.2 KB when using the smallest parameter set. Note that no precise measurements of the required storage in the implementation were done due to the fact that it is unoptimized. This means that the value above is just for the raw data and in particular does not account for overhead due to the structures like vectors.

For larger  $q$ , the 64-bit integers are not sufficient and overflows occur during multiplication. To combat that, an option was implemented to store the coefficients as integer objects. This in turn increases the runtime as the operations on 64-bit integers are more optimized. The required storage with the integer objects in comparison to 64-bit integers was not measured but is expected to increase. Testing indicates that for  $\log q \geq 28$ , big integers should be used to ensure reliable results.

## 5.3 Gaussian Sampling

The construction requires sampling from a discrete Gaussian. This is approximated by sampling from a continuous Gaussian distribution and then rounding each index to the nearest integer. The  $n$ -dimensional continuous Gaussian of width 1 is sampled by sampling  $n$  one-dimensional Gaussians of width 1. To sample  $x$  with index  $S \in \mathbb{R}^{n \times n}$ ,  $y$  is sampled from  $\mathcal{D}_1^n$  and  $x = Sy$ . If  $S = sI_n$  is a multiple of the identity, each index is sampled from an one dimensional Gaussian of width  $s$  instead.

---

<sup>2</sup><https://numpy.org/doc/stable/reference/routines.polynomials-package.html>

A polynomial is sampled from the Gaussian distribution, by sampling the  $n$  coefficients as an  $n$ -dimensional discrete Gaussian as described above. For a uniform polynomial, each coefficient is sampled uniformly from  $\mathbb{Z}_q$ . For the invertible elements, rejection sampling is deployed. That means, a polynomial is sampled uniformly, checked whether it is invertible and resampled until it is invertible. The probability of an invertible element can be estimated using the decomposition of  $f = X^n + 1$  into  $r$  factors  $f_i$ . By Theorem 2 there are

$$f_i = X^{n/r} - a_i, \quad a_i \in \mathbb{Z}_q^*, \quad 1 \leq i \leq r,$$

for which we have

$$f(X) = \prod_{i=1}^r f_i \pmod{q}$$

and thus

$$\mathbb{Z}_q[X]/f(X) \cong \mathbb{Z}_q[X]/f_1(X) \times \dots \times \mathbb{Z}_q[X]/f_r(X).$$

As each  $f_i$  is a monic polynomial of degree  $n/r$ ,  $\mathbb{Z}_q[X]/f_i(X)$  contains  $q^{n/r}$  elements for each  $i$ . As an element in  $\mathbb{Z}_q[X]/f_1(X) \times \dots \times \mathbb{Z}_q[X]/f_r(X)$  is invertible iff all polynomials are not zero, there are  $(q^{n/r} - 1)^r$  invertible elements. For  $n = 512, r = 16, \log q = 17$ , the chance of a non-invertible polynomial is

$$1 - \frac{(q^{n/r} - 1)^r}{q^n} \approx 10^{-163}.$$

For larger values of  $n$  and  $q$ , this chance only decreases. Therefore, with very high likelihood, the rejection sampling has to perform only one sampling. Small testing of 10000 tests reinforce that thesis. Thus, for the power of two cyclotomics with these values of  $q$ , one might consider sampling a uniform polynomial and not checking whether it is invertible to reduce runtime.

When it comes to the width of the discrete Gaussians, if the width is a multiple of the identity, the Gaussian can be drawn in the coefficient embedding after scaling its width with factor  $1/\sqrt{n}$  as stated in Lemma 15. This is done for all discrete Gaussian distributions except the one for the error  $\bar{f}$ . The parameter choice accounts for that and thus the given parameters  $t\beta_{Tk}$  and  $tq\alpha_k$  are those used in the implementation as width of the Gaussian for  $T$  and the errors respectively. However, for the error  $\bar{f}$  the variance is calculated dynamically. Its equation in the canonical embedding is

$$\Sigma_f = t^2 \beta_T^2 \left( \sum_{i=1}^m \Sigma_{e_i} \right) + t^2 \beta_T^2 m (q\alpha)^2 I_n.$$

There is no formula to calculate  $\Sigma_{e_i}$  from the coefficient embedding of  $e_i$  without using the canonical embedding. Hence, to calculate  $\Sigma_f$ , the errors  $e_i$  are canonically embedded by calculating  $B_K \cdot e_i$  for each  $i$ . Then, the variance in the coefficient embedding is  $B_K^{-1} \Sigma_f B_K^{-H}$  as stated in Lemma 15. The resulting matrix contains only real entries as  $\Sigma_f$  has the same entry



for the complex conjugated pairs in  $B_K^{-1}$ , hence they cancel the imaginary part out. However, the implementation suffers from rounding errors, which result in relatively small imaginary parts unequal to zero. These imaginary parts are removed manually after confirming they are small rounding errors. As the different parameter sets induce very different orders of the values, checking the imaginary parts by comparing them to a constant value does not work. Instead, the imaginary parts are compared to the real parts and checked that they are multiple orders smaller. Note that this is the only occurrence of rounding errors as all other operations are performed on integers.

## 5.4 Inverting RLWE

As the used cyclotomic is a power of two cyclotomic, the inversion algorithm described by Lai, Cheung, and Chow [29] can be used to determine  $s \in R_q$  given  $b = gs + e$  with gadget vector  $g$  and small error  $e$ . It is based on Babai's nearest plane algorithm [4] and is as follows. Let  $S = (s_1, \dots, s_n)$  be a basis of  $\Lambda_q^\perp(g)$  and  $V := qS^{-T} = (v_1, \dots, v_n)$ . The goal of the algorithm is to recover  $s \in R_q$  from  $b = gs + e$ . For this, it considers all coefficients independent of each other. The  $i$ -th coefficient is retrieved with Babai's nearest plane algorithm, where the input  $b_{i,k} \in \mathbb{Z}_q^k$  is the vector with the  $i$ -th coefficient of each polynomial in  $b$ . That means, if the first index of  $b$  is  $c_1 + c_2X + \dots + c_nX^{n-1}$ , the first index of  $b_{i,k}$  is  $c_i$ . The  $i$ -th coefficient  $z_i$  of  $s$  is recovered from  $b_{i,k}$  as follows.

1. Compute the Gram-Schmidt orthogonalization  $\tilde{V} = (v_1^*, \dots, v_n^*)$  of  $V$ .
2. For  $j = k \rightarrow 1$ :
  - a) Compute  $l_{i,j} = \frac{\langle b_{i,j}, v_j^* \rangle}{\langle v_j^*, v_j^* \rangle}$
  - b) Set  $b_{i,j-1} = b_{i,j} - (l_{i,j} - \lfloor l_{i,j} \rfloor)v_j^* - \lfloor l_{i,j} \rfloor v_j$
3. Return  $z_i = \sum_{j=1}^k \lfloor l_{i,j} \rfloor c_j \pmod q$ , where  $v_j = c_j g \pmod q$

Hereby,  $\lfloor x \rfloor$  denotes rounding  $x$  to the nearest integer. Note that the Gram-Schmidt orthogonalization is only calculated once. For the basis  $S$ ,  $B_g$  as in Eq. (2.3) is used. As each polynomial in  $B_g$  is a constant polynomial, so are the ones in  $V$ . In addition,  $V$  is a basis of  $\{ag \pmod q \mid a \in R_q\}$  [29]. Therefore, there is an  $c_j \in \mathbb{Z}_q$  such that  $v_j = c_j g \pmod q$ . For the implementation,  $c_j$  was determined by taking the first index of  $v_j$  modulo  $q$  as  $g_1 = 1$ . Another important detail is that the vectors in the Gram-Schmidt orthogonalization are not normalized.

For any  $a \in R_q^{k+m}$ ,  $c = as + e$  is inverted as described in Section 2.8.1. This means, that  $b = (T, I)c$ ,  $\hat{s}$  with  $b = g\hat{s} + \hat{e}$  is determined with the algorithm above and  $s = h^{-1}\hat{s}$ . As  $c$  is the cipher, it is split into  $c_1$  and  $c_2$ . Therefore, the multiplication of  $(T, I)$  is performed by multiplying  $c_1$  with  $T$  and adding  $c_2$ .

## 6 Conclusion

This work improves the current state of secure channels, which are proven secure without the ROM by adapting the SB-KEM from Benz et al. [7] to the ring setting. In addition to proving the correctness and security based on RLWE, concrete parameters for the new construction are proposed. With these, the key sizes are in the orders of kilobytes, which is a vast improvement over existing KEMs that do not use the ROM. However, it is still significantly larger than Kyber, a standardized KEM proven secure in the ROM. The SB-KEM is combined with the SPHINCS256 signature scheme and one-time pads to realize  $\mathcal{F}_{SMT}$  in the UC framework using only  $\mathcal{F}_{CA}$  as an ideal functionality. Thereby, a secure channel with message sizes of about 69.2 KB for 87 bit and 97.3 KB for 128 bit security is proven. Lastly, a proof of concept of the SB-KEM implemented in python confirms the theoretical results.

As the security proof of this SB-KEM is similar to the IND-CCA proof for the hybrid scheme from Boyen, Izabachène, and Li [13], it covers many steps involved for proving IND-CCA security for the SB-KEM. The missing detail is the binding property of Ring LWE samples, which states that for a given  $a, b$ , there is only one  $s$  such that it is  $b = as + e$  for small  $e$ . On the side of implementation, the proof of concept is not optimized. Improvements in terms of runtime and required storage can be scope of future work. For one, the decomposition described in Section 5.1 would speed up the ring operations. The generation of the decomposition but also other calculations like the inversion algorithm could profit from more parallelism. With these speed ups, the runtime of the SB-KEM might be competitive to those of Kyber.



# Acronyms

**CA** Certificate Authority.

**CRT** Chinese Remainder Theorem.

**d-NRLWE** Decision Normal Ring Learning with Errors.

**DEM** Data Encapsulation Mechanism.

**EUF-CMA** Existential Unforgability under Chosen Message Attack.

**FRD** Full-Rank Difference Encoding.

**ID** Identity.

**IND-aTAG-wCCA** Indistinguishability under Selective-Tag Weakly Chosen Ciphertext Attack.

**IND-aTAG-wCCA** Indistinguishability under Adaptive-Tag Weakly Chosen Ciphertext Attack.

**IND-OT** Indistinguishability under One-Time Attack.

**IND-SB-CPA** Indistinguishability under Sender-Binding Chosen Plaintext Attack.

**IND-SB-CPA<sub>SB-KEM</sub>** Indistinguishability under Sender-Binding Chosen Plaintext Attack.

**IND-CCA** Indistinguishability under Chosen Ciphertext Attack.

**KB** Kilobytes.

**KDF** Key Derivation Function.

**KEM** Key Encapsulation Mechanism.

**LWE** Learning with Errors.

**MB** Megabytes.

**OW-CPA** One-Way under Chosen-Plaintext Attacks.

**PKE** Public Key Encryption Scheme.

**PPT** Probabilistic Polynomial Time.

**RCCA** Replayable Chosen Ciphertext Attack.

**RLWE** Ring Learning with Errors.

**ROM** Random Oracle Model.

**sid** Session Identifier.

**s-NRLWE** Search Normal Ring Learning with Errors.

**SB-KEM** Sender-Binding Key Encapsulation Mechanism.

**SMT** Secure Message Transfer.

**ssid** Subsession identifier.

**TBE** Tag Based Encryption.

**TLS** Transport Layer Security.

**UC** Universal Composability.

# List of Figures

2.1	The IND-CCA Game for KEMs . . . . .	20
2.2	The IND-OT Game for DEMs . . . . .	21
2.3	The IND-SB-CPA <sub>SB-KEM</sub> Game . . . . .	22
2.4	The EUF-CMA Game . . . . .	23
2.5	The Ideal $\mathcal{F}_{AUTH}$ Functionality . . . . .	28
2.6	The Ideal $\mathcal{F}_{CERT}$ Functionality [15] . . . . .	29
2.7	The Protocol $\pi_{AUTH}^{\mathcal{F}_{CERT}}$ Realizing $\mathcal{F}_{AUTH}$ . . . . .	29
2.8	The Ideal Functionality $\mathcal{F}_{CA}$ . . . . .	30
2.9	The Protocol $\pi_{CERT}^{\mathcal{F}_{CA}}$ Realizing $\mathcal{F}_{CERT}$ using an EUF-CMA Secure Signature Scheme . . . . .	31
4.1	The Ideal Functionality $\mathcal{F}_{M-SMT}$ . . . . .	47
4.2	The Setup for the Protocol $\pi_{M-SMT}^{\mathcal{F}_{CA}}$ , which Realizes $\mathcal{F}_{M-SMT}$ using $\mathcal{F}_{CA}$ . . . . .	48
4.3	The Behavior of Each Party in the Protocol $\pi_{M-SMT}^{\mathcal{F}_{CA}}$ , which realizes $\mathcal{F}_{M-SMT}$ using $\mathcal{F}_{CA}$ . . . . .	49



# Bibliography

- [1] Masayuki Abe et al. “Tag-KEM/DEM: A new Framework for Hybrid Encryption and a new Analysis of Kurosawa-Desmedt KEM”. In: *Advances in Cryptology – EUROCRYPT 2005*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 128–146. ISBN: 978-3-540-32055-5.
- [2] Martin R Albrecht, Rachel Player, and Sam Scott. “On the Concrete Hardness of Learning with Errors”. In: *Journal of Mathematical Cryptology* 9.3 (2015), pp. 169–203. DOI: 10.1515/jmc-2015-0016.
- [3] Nicolas Aragon et al. *BIKE: Bit Flipping Key Encapsulation*. hal-04278509. 2022. URL: <https://inria.hal.science/hal-04278509v1>.
- [4] László Babai. “On Lovász’lattice Reduction and the Nearest Lattice Point Problem”. In: *Combinatorica* 6 (1986), pp. 1–13. DOI: 10.1007/BF02579403.
- [5] Wojciech Banaszczyk. “New Bounds in some Transference Theorems in the Geometry of Numbers”. In: *Mathematische Annalen* 296 (1993), pp. 625–635.
- [6] Fabrice Benhamouda et al. “Hash Proof Systems over Lattices Revisited”. In: *Public-Key Cryptography – PKC 2018*. Cham: Springer International Publishing, 2018, pp. 644–674. ISBN: 978-3-319-76581-5.
- [7] Laurin Benz et al. “Sender-binding Key Encapsulation”. In: *Public-Key Cryptography – PKC 2023*. Cham: Springer Nature Switzerland, 2023, pp. 744–773. ISBN: 978-3-031-31368-4.
- [8] Daniel J Bernstein et al. “SPHINCS: Practical Stateless hash-Based Signatures”. In: *Advances in Cryptology – EUROCRYPT 2015*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 368–397. ISBN: 978-3-662-46800-5.
- [9] Pauline Bert et al. “Implementation of Lattice Trapdoors on Modules and Applications”. In: *Post-Quantum Cryptography*. Cham: Springer International Publishing, 2021, pp. 195–214. ISBN: 978-3-030-81293-5.
- [10] Wasilij Beskorovajnov et al. “A New Security Notion for PKC in the Standard Model: Weaker, Simpler, and Still Realizing Secure Channels”. In: *Public-Key Cryptography – PKC 2022*. Cham: Springer International Publishing, 2022, pp. 316–344. ISBN: 978-3-030-97131-1.
- [11] Olivier Blazy, Céline Chevalier, and Quoc Huy Vu. “Post-Quantum UC-Secure Oblivious Transfer in the Standard Model with Adaptive Corruptions”. In: *Proceedings of the 14th International Conference on Availability, Reliability and Security*. 2019, pp. 1–6. DOI: 10.1145/3339252.3339280.



- [12] Jean-Philippe Bossuat et al. *Security Guidelines for Implementing Homomorphic Encryption*. Cryptology ePrint Archive, Paper 2024/463. Accessed 11-7-2024. 2024. URL: <https://eprint.iacr.org/2024/463>.
- [13] Xavier Boyen, Malika Izabachène, and Qinyi Li. “Secure Hybrid Encryption in the Standard Model from Hard Learning Problems”. In: *Post-Quantum Cryptography*. Cham: Springer International Publishing, 2021, pp. 399–418. ISBN: 978-3-030-81293-5.
- [14] Ran Canetti. “Universally Composable Security: A new Paradigm for Cryptographic Protocols”. In: *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. IEEE. 2001, pp. 136–145. DOI: 10.1109/SFCS.2001.959888.
- [15] Ran Canetti. “Universally Composable Signature, Certification, and Authentication”. In: *Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004*. IEEE. 2004, pp. 219–233. DOI: 10.1109/CSFW.2004.1310743.
- [16] Ran Canetti, Hugo Krawczyk, and Jesper B Nielsen. “Relaxing Chosen-Ciphertext Security”. In: *Advances in Cryptology - CRYPTO 2003*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 565–582. ISBN: 978-3-540-45146-4.
- [17] Ran Canetti and Tal Rabin. “Universal Composition with Joint State”. In: *Advances in Cryptology - CRYPTO 2003*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 265–281. ISBN: 978-3-540-45146-4.
- [18] Allan Clark. *Elements of Abstract Algebra*. Dover Books on Mathematics Series. Courier Corporation, 1984. ISBN: 9780486647258.
- [19] David A Cooper et al. “Recommendation for Stateful Hash-Based Signature Schemes”. In: *NIST Special Publication 800.208 (2020)*, pp. 800–208. DOI: 10.6028/NIST.SP.800-208.
- [20] Ronald Cramer and Victor Shoup. “Design and Analysis of Practical Public-Key Encryption Schemes secure against Adaptive Chosen Ciphertext Attack”. In: *SIAM Journal on Computing* 33.1 (2003), pp. 167–226. DOI: 10.1137/S0097539702403773.
- [21] Eric Crockett and Chris Peikert. *Challenges for Ring-LWE*. Cryptology ePrint Archive, Paper 2016/782. Accessed 11-7-2024. 2016. URL: <https://eprint.iacr.org/2016/782>.
- [22] Nicholas Genise et al. “Improved Discrete Gaussian and Subgaussian Analysis for Lattice Cryptography”. In: *Public-Key Cryptography – PKC 2020*. Cham: Springer International Publishing, 2020, pp. 623–651. ISBN: 978-3-030-45374-9.
- [23] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. “A Modular Analysis of the Fujisaki-Okamoto Transformation”. In: *Theory of Cryptography*. Cham: Springer International Publishing, 2017, pp. 341–371. ISBN: 978-3-319-70500-2.
- [24] Andreas Hülsing et al. “High-Speed Key Encapsulation from NTRU”. In: *Cryptographic Hardware and Embedded Systems – CHES 2017*. Cham: Springer International Publishing, 2017, pp. 232–252. ISBN: 978-3-319-66787-4.
- [25] Kiyosi Itô and Nihon Sūgakkai. *Encyclopedic Dictionary of Mathematics*. Encyclopedic Dictionary of Mathematics Bd. 1. MIT Press, 1993. ISBN: 9780262590204. URL: <https://books.google.de/books?id=WHj09K6xE4C>.

- 
- [26] Dan Kalman. “The Generalized Vandermonde Matrix”. In: *Mathematics Magazine* 57.1 (1984), pp. 15–21. DOI: 10.1080/0025570X.1984.11977069.
- [27] Eike Kiltz. “Chosen-Ciphertext Security from Tag-Based Encryption”. In: *Theory of Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 581–600. ISBN: 978-3-540-32732-5.
- [28] Neal Koblitz and Alfred J Menezes. “The Random Oracle Model: a Twenty-Year Retrospective”. In: *Designs, Codes and Cryptography* 77 (2015), pp. 587–610. DOI: 10.1007/s10623-015-0094-2.
- [29] Russell WF Lai, Henry KF Cheung, and Sherman SM Chow. “Trapdoors for Ideal Lattices with Applications”. In: *Information Security and Cryptology*. Cham: Springer International Publishing, 2014, pp. 239–256. ISBN: 978-3-319-16745-9.
- [30] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. “A Toolkit for Ring-LWE Cryptography”. In: *Advances in Cryptology – EUROCRYPT 2013*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 35–54. ISBN: 978-3-642-38348-9.
- [31] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. “On Ideal Lattices and Learning with Errors Over Rings”. In: *Advances in Cryptology – EUROCRYPT 2010*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–23. ISBN: 978-3-642-13190-5.
- [32] Vadim Lyubashevsky and Gregor Seiler. “Short, Invertible Elements in Partially Splitting Cyclotomic Rings and Applications to Lattice-Based zero-knowledge Proofs”. In: *Advances in Cryptology – EUROCRYPT 2018*. Cham: Springer International Publishing, 2018, pp. 204–224. ISBN: 978-3-319-78381-9.
- [33] Philip MacKenzie, Michael K Reiter, and Ke Yang. “Alternatives to Non-Malleability: Definitions, Constructions, and Applications”. In: *Theory of Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 171–190. ISBN: 978-3-540-24638-1.
- [34] MATZOV. *Report on the Security of LWE: Improved Dual Lattice Attack*. <https://zenodo.org/records/6412487>. Accessed 08-24-2024. 2022.
- [35] Daniele Micciancio and Chris Peikert. “Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller”. In: *Advances in Cryptology – EUROCRYPT 2012*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 700–718. ISBN: 978-3-642-29011-4.
- [36] Daniele Micciancio and Oded Regev. “Worst-Case to Average-Case Reductions Based on Gaussian Measures”. In: *SIAM Journal on Computing* 37.1 (2007), pp. 267–302. DOI: 10.1137/S0097539705447360.
- [37] Frédérique Oggier and Emanuele Viterbo. *Algebraic Number Theory and Code Design for Rayleigh Fading Channels*. Foundations and trends in communications and information theory. Now, 2004. ISBN: 9781933019079.
- [38] Dingyi Pei, Arto Salomaa, and Cunsheng Ding. *Chinese Remainder Theorem: Applications In Computing, Coding, Cryptography*. World Scientific Publishing Company, 1996. ISBN: 9789814498364. URL: <https://books.google.de/books?id=RQLtCgAAQBAJ>.
- [39] Chris Peikert. “How (Not) to Instantiate Ring-LWE”. In: *Security and Cryptography for Networks*. Cham: Springer International Publishing, 2016, pp. 411–430. ISBN: 978-3-319-44618-9.

- [40] Oded Regev. “On Lattices, Learning with Errors, Random Linear Codes, and Cryptography”. In: *Journal of the ACM (JACM)* 56.6 (2009), pp. 1–40. DOI: 10.1145/1568318.1568324.
- [41] Ronald L Rivest, Adi Shamir, and Leonard Adleman. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. In: *Communications of the ACM* 21.2 (1978), pp. 120–126. DOI: 10.1145/359340.359342.
- [42] Peter W Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM review* 41.2 (1999), pp. 303–332. DOI: 10.1137/S0036144598347011.
- [43] National Institute of Standards and Technology. “Module-Lattice-Based Key-Encapsulation Mechanism Standard”. In: *Federal Information Processing Standards Publication* (2024). DOI: 10.6028/NIST.FIPS.203.
- [44] National Institute of Standards and Technology. “Stateless Hash-Based Digital Signature Standard”. In: *Federal Information Processing Standards Publication* (2024). DOI: 10.6028/NIST.FIPS.205.
- [45] Shengfeng Xu and Xiangxue Li. “Chosen-Ciphertext Secure Key Encapsulation Mechanism in the Standard Model”. In: *IEEE Access* 9 (2021), pp. 13683–13690. DOI: 10.1109/ACCESS.2021.3051047.
- [46] Xiaopeng Yang, Wenping Ma, and Chengli Zhang. “Efficient Chosen Ciphertext Secure Key Encapsulation Mechanism in Standard Model over Ideal Lattices”. In: *International Journal of Computer Mathematics* 94.5 (2017), pp. 866–883. DOI: 10.1080/00207160.2016.1149578.
- [47] Yu Yu and Jiang Zhang. “Cryptography with Auxiliary Input and Trapdoor from Constant-Noise LPN”. In: *Advances in Cryptology – CRYPTO 2016*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 214–243. ISBN: 978-3-662-53018-4.