Efficient Fully Homomorphic Encryption and Digital Signatures Secure from Standard Assumptions

Ryo Hiromasa

Abstract

The two main goals of cryptography are to achieve an important cryptosystem in an efficient way, and provide a theoretical proof for ensuring its security.

In this thesis, we investigate ways to achieve an advanced encryption method (which is now considered to be the most important cryptosystem and whose typical example is *fully homomorphic encryption* (FHE)), and authentication methods (e.g., *digital signatures* and *blind message signatures*). We propose cryptosystems that have *practical efficiency* and *proven security under standard assumptions*. These proposed cryptosystems are more or equally efficient than the existing cryptosystems with the same security guarantee and are proven secure under the standard assumptions (the security of a cryptosystem is guaranteed by a certain computational hardness assumption, and the most desirable assumptions are the standard assumptions).

FHE allows us to evaluate any function over encrypted data by only using public information. This can be used, for example, for outsourcing computations over encrypted data to a remote server. We present the first FHE scheme that encrypts matrices into one ciphertext and supports homomorphic matrix operations, which lead to homomorphic single-instruction-multiple-data (SIMD) operation. Our FHE scheme has more efficient homomorphic operation algorithms than all known FHE schemes based on the standard assumptions.

Digital signatures are a way to ensure that data are actually from the sender of the data, or the data have not been tampered. Our proposed signatures are proven secure under the RSA assumption in the random oracle model (which is an idealized model of hash functions). The signature scheme has a simpler construction than previous schemes with the same security guarantee, since the number of (idealized) hash functions used in our scheme is optimal. The simpler construction of the proposed scheme enables lower implementation costs for the secure digital signatures.

Blind message signatures are a variant of digital signatures, and they enable us to sign data that are hidden from a signer. The proposed scheme is the first efficient blind message signatures based on the RSA assumption in the random oracle model. The RSA assumption is the most widely used and so thought as the most reliable cryptographic assumption.

Contents

Chapter 1

Introduction

1.1 Modern Cryptography

Cryptography is related to secure communications in the presence of an adversary. Behaviors of the adversary are roughly classified into three types of attacks: *wiretapping*, *tampering*, and *spoofing*. The first attack, wiretapping, is to monitor what data flow through communication lines, and to obtain secret information from the data. The second attack, tampering, is to change the contents of the data flowing through that line. The third attack, spoofing, is to impersonate the sender of the data in order to manipulate the receiver to take some actions. Main goals of cryptography are to create a method for preventing such attacks: cryptosystems to *hide* secret transmitted data and *authenticate* received data.

The first important objective of cryptography is to construct a cryptosystem for hiding secret information from an adversary. Let us suppose that Alice wants to send her secret message to Bob over an insecure communication channel. Then an adversary, Eve, may wiretap the message to know what Alice told Bob. Public key encryption (PKE) is a way to deal with this problem. The concept of PKE was first proposed by Diffie and Hellman [DH76] in 1976, and it enabled secure communications between Alice and Bob. A particular PKE consists of a triple of public algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. Suppose that Alice has message m and Bob has two keys $(\mathsf{pk}, \mathsf{sk})$, public and secret keys, generated by the key generation algorithm KeyGen. The two keys are used to encrypt messages and decrypt ciphertexts, respectively. Alice first uses Bob's public key to encrypt m by computing a ciphertext $c = \mathsf{Enc}(\mathsf{pk}, m)$, and sends c to Bob. Bob uses sk to decrypt the ciphertext and obtain the message from Alice, $m = \mathsf{Dec}(\mathsf{sk}, c)$. Since ciphertexts encrypted under a public key cannot be decrypted without the corresponding secret key, no one can know the message from Alice unless Bob's secret key is exposed.

An advanced encryption method is a cryptosystem that provides a certain functionality in addition to PKE. A typical example of this method is fully homomorphic encryption (FHE), which is a variant of PKE and allows us to evaluate any function over encrypted data by only using public information. A FHE scheme has, in addition to the PKE algorithms, efficient algorithm Eval that for any valid public key pk, any circuit C [1] , and any ciphertext $c_i \leftarrow \mathsf{Enc}(\mathsf{pk}, m_i)$, outputs $c \leftarrow \mathsf{Eval}(\mathsf{pk}, c_1, \ldots, c_r)$ such that the decryption holds $C(m_1, \ldots, m_r) \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$ for the secret key sk corresponding to pk. This has various applications in practical and theoretical studies. One typical practical application of FHE is to outsource computations to remote servers without compromising privacy. Suppose that Alice has message m, circuit C, and public and secret keys of a FHE scheme. Alice first encrypts m under her public key and sends it to a server. The server receives the ciphertext of m, evaluates C on the ciphertext of m to return a ciphertext of $C(m)$. Alice uses her secret key to decrypt the ciphertext from the server and obtain $C(m)$. In theoretical studies, FHE can be used to construct many powerful cryptographic tools, such as the candidates of cryptographic multilinear maps [GGH13a, CLT13, GGH15] and a program obfuscator [GGH+13b].

The second important objective of cryptography is to construct a way for authenticating data. We want to ensure that the data are actually from the sender of the data, or the data have not been tampered. Suppose that Bob receives a message addressed from Alice. Then how does Bob ensure that the received message is the same as the message actually sent from Alice? In particular, how does he ensure that the received message has not been tampered with by an adversary? Digital signatures are a way to accomplish this. A particular digital signature scheme consists of three algorithms $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$. The key generation algorithm KeyGen generates a pair of keys $(\mathsf{pk}, \mathsf{sk})$ as well as the PKE. Anyone who knows the public key of Alice can verify that a digital signature is generated by Alice, and such a signature is generated only by someone who knows the corresponding secret key. Alice generates a signature σ for a message m by computing $\sigma \xleftarrow{R} \mathsf{Sign}(\mathsf{sk}, m)$ with her secret key sk, and sends to Bob σ with m (practically, the message is encrypted by public or private key encryption algorithm). By using Alice's public key pk to verify that the received signature is a valid signature for m, Bob ensures that m is not tampered with by an adversary and the data is certainly sent from Alice.

The importance of the above cryptosystems will grow in the future as more everyday tasks, processes, and communications are computationalized. The most essential things in cryptography are to contrive *efficient* ways for achieving such important cryptosystems, and provide a *theoretical proof* for ensuring their security.

[1] Precisely, C will always be either a Boolean or an arithmetic circuit.

Constructing more efficient cryptosystems that implement important cryptographic functions allows us to apply their functions to wider areas. An efficient cryptosystem has smaller keys, ciphertexts, or signatures, or has smaller complexity of algorithms (e.g., key generation, encryption, decryption, signature generation, or verification). A cryptosystem with smaller keys or smaller complexity of the algorithms can be used in resource constrained devices. The smaller ciphertexts, or signatures enable secure communications on capacity-restrained channels.

Provable security is a central notion in modern cryptography. A cryptosystem is said to be provably secure if breaking security of the cryptosystem leads breaking a certain assumption (which says that a certain problem is hard to solve) To prove formally the security of a cryptosystem, we construct a reduction (algorithm) that uses an adversary against the cryptosystem to break the assumption, i.e., solve the problem assumed to be intractable in the assumption.

In cryptography, the security assumptions are classified into the following five types of assumptions:

- **Standard assumptions.** Standard assumptions are the security assumptions that are widely known and used in cryptography. For example, in the RSA assumption [RSA78], an adversary is given an RSA modulus N (that is a composite number of two primes), an integer $e \in \mathbb{Z}$ such that $\gcd(e, \phi(N)) = 1$, and a target $y \in \mathbb{Z}_N$. The task of the adversary is to compute an integer x such that $x^e \equiv y \pmod{N}$. Other examples of the standard assumptions are the factoring assumption, the discrete log assumption, and the learning with errors (LWE) assumption [Reg05].

- **Non-standard, falsifiable, non-interactive, and static-size assumptions.** A falsifiable assumption can be modeled as an interactive game between an challenger and adversary. At the end of the game, the challenger can efficiently decide whether the adversary won that game. The assumption states that every efficient adversary win the game with negligible probability. If the falsifiable assumption is false, we can construct an efficient process to show that the assumption is false. The standard assumptions are also included in the falsifiable assumptions. In the non-interactive assumption, the adversary is not given access to any oracles, and the static-size assumptions do not depend on any system parameter and only depend on the security parameter. Examples of this type of assumption are the strong-RSA assumption [CS00], the (asymmetric or symmetric) external Diffie-Hellman assumption, the Φ-hiding assumption [CMS99], the learning parity with noise (LPN) assumption [BFKL94], and so on.

- **Falsifiable and non-interactive, but dynamic-size (q-type) assumptions.**

These assumptions are the so called q-type assumptions, in which the size of the assumption grows dynamically. For example, in the q-weak Diffie-Hellman assumption [MSK02], the adversary is given $(g, g^x, g^{x^2}, \ldots, g^{x^q})$ and asked to compute $g^{1/x}$. Other examples of this type of assumption are the q-strong Diffie-Hellman assumption [BB04b], the q-bilinear Diffie-Hellman inversion assumption [BB04a], and the q-bilinear Diffie-Hellman exponent assumption [BBG05].

- **Falsifiable and interactive assumptions.** In the interactive assumptions, the adversary is given access to some oracles. For example, in the one-more RSA inversion assumption [BNPS03], the adversary is given $\ell + 1$-targets and access to an RSA-inversion oracle $(\cdot)^d \bmod N$ that it takes input $y \in \mathbb{Z}_N^*$ and returns its RSA-inverse $y^d \bmod N$, The task of the adversary is to compute the RSA-inverses of all the given targets while submitting at most ℓ queries to the oracle. Another example of this type of assumption is the LRSW assumption [LRSW99].

- **Unfalsifiable assumptions.** This type of assumption is not included in the falsifiable assumptions. A typical example of the unfalsifiable assumption is the knowledge exponent assumption (KEA1) [Dam91], in which for any adversary that takes as input $q, g, g_1 = g^x$ and returns g_2, g_3 with $g_3 = g_2^x$, there exists an extractor that takes the same inputs as the adversary and outputs y such that $g^y = g_2$. In the falsifiable assumptions, the adversary is asked to produce a certain output on certain inputs. To show that such an assumption is false, we simply give an attack in the form of an adversary whose success probability is not negligible. However, the KEA1 assumption has a more complex format: it states "for any adversary there exists an extractor such that ...". To show that this assumption is false, we must prove that there is an adversary for which there exists no extractor, so it is difficult to construct an efficient process to show that the assumption is false.

From the above, we can see that there is a vast number of security assumptions in cryptography. Among these assumptions, the most desirable assumptions are the standard assumptions.

To make cryptography an integral part of the information security supporting our networked society, cryptography must provide important cryptosystems, be implemented efficiently, and its security be guaranteed theoretically. For achieving cryptosystems that will become part of such a foundation, this thesis proposes efficient and theoretically secure schemes (under the standard assumptions) to implement FHE and signatures that are now considered as the most important cryptosystems.

1.2 Results of This Thesis

As described in the last section, the most essential goals in cryptography are to construct an efficient scheme for implementing an important cryptosystem, and to prove theoretically its security. The security of a modern cryptosystem is proved under certain computational assumptions and the most desirable assumption is the standard assumption. Hence, constructing an important cryptosystem that has both better efficiency and a theoretical security guarantee under the standard assumption is the ultimate goal of cryptography.

In this thesis, we propose three efficient schemes to implement an advanced encryption method or a data authentication method. The proposed schemes are proven secure based on the standard assumptions, and they are the most efficient ones under the same security assumptions. In particular, we construct the following cryptographic schemes and protocols:

- An efficient FHE scheme based on the LWE assumption [HAO16a] [2].
- A tightly secure digital signatures based on the RSA assumption [HAO16b].
- An efficient blind message signatures based on the RSA assumption.

In the next three sections, we briefly provide the detail of our constructions.

1.2.1 Efficient FHE based on the LWE Assumption

In 1978, Rivest, Shamir, and Adleman [RSA78] first constructed a PKE scheme, called RSA. The encryption function of the basic RSA scheme is (multiplicative) homomorphic [3]: given a public key $\mathsf{pk} := (n, e) \in \mathbb{Z} \times \mathbb{Z}$ and ciphertexts $c_i := m_i^e \bmod n$ of messages m_i (for $i = 1, 2$), we can efficiently compute the ciphertext of the product $m_1 \cdot m_2$ by $c_1 \cdot c_2 \equiv (m_1 \cdot m_2)^e \pmod{n}$. That is, in the basic RSA scheme, we can compute multiplication of data without decrypting the ciphertexts of them. Shortly after the invention of the RSA scheme, Rivest, Adleman, and Dertouzos [RAD78] raised an interesting question:

> *Can we compute arbitrary operations on encrypted data without decrypting them?*

The way to answer this question had been regarded to be a "holy grail" of cryptography over the years. In 2009, Gentry first presented an answer to it, *Fully Homomorphic Encryption* (FHE).

[2] The preliminary version of this paper is [HAO15].

[3] Let $\mathbb{G}$ and $\mathbb{G}'$ be groups with operations $\circ$ and $\cdot$, respectively. We say that a function $f : \mathbb{G} \to \mathbb{G}'$ is homomorphic if it holds that $f(x \circ y) = f(x) \cdot f(y)$ for any $x, y \in \mathbb{G}$.

FHE allows us to evaluate any function over encrypted data by only using public information. A natural example of its application is searching on encrypted data. Suppose that Alice stores her files on a remote server so that she can access to the files without having her own computer, and that she encrypts her files to prevent the server from reading or leaking her private data. Let $m_1, \ldots, m_r$ be the files, and let them be encrypted to the ciphertexts $c_1, \ldots, c_r$. When Alice later wants to download the encrypted files satisfying a query, she sends to the server her query, which is expressed as a circuit C. The server homomorphically evaluates C on the encrypted files $c_1, \ldots, c_r$, and returns the ciphertext $\hat{c}$ to Alice. She decrypts $\hat{c}$ to obtain $C(m_1, \ldots, m_r)$, which satisfies her query.

Since the breakthrough work by Gentry [Gen09a, Gen09b], many different varieties of FHE have been proposed [DGHV10, BV11a, BV11b, BGV12, Bra12, GSW13, CLT14]. To date, the fastest (and simplest) FHE based on the *standard* assumption is the one proposed by Gentry, Sahai, and Waters [GSW13] (hereafter, referred to as GSW-FHE). However, it is required to take heavy cost for evaluating a large number of ciphertexts. A way to deal with this issue is to *pack* multiple messages into one ciphertext.

Packing messages allows us to apply *single-instruction-multiple data* (SIMD) homomorphic operations to all encrypted messages. In the case where a remote server stores encrypted data and we want to retrieve certain data from this server, we first apply the equality function to every encrypted data. If the stored data have been packed into one ciphertext, we can retrieve the desired data by only one homomorphic evaluation of the equality function.

In this thesis, we construct a variant of the GSW-FHE that supports homomorphic matrix operations. Homomorphic matrix operations immediately lead to implementing SIMD homomorphic operations, so we can obtain SIMD FHE with very simple (actually, just matrix addition and multiplication) homomorphic operation algorithms like GSW-FHE. Our construction is based on the LWE assumption, and is more efficient than the previous SIMD FHE based on the same assumption. Our construction is an extension of the GSW-FHE scheme [GSW13], which greatly influenced on the construction ideas of some cryptosystems based on LWE such as fully homomorphic signatures [GVW15, FMNP16], attribute based encryption [BGG+14, BV16, BCTW16], and multilinear maps [GGH15]. Hence, the idea for constructing our FHE may also be a foundation block for future cryptographic constructions from the LWE assumption.

1.2.2 Tightly Secure Efficient Digital Signatures from the RSA assumption

As described in Section 1.1, the security of a cryptosystem is guaranteed by a certain computational hardness assumption. To prove the security of the cryptosystem, we reduce breaking the security to break the assumption (i.e., to solve the problem assumed to be hard in the underlying security assumption). There is a gap, which is called *reduction efficiency*, between hardnesses of breaking a cryptosystem and solving a security problem. The reduction efficiency is defined as the probability that breaking security of a cryptosystem leads solving a problem on which the security of the cryptosystem is based. We say that a reduction from security of a cryptosystem to an underlying problem is tight if its reduction efficiency is equal to 1 (that is, if we can break the cryptosystem, we can solve its underlying problem with probability 1). If a security reduction is tight, breaking the cryptosystem is as hard as solving the underlying problem. Hence, if we can prove the security of a cryptosystem with a highly efficient security reduction, the cryptosystem can be implemented with smaller parameter settings (that is, a smaller key size). We particularly focus on tightly secure digital signatures in the *random oracle* model.

The random oracle model, which was first introduced by Bellare and Rogaway in 1993 [BR93], is an idealized paradigm in which a hash function is viewed as an oracle that outputs a random value for every input query. Bellare and Rogaway in [BR96] proposed full domain hash (FDH) signature scheme that is implemented by the random oracle in the security proof. The FDH signature scheme is now used in a wide variety of applications, and serves as the foundation of several standards such as [RSA93]. The reduction efficiency of the FDH signatures was improved by Coron in [Cor02]. In [BR96], Bellare and Rogaway also proposed a probabilistic signature scheme (PSS) whose security is tightly reduced to the RSA assumption. Since the PSS is tightly secure only for longer random salts, Coron introduced a probabilistic full-domain hash (pFDH) implemented by the random oracle to prove that the PSS has a tight security reduction also for shorter random salts, but the Coron's signature scheme has a complex construction since it uses the random oracle multiple times. The above signature schemes are secure in the random oracle model, and their random oracles are replaced by concrete hash functions when implementing the signatures in the real world. A security proof for a cryptographic scheme in the random oracle model does not mean that it is secure in the real world, but it provides some kind of security guarantee, and it is still important in a practical sense to prove the security in the random oracle model.

In this thesis, we propose new digital signature scheme that is tightly secure based on the RSA assumption in the random oracle model. Our signatures

have a simpler construction than the previous tightly secure RSA-based signatures [BR96, Cor02], since the number of random oracles used in our signatures is less than those previous ones (and in fact, the number is minimum). The simplicity of our construction leads to lower implementation cost of the secure digital signatures. To prove the security of our signatures, we introduce a new technique called $\alpha - \beta$ *hiding technique*. This technique relies on the mathematics of the RSA, so it may become a useful tool to prove the security of other cryptosystems based on the RSA assumption. Our proposed signatures are tightly secure as well as the PSS [BR96] that is a foundation of PKCS #1 standard [RSA93], and have a simpler construction than the PSS since the number of random oracles (implemented by a hash function) is optimal. Therefore, our signatures may be an alternative for the PSS.

1.2.3 Efficient Blind Message Signatures from the RSA assumption

Digital signatures have many applications such as in e-government and e-business, but they cannot be applied to information systems in which secrecy is required for messages to be signed. Blind signatures provide a way to satisfy this requirement. In the blind signature schemes (actually a cryptographic *protocol* between two participants), we can obtain signatures for arbitrary messages while keeping the messages hidden.

Blind signatures are a variant of digital signatures that were first introduced by Chaum [Cha82]. They are a cryptographic protocol between two parties (user and signer) in which the user requests a signature for his message and gets the signature from the signer, where the signed message is hidden from the signer (blindness), and the number of signatures generated by the user is not larger than the number of runs of the blind signature protocol (unforgeability). In particular, because of the blindness, blind signatures have an important role in applications such as the electronic cash and electronic voting.

Chaum's blind signatures [Cha82] from the RSA signatures [RSA78] were not provably secure. In [BNPS03], Bellare et al. showed that the Chaum's blind signature scheme is provably secure, but the underlying assumption is not standard. Secure blind signatures from the standard assumptions in the random oracle model were proposed in [PS96, Poi98, AO00, Abe01, AO01], and the most efficient blind signatures in these works are the ones by Abe [Abe01].

In this thesis, we introduce a new notion *blind message signatures*, which has the following features. A signer $\mathcal{S}$ executes a blind signature protocol, $\mathcal{P}$, with a user $\mathcal{U}$ and $\mathcal{S}$ is divided into two parts,$\mathcal{S}_0$ and $\mathcal{S}_1$. $\mathcal{S}_0$ accepts a request from the user $\mathcal{U}$ and knows the identity of $\mathcal{U}$. $\mathcal{S}_0$ then runs the sub-protocol of $\mathcal{P}$ with

$\mathcal{U}$ (say $\mathcal{P}_0$) which is $\mathcal{P}$ excepting the final round. $\mathcal{S}_1$ executes the final round of $\mathcal{P}$ (say $\mathcal{P}_1$), i.e., $\mathcal{S}_1$ just sends a value to $\mathcal{U}$. Here, unless $\mathcal{S}_0$ and $\mathcal{S}_1$ collaborate, the protocol satisfies the requirements of blind signatures. A message m is hidden before use $\mathcal{U}$ releases the message m with a signature σ even if $\mathcal{S}_0$ and $\mathcal{S}_1$ collude.

We now assumes that the link between the message m and the user $\mathcal{U}$ revealed if $\mathcal{S}_0$ and $\mathcal{S}_1$ collude. Then we show an application of this concept, blind message signature. First, we assume that $\mathcal{S}_0$ and $\mathcal{S}_1$ do not collude usually. For example, in this application, $\mathcal{S}_0$ knows the identity of user, $\mathcal{U}$, and receives a signing request with some value B from $\mathcal{U}$. $\mathcal{S}_0$ then runs sub-protocol $\mathcal{P}_0$. After completing $\mathcal{P}_0$ with $\mathcal{U}$, $\mathcal{S}_0$ gives a string t to $\mathcal{S}_1$. Here, $\mathcal{S}_0$ keeps $(\mathcal{U}, t)$. $\mathcal{S}_1$, given t from $\mathcal{S}_0$, executes $\mathcal{P}_1$, i.e., computes Y and sends it to $\mathcal{U}$ (without knowing the identity of $\mathcal{U}$). Here, $\mathcal{S}_1$ keeps (t, Y). User $\mathcal{U}$, given Y, computes a signature σ for the message m. If $\mathcal{U}$ keeps the message m in secret for a certain period, (for example, m is a secret patent document), the message m is kept secret even if $\mathcal{S}_0$ and $\mathcal{S}_1$ collude. After a period, $\mathcal{U}$ releases m along with the signature σ. Since $\mathcal{S}_0$ and $\mathcal{S}_1$ do not collude usually, the privacy of (m, σ) is preserved, i.e., it is a blind signature. If a warrant of arrest is given to user $\mathcal{U}$ under suspicion of e.g., money laundering and illegal dealing of drugs, the police orders $\mathcal{S}_0$ and $\mathcal{S}_1$ to provide the record on $\mathcal{U}$. Given $(\mathcal{U}, t)$ and (t, Y) from $\mathcal{S}_0$ and $\mathcal{S}_1$,, the police traces (m, σ) to be the signature message of $\mathcal{U}$ from information $(\mathcal{U}, Y)$.

Our construction is the first efficient blind message signatures secure under the RSA assumption. The key generation and verification algorithms are the same as our digital signature scheme described in the last section. If the proposed digital signature is implemented to some information systems instead of the PSS, we can use the same signing and verification key to run our blind signature protocol, and also generate keys and verify signatures without changing the algorithms.

1.3 Thesis Outline

In Chapter 2, we introduce mathematical definitions used in this thesis. In Chapter 3, we propose FHE with more efficient homomorphic operation algorithms than the previous FHE schemes based on the standard assumptions. In Chapter 4, we show tightly secure efficient digital signatures based on the RSA assumption, which is thought as the most reliable standard assumption. In Chapter 5, we construct efficient blind message signatures also from the RSA assumption. In Chapter 6, we summarize the results presented in this thesis.

Chapter 2

Notations and Security

In this chapter, we give the mathematical preliminaries commonly required in all of this thesis. In Section 2.1, we list mathematical notations used in this thesis. In Section 2.2, we briefly introduce about the notion of provable security.

2.1 Notation

We denote the set of natural numbers by $\mathbb{N}$, the set of integers by $\mathbb{Z}$, the set of rational numbers by $\mathbb{Q}$, and the set of real numbers by $\mathbb{R}$. For any positive integer d, let $[d]$ be the set $\{1, 2, \ldots, d\}$. Let S be some set and $\mathcal{P}$ be some probability distribution over S, then we use $a \xleftarrow{U} S$ to denote that $a \in S$ is chosen from S uniformly at random, and use $b \xleftarrow{R} \mathcal{P}$ to denote that $b \in S$ is chosen along $\mathcal{P}$. We take all logarithms to base 2, unless otherwise noted. We denote by $U(S)$ the uniform distribution over the set. Let $\mathsf{negl}(\lambda)$ be a set of functions negligible in $\lambda \in \mathbb{N}$ For any two probability distributions $\mathcal{X}$ and $\mathcal{Y}$, we define the statistical distance between them by $\Delta(\mathcal{X}, \mathcal{Y}) := \frac{1}{2} \sum_{x \in S} |\Pr[x \xleftarrow{R} \mathcal{X}] - \Pr[x \xleftarrow{R} \mathcal{Y}]|$.

Vectors are in column form and are written using bold lower-case letters, e.g., $\boldsymbol{x}$, and the i-th element of a vector is denoted by x_i. We denote the ℓ_∞ norm (the maximum norm) of vector $\boldsymbol{x}$ by $\|\boldsymbol{x}\|_\infty$, and the ℓ_2 norm (the Euclidean norm) of $\boldsymbol{x}$ by $\|\boldsymbol{x}\|_2$. The inner product between two vectors is denoted by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$. Matrices are written by using bold capital letters, e.g., $\boldsymbol{X}$, and the i-th column vector of a matrix is denoted by $\boldsymbol{x}_i$. For matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, we define the ℓ_∞ and ℓ_2 norms of $\boldsymbol{X}$ as $\|\boldsymbol{X}\|_\infty := \max_{i \in [n]} \{\|\boldsymbol{x}_i\|_\infty\}$ and $\|\boldsymbol{X}\|_2 := \max_{i \in [n]} \{\|\boldsymbol{x}_i\|_2\}$, respectively. For matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, notation $\boldsymbol{X}^T \in \mathbb{R}^{n \times m}$ denotes the transpose of $\boldsymbol{X}$. For matrices $\boldsymbol{A} \in \mathbb{R}^{m \times n_1}$ and $\boldsymbol{B} \in \mathbb{R}^{m \times n_2}$, $[\boldsymbol{A} \parallel \boldsymbol{B}] \in \mathbb{R}^{m \times (n_1 + n_2)}$ denotes the concatenation of $\boldsymbol{A}$ with $\boldsymbol{B}$. When we refer to the $n \times n$ identity matrix, we denote it by $\boldsymbol{I}_n$.

2.2 Provable Security

Provable Security. *Provable security* is the central notion of modern cryptography. The securities of cryptosystems that we construct are guaranteed by giving a mathematical proof that breaking the securities is difficult.

To prove the security of a cryptosystem, we show that if assumption A holds, then cryptosystem S satisfies security notion N in security model M. Typical examples of M are the standard model and the random oracle model. In the standard model, adversaries are modeled as probabilistic polynomial time (PPT) algorithms (Turing machines). So, assumption A states that a certain mathematical problem is intractable for any PPT algorithm, cryptosystem S consists of PPT algorithms, and notion N defines that it is difficult for any PPT algorithm to break cryptosystem S. To show the above statement, we construct a PPT reduction (algorithm) that uses an adversary against notion N to break assumption A (i.e., solve the problem assumed to be intractable in A) with noticeable probability.

In cryptography, security assumptions are classified into the following five types of assumptions:

- **Standard assumptions.** The standard assumptions are security assumptions that are widely known and used in cryptography. For example, in the RSA assumption [RSA78], an adversary is given an RSA modulus N (that is a composite number of two primes), integer $e \in \mathbb{Z}$ such that $\gcd(e, \phi(N)) = 1$, and target $y \in \mathbb{Z}_N$. The task of the adversary is to compute integer x such that $x^e \equiv y \pmod{N}$. Other examples of the standard assumptions are the factoring assumption, the discrete log assumption, and the LWE assumption [Reg05].

- **Non-standard, falsifiable, non-interactive, and static-size assumptions.** A falsifiable assumption can be modeled as an interactive game between a challenger and adversary. At the end of the game, the challenger can efficiently decide whether the adversary won that game. The assumption states that every efficient adversary wins the game with negligible probability. If the falsifiable assumption is false, we can construct an efficient process to show that the assumption is false. The standard assumptions are also included in the falsifiable assumptions. In the non-interactive assumption, the adversary is not given access to any oracles, and the static-size assumptions do not depend on any system parameter and only depend on the security parameter. Examples of these assumptions are the strong-RSA assumption [CS00], the (asymmetric or symmetric) external Diffie-Hellman assumption, the Φ-hiding assumption [CMS99], the LPN assumption [BFKL94], and so on.

- **Falsifiable and non-interactive, but dynamic-size (q-type) assumptions.** Theses assumptions are the so called q-type assumptions, in which the size of the assumption grows dynamically. For example, in the q-weak Diffie-Hellman assumption [MSK02], the adversary is given $(g, g^x, g^{x^2}, \ldots, g^{x^q})$ and asked to compute $g^{1/x}$. Other examples of this type of assumption are the q-strong Diffie-Hellman assumption [BB04b], the q-bilinear Diffie-Hellman inversion assumption [BB04a], and the q-bilinear Diffie-Hellman exponent assumption [BBG05].

- **Falsifiable and interactive assumptions.** In the interactive assumptions, the adversary is given access to some oracles. For example, in the one-more RSA inversion assumption [BNPS03], the adversary is given $\ell + 1$-targets and access to an RSA-inversion oracle $(\cdot)^d \bmod N$ that it takes input $y \in \mathbb{Z}_N^*$ and returns its RSA-inverse $y^d \bmod N$, The task of the adversary is to compute the RSA-inverses of all the given targets while submitting at most ℓ queries to the oracle. Another example of this type of assumption is the LRSW assumption [LRSW99].

- **Unfalsifiable assumptions.** This type of assumption is not included in the falsifiable assumptions. A typical example of the unfalsifiable assumption is the Knowledge Exponent Assumption (KEA1) [Dam91], in which for any adversary that takes as input $q, g, g_1 = g^x$ and returns g_2, g_3 with $g_3 = g_2^x$, there exists an extractor that takes the same inputs as the adversary and outputs y such that $g^y = g_2$. In the falsifiable assumptions, the adversary is asked to produce a certain output on certain inputs. To show that such an assumption is false, we simply give an attack in the form of an adversary whose success probability is not negligible. However, the KEA1 assumption has more complex format. It states "for any adversary there exists an extractor such that ...". To show that this assumption is false, we must prove that there is an adversary for which there exists no extractor, so it is difficult to construct an efficient process to show that the assumption is false.

From the above, we can see that there are a great number of security assumptions in cryptography. Among these assumptions, the most desirable assumptions are the standard assumptions.

Random Oracle Model. The random oracle model, which was first introduced by Bellare and Rogaway [BR93], is a security model where hash functions are idealized. In this model, a hash function is considered as a random but public function. Anyone can query this oracle at arbitrary input x and obtain a completely random value. That is, the random oracle model allows a cryptographic scheme and adversaries to make such queries. In the random oracle model, we

can design a simple and efficient construction for cryptographic schemes. This is why a security reduction can give an adversary the view of oracles, namely, the reduction can control the view of the adversary. This means that the reduction not only knows what the adversary queries to the oracle but also can control what the adversary sees.

Formally, the random oracle model is a model where all the parties, e.g., algorithms and adversaries, have an oracle access to a random function $H(\cdot) : \{0,1\}^* \rightarrow \{0,1\}^n$ for some n. Whenever a new input x is queried, H chooses random output y and defines $y := H(x)$. A cryptographic scheme is secure in the random oracle model if the scheme satisfies a standard syntax, correctness and security definition, and the scheme and adversaries have an oracle access to $H(\cdot)$ in the security definition.

As shown in the above, the random oracle model is mathematically precise and provides formal definitions and security proofs. Let us see what this model said to us in the real world. It is desired that a cryptographic scheme secure in the random oracle model is still secure when replacing the random oracle with a real cryptographic hash function. But, in general, there are no theorem to say that, so we only have a heuristic way to consider about security in the random oracle model. It seems that cryptographic schemes proven secure in the random oracle model work well in the real world, but it is known that in general, such schemes cannot work. In fact, there are known results [CGH98, GK03, BBP04] where there exists a cryptographic construction that is secure in the random oracle model but not secure when the random oracle is replaced by any concrete hash function. This is a motivation for our studies in Chapters 4 and 5.

Chapter 3

Packing Messages and Optimizing Bootstrapping in GSW-FHE

In this chapter, we construct the first fully homomorphic encryption (FHE) scheme that encrypts *matrices* and supports homomorphic *matrix* addition and multiplication. This is a natural extension of packed FHE and thus supports more complicated homomorphic operations. We optimize the bootstrapping procedure of Alperin-Sheriff and Peikert (CRYPTO 2014) by applying our scheme. Our optimization decreases the lattice approximation factor from $\tilde{O}(n^3)$ to $\tilde{O}(n^{2.5})$. By taking a lattice dimension as a larger polynomial in a security parameter, we can also obtain the same approximation factor as the best known one of standard lattice-based public-key encryption *without* successive dimension-modulus reduction, which was essential for achieving the best factor in prior works on bootstrapping of standard lattice-based FHE.

3.1 Fully Homomorphic Encryption

3.1.1 Background

Fully homomorphic encryption (FHE) allows us to evaluate any function over encrypted data by only using public information. This can be used, for example, to outsource computations to remote servers without compromising privacy. Since the breakthrough work by Gentry [Gen09a, Gen09b], many different varieties of FHE have been proposed [DGHV10, BV11b, BV11a, BGV12, Bra12, LTV12, GSW13]. Here we focus on the noise-reduction techniques to see literature of FHE. Gentry [Gen09a, Gen09b] first constructed a pure FHE scheme using ideals of polynomial rings. van Dijk, Gentry, Halevi, and Vaikuntanathan [DGHV10] showed that a pure FHE scheme can also be constructed based on

integers. In [BV11b], Brakerski and Vaikuntanathan constructed a pure FHE scheme from polynomial rings, and proved that the scheme do not need to assume a certain security with the bootstrapping technique. The security of the constructions [Gen09b, Gen09a, DGHV10, BV11b] relies on the hardness of solving the non-standard (not well studied) mathematical problems. Brakerski and Vaikuntanathan [BV11a] used the new noise-management approach, dimension and modulus reduction, to construct a leveled FHE scheme based on a well-established problem on integer lattices. Brakerski, Gentry, and Vaikuntanathan [BGV12] then improved the [BV11b] scheme. In [Bra12], Brakerski constructed a leveled FHE scheme without the modulus reduction procedure. Coron, Lepoint, and Tibouchi [CLT14] presented a integer-based variant of [Bra12]. Since the noise-reduction techniques are still very complex, Gentry, Sahai, and Waters [GSW13] constructed a leveled FHE scheme (hereafter, referred to as GSW-FHE) without the dimension and modulus reduction technique. This scheme does not use any noise-reduction procedure, and so has very simple homomorphic operation algorithms (just matrix addition and multiplication). To date, the fastest (and simplest) FHE based on the *standard* lattice assumption is the one by Gentry, Sahai, and Waters [GSW13]. (hereafter, referred to as GSW-FHE). However, it is required to take heavy cost for evaluating a large number of ciphertexts. The way to deal with this issue is to *pack* multiple messages into one ciphertext.

Packing messages allows us to apply *single-instruction-multiple data* (SIMD) homomorphic operations to all encrypted messages. In the case where a remote server stores encrypted data and we want to retrieve certain data from this server, we first apply the equality function to every encrypted data. If the stored data have been packed into one ciphertext, we can do that by only one homomorphic evaluation of the equality function. Smart and Vercautren [SV10], for the first time, showed that applying the Chinese reminder theorem (CRT) to number fields partitions the message space of the Gentry's FHE [Gen09a, Gen09b] scheme into a vector of *plaintext slots*. On the standard lattice-based FHE schemes, Brakerski, Gentry, and Halevi [BGH13] used the method of [PVW08], which described a way to construct packed Regev's encryption [Reg05], to pack messages in the FHE variants [BV11b, BGV12, Bra12] of [Reg05].

Similar to the literature of FHE, several SIMD FHE schemes have been proposed [SV10, BGH13, CCK+13]. Smart and Vercautren [SV10], for the first time, showed that applying the Chinese reminder theorem (CRT) to number fields partitions the message space of the Gentry's FHE [Gen09a,Gen09b] scheme into a vector of plaintext slots. The technique of [SV10] can be applied to [BV11b]. Brakerski, Gentry, and Halevi [BGH13] used the technique of [PVW08], which described a way to construct packed Regev's encryption [Reg05], to pack messages in the FHE variants [BV11a, BGV12, Bra12] of [Reg05]. Cheon et al. [CCK+13] observed that the integer-based FHE [DGHV10, CLT14] can also be transformed

into a SIMD variant. A comparison of the recent progresses in normal and SIMD FHE leads the following natural and important question:

> *Can we construct a SIMD FHE scheme with simple homomorphic operations?*

We here say that homomorphic operations are simple if the FHE scheme does not proceed any noise-reduction technique (e.g., dimension-modulus reduction or bootstrapping) in its homomorphic operation algorithms.

In this chapter, we construct a matrix variant of [GSW13] (whose security is also based on the standard lattice assumption) to implement SIMD homomorphic operations, and describe how to bring out the potential of our scheme: specifically optimizing *bootstrapping*.

The bootstrapping technique [Gen09a, Gen09b] is currently the only way to go from limited amount of homomorphism to unlimited amount of homomorphism. The limited nature is caused by noise terms included in ciphertexts of all known FHE, which are needed to ensure security. Since homomorphic operations increases the noise level and the noise prevents us from correctly decrypting ciphertexts if the level increases too high, it is required to consider methods that reduce the noise. The bootstrapping technique is the one of such a methods, and achieved by homomorphically evaluating the decryption circuit of FHE.

There have recently been the significant progresses [DM15, HS15, CGGI16] in reducing the cost of the bootstrapping procedure. In [DM15], Ducas and Micciancio presented the new computation method of the NAND gate on the Regev's encryption [Reg05] that encrypts a single bit, and use a variant of [GSW13] to bootstrap Regev's ciphertexts. The implementation of their scheme performs a homomorphic NAND operation followed by bootstrapping (refreshing noise) in less than one second. In [HS15], Halevi and Shoup implemented a fast bootstrapping method, which is a packed variant of bootstrapping on HElib [HS14]. Their scheme has a slower clock time than [DM15], but supports SIMD homomorphic operations or has a larger plaintext space. In [CGGI16], Chillotti et al. reviewed the bootstrapping scheme of Ducas and Micciancio, and rewrote their scheme in terms of the external products. They obtain a speed up from less than one second to less than 0.1 second. All of the recent fast bootstrapping schemes [DM15, HS15, CGGI16] rely on the intractability of a problem on polynomial rings. The problem is called the ring-LWE, and reduced to a shortest vector problem on a special case of lattices.

There also exist the progresses on the standard lattice-based bootstrapping [BV14, AP14] schemes. Their progresses stem from the observation that noise terms in ciphertexts of GSW-FHE grow *asymmetrically*: for a parameter n (the dimension in the underlying lattice assumption), the noise of multiplication between two ciphertexts with noise size e_1 and e_2 grows to $e_1 + \mathrm{poly}(n) \cdot e_2$. For example, if

we want to multiply ℓ ciphertexts with the same noise size in *sequence*, the noise in the result increases by a factor of $\ell \cdot \mathrm{poly}(n)$, which is in contrast to the noise blowup factor by a multiplication tree, $\mathrm{poly}(n)^{\log \ell}$. To suppress the growth in noise from the bootstrapping procedure, the two recent developments [BV14, AP14] tried to *sequentialize* the decryption circuit.

Brakerski and Vaikuntanathan [BV14] transformed the decryption circuit of [GSW13] to a branching program by using the Barrington's theorem [Bar86], and homomorphically evaluated the program. Since the Barrington's theorem can convert the decryption circuit to a polynomial length branching program, evaluating the program increases the noise by a factor of $\mathrm{poly}(n)$. This procedure, however, has a significant drawback: the Barrington's theorem generates a branching program of *large* polynomial length. The scheme [BV14] also used a kind of *dimension leveraging* technique and successive dimension-modulus reduction to obtain the best approximation factor that is the same as standard lattice-based (plain) PKE.

Unlike most previous works, Alperin-Sheriff and Peikert [AP14] viewed the decryption as an arithmetic circuit. The decryption of all known standard lattice-based FHE consists of the inner product and rounding: for a ciphertext vector $\boldsymbol{c}$ and secret key vector $\boldsymbol{s}$, the decryption algorithm computes $\lfloor \langle \boldsymbol{c}, \boldsymbol{s} \rangle \rceil_2 \in \{0, 1\}$ (where $\lfloor \cdot \rceil_2$ is the rounding function introduced later). The authors observed that the inner product in the decryption can be expressed as a subset sum of the secret key elements. The subset sum can be computed only in the additive group, and the additive group is isomorphic to a group of cyclic permutations. The authors rewrote the inner product to the sequence of compositions of the cyclic permutations. Since this does not use the Barrington's theorem, the bootstrapping procedure of [AP14] can refresh ciphertexts faster and keep the noise growth in a *smaller* polynomial than that of [BV14], but the underlying security assumption was slightly stronger than that of [BV14] [1]. In addition, the procedure of [AP14] was not fully sequentialized, that is, there is a little room for sequentializing the decryption: the rounding.

3.1.2 Our Results

In this chapter, we construct the first FHE scheme that encrypts matrices and supports homomorphic matrix operations. This is a natural extension of packed FHE and supports more complicated homomorphic operations. Using this scheme, we fully sequentialize and thus optimize the bootstrapping procedure of [AP14]. The result of the optimization is described in the following:

[1] By using successive dimension-modulus reduction, [AP14] can also obtain the same approximation factor as that of [BV14].

Theorem 3.1.1. *Our optimized bootstrapping scheme can be secure assuming the hardness of approximating the standard lattice problem to within the factor* $\tilde{O}(n^{1.5}\lambda)$ *on any n dimensional lattices.*

For 2^λ hardness, we need to take $n = \Omega(\lambda)$. Asymptotically minimal selection of $n = \tilde{O}(\lambda)$ leads to the approximation factor $\tilde{O}(n^{2.5})$ for the underlying worst-case lattice assumption, which is smaller than $\tilde{O}(n^3)$, the factor of [AP14]. Using a kind of dimension leveraging technique: selecting a larger dimension $n = \lambda^{1/\epsilon}$ for $\epsilon \in (0, 1)$, we can also obtain the best known approximation factor, $\tilde{O}(n^{1.5+\epsilon})$, *without* successive dimension-modulus reduction, which was essential for achieving the best factor in the prior works on bootstrapping of standard lattice-based FHE.

3.1.3 Our Techniques

Matrix GSW-FHE. The starting point of our scheme is the GSW-FHE scheme. In that scheme, a ciphertext of a plaintext $m \in \{0, 1\}$ is a matrix $\boldsymbol{C} \in \mathbb{Z}_q^{(n+1)\times N}$ such that $\boldsymbol{sC} = m \cdot \boldsymbol{sG} + \boldsymbol{e}$ for a secret key vector $\boldsymbol{s} \in \mathbb{Z}_q^{n+1}$, small noise vector $\boldsymbol{e} \in \mathbb{Z}^N$, and fixed matrix $\boldsymbol{G} \in \mathbb{Z}_q^{(n+1)\times N}$. A simple extension of the plaintext space from bits to binary vectors cannot yield plaintext-slot-wise addition and multiplication. Instead, we use matrices to store binary vectors in their diagonal entries. Actually, our construction even supports homomorphic matrix addition and multiplication that are richer than homomorphic plaintext-slot-wise operations.

Let $\boldsymbol{S} \in \mathbb{Z}_q^{r\times(n+r)}$ be a secret key matrix, $\boldsymbol{B} \in \mathbb{Z}_q^{n\times m}$ be a Learning with Errors (LWE) matrix such that $\boldsymbol{SB} \approx \boldsymbol{0}$, and $\boldsymbol{G} \in \mathbb{Z}^{(n+r)\times N}$ be a fixed matrix. To encrypt a square integer matrix $\boldsymbol{M} \in \{0, 1\}^{r\times r}$, the ciphertext $\boldsymbol{C} \in \mathbb{Z}^{(n+r)\times N}$ must be of the form $\boldsymbol{BR} + \boldsymbol{XG}$ for a matrix $\boldsymbol{X} \in \mathbb{Z}^{(n+r)\times(n+r)}$ such that $\boldsymbol{SX} = \boldsymbol{MS}$, and small random matrix $\boldsymbol{R} \in \mathbb{Z}^{m\times N}$. The ciphertext $\boldsymbol{C}$ satisfies $\boldsymbol{SC} = \boldsymbol{E} + \boldsymbol{MSG}$ for a small noise matrix $\boldsymbol{E} \in \mathbb{Z}^{r\times N}$. Homomorphic matrix addition is just matrix addition. For example, given two ciphertexts $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, it holds that

$$\boldsymbol{S}(\boldsymbol{C}_1 + \boldsymbol{C}_2) = (\boldsymbol{E}_1 + \boldsymbol{E}_2) + (\boldsymbol{M}_1 + \boldsymbol{M}_2)\boldsymbol{SG}.$$

Homomorphic matrix multiplication corresponds to a simple preimage sampling and matrix multiplication. For a matrix $\boldsymbol{C} \in \mathbb{Z}_q^{(n+r)\times N}$, let $\boldsymbol{G}^{-1}(\boldsymbol{C})$ be the function that outputs a matrix $\boldsymbol{X}' \in \mathbb{Z}_q^{N\times N}$ such that $\boldsymbol{GX}' \equiv \boldsymbol{C} \pmod q$. If we let $\boldsymbol{X}_2' \xleftarrow{R} \boldsymbol{G}^{-1}(\boldsymbol{C}_2)$, then it holds that

$$\begin{aligned}\boldsymbol{SC}_1\boldsymbol{X}_2' &= (\boldsymbol{E}_1 + \boldsymbol{M}_1\boldsymbol{SG})\boldsymbol{X}_2' \\ &= \boldsymbol{E}_1\boldsymbol{X}_2' + \boldsymbol{M}_1\boldsymbol{E}_2 + \boldsymbol{M}_1\boldsymbol{M}_2\boldsymbol{SG}.\end{aligned}$$

Now, the problem is how to construct a matrix $\boldsymbol{X}$ such that $\boldsymbol{SX} = \boldsymbol{MS}$. By construction, $\boldsymbol{S}$ includes an identity matrix: $\boldsymbol{S} = [\boldsymbol{I} \parallel \boldsymbol{S}']$ for a matrix $\boldsymbol{S}' \in \mathbb{Z}_q^{r\times n}$.

The idea is to make $\boldsymbol{X}$ have $\boldsymbol{MS}$ in its top rows and 0 below. This $\boldsymbol{X}$ clearly satisfies the condition, but cannot publicly be computed without knowing the secret key. We translate the resulting symmetric scheme to the asymmetric one by using the method similar to [Bar10,Rot11]. In particular, let $\boldsymbol{M}_{(i,j)} \in \{0,1\}^{r\times r}$ $(i, j = 1, \ldots, r)$ be the matrix with 1 in the (i, j)-th entry and 0 in the others. We first publish symmetric encryptions of $\boldsymbol{M}_{(i,j)}$ for all $i, j \in [r]$. A ciphertext for a plaintext matrix $\boldsymbol{M}$ is publicly computed by summing up all encryptions of $\boldsymbol{M}_{(i,j)}$ such that the (i, j)-th entry of $\boldsymbol{M}$ is equal to 1, and using $\boldsymbol{B}$ to randomize the sum. Since the public key includes the ciphertexts that encrypt partial information of the secret key, security of our scheme cannot directly be proven from the LWE assumption. The way to deal with this problem is to introduce a circular security assumption.

Optimizing Bootstrapping of [AP14]. For a dimension d and modulus q, let $\boldsymbol{c} \in \{0,1\}^d$ be the $\ell - 1$-th column of a binary GSW-FHE ciphertext under a secret key $\boldsymbol{s} \in \mathbb{Z}_q^d$. Since the decryption algorithm of GSW-FHE computes $\lfloor\langle \boldsymbol{c}, \boldsymbol{s}\rangle\rceil_2$ ($\lfloor\cdot\rceil_2$ is the rounding function that outputs 1 if the input is close to $q/4$ and 0 otherwise), and $\langle \boldsymbol{c}, \boldsymbol{s}\rangle = \sum_{i=1}^{d} c_i s_i = \sum_{i\in[d]:c_i=1} s_i$, the decryption can be viewed as a subset sum of $\{s_i\}_{i\in[d]}$. To bootstrap ciphertexts, we only have to be able to compute additions in $\mathbb{Z}_q$ homomorphically. The additive group $\mathbb{Z}_q^+$ is isomorphic to a group of cyclic permutations, where $x \in \mathbb{Z}_q^+$ corresponds to a cyclic permutation that can be represented by an indicator vector with 1 in the x-th position. The permutation matrix for x can be obtained from cyclic rotations of its indicator vector. The addition in $\mathbb{Z}_q^+$ leads to the composition of the permutations (i.e., the multiplication of the corresponding permutation matrices), and the rounding function $\lfloor\cdot\rceil_2 : \mathbb{Z}_q \to \{0,1\}$ can be computed by summing the entries of the indicator vector corresponding to those values in $\mathbb{Z}_q$.

The bootstrapping procedure of [AP14] consists of two parts that compute an inner product and a rounding operation. The rounding checks equalities and computes summation. Our matrix GSW-FHE scheme allows us to rewrite the bootstrapping procedure except for the summation as a *sequence* of homomorphic matrix multiplications, while the procedure of [AP14] computes only the inner product part as a sequence. Intuitively, our optimization use the matrix GSW-FHE scheme to *sequentialize* the bootstrapping procedure of [AP14]. The asymmetric noise growth property is more effective in estimating how much noise the procedure yields.

The inner product can be computed by compositions of cyclic permutations. The bootstrapping procedure of [AP14] represents elements in $\mathbb{Z}_q$ as cyclic permutations, and evaluates their compositions by the naive matrix multiplication algorithm on the ciphertexts that encrypt every elements in the permutation matrices. Instead of that, our bootstrapping procedure uses homomorphic matrix multiplication to directly evaluate the compositions. The rounding part tests for

every value close to $q/4$ whether the output of the inner product part encrypts the permutation corresponding to the value, and sums their results (that are 0 or 1). Our procedure also use homomorphic matrix multiplication to realize the equality test. The result of the inner product is represented as an indicator vector, and encrypted component-wise in a SIMD encryption. The inner product equals to x if and only if its indicator vector has 1 in the x-th position. The homomorphic equality test between the inner product and x is computed by homomorphically permuting x-th slot to the first slot in the SIMD ciphertext. The result of the test is encrypted in the first slot. From the above, the bootstrapping procedure except for the summation can be represented as a sequence of $\tilde{O}(\lambda)$ homomorphic multiplications for a security parameter λ.

3.1.4 Related Work

Multilinear maps [GGH13a, CLT13, GGH15] are extensions of bilinear maps, and built from variants of FHE. The new multilinear maps construction of Gentry, Gorbunov, and Halevi [GGH15] also starts from GSW-FHE. Recall that in GSW-FHE, a ciphertext of $m \in \{0, 1\}$ is a matrix $\boldsymbol{C} \in \mathbb{Z}_q^{(n+1)\times N}$ such that $\boldsymbol{sC} = m \cdot \boldsymbol{sG} + \boldsymbol{e}$ for a secret key vector $\boldsymbol{s} \in \mathbb{Z}_q^{(n+1)}$ and small noise vector $\boldsymbol{e} \in \mathbb{Z}^N$. That is, valid ciphertexts of GSW-FHE have the secret key as the *approximate eigenvector* and the message as the eigenvalue. The multilinear maps construction of [GGH15] replaced the approximate eigenvector with the *approximate eigenspace* by increasing the dimension. In the construction, an encoding of $\boldsymbol{M} \in \mathbb{Z}^{r\times r}$ is a matrix $\boldsymbol{C} \in \mathbb{Z}_q^{N\times N}$ such that $\boldsymbol{SC} = \boldsymbol{E} + \boldsymbol{MS}$ for a random matrix $\boldsymbol{S} \in \mathbb{Z}_q^{r\times N}$ and small noise matrix $\boldsymbol{E} \in \mathbb{Z}^{r\times N}$. The approximate eigenspace is the matrix $\boldsymbol{S}$. To obtain the encoding $\boldsymbol{C}$, the construction samples a preimage of $\boldsymbol{MS} + \boldsymbol{E}$ for the function $f_{\boldsymbol{S}}(\boldsymbol{x}) = \boldsymbol{Sx} \bmod q$. In our scheme, a ciphertext $\boldsymbol{C} \in \mathbb{Z}_q^{N\times N}$ of $\boldsymbol{M} \in \mathbb{Z}^{r\times r}$ is a preimage of

$$\boldsymbol{BR} + \left(\frac{\boldsymbol{MS}}{\boldsymbol{0}}\right)\boldsymbol{G}$$

for the function $f_{\boldsymbol{G}}$. Since the ciphertext $\boldsymbol{C}$ satisfies $(\boldsymbol{SG})\boldsymbol{C} = \boldsymbol{M}(\boldsymbol{SG}) + \boldsymbol{E}$ for a small noise matrix $\boldsymbol{E} \in \mathbb{Z}^{r\times N}$, the matrix $\boldsymbol{SG}$ can be seen as the approximate eigenspace.

3.1.5 Organiation of This Chapter

In Section 3.2, we introduce some mathematics needed to go through with this chapter. In Section 3.3, we construct a variant of GSW-FHE whose plaintext space is a set of matrices over $\{0, 1\}$, while the GSW-FHE can encrypt a single bit (or element in $\mathbb{Z}_q$). In Section 3.4, we use the above construction to optimize the bootstrapping procedure proposed by Alperin-Sheriff and Peikert [AP14].

3.2 Preliminaries

3.2.1 Subgaussian

A real random variable X is subgaussian with parameter s if for all $t \in \mathbb{R}$, its (scaled) moment generating function holds $\mathbb{E}[\exp(2\pi tX)] \leq \exp(\pi s^2 t^2)$. Subgaussian random variables have the following two properties that can be easily obtained from the definition of subgaussian random variables:

- Homogeneity: If the subgaussian random variable X has parameter s, then cX is subgaussian with parameter cs.
- Pythagorean additivity: For two subgaussian random variables X_1 and X_2 (that is independent from X_1) with parameter s_1 and s_2, respectively, $X_1 + X_2$ is subgaussian with parameter $\sqrt{s_1^2 + s_2^2}$.

The above can be extended to vectors. A real random vector $\boldsymbol{x}$ is subgaussian with parameter s if for all real unit vectors $\boldsymbol{u}$, their marginal $\langle \boldsymbol{u}, \boldsymbol{x} \rangle$ is subgaussian with parameter s. It is clear from the definition that the concatenation of subgaussian variables or vectors, each of which has a parameter s and is independent of the prior one, is also subgaussian with parameter s. The homogeneity and Pythagorean additivity also hold from linearity of vectors. It is known that the euclidean norm of the subgaussian random vector has the following upper bound.

Lemma 3.2.1 ([Ver12]). *Let $\boldsymbol{x} \in \mathbb{R}^n$ be a random vector that has independent subgaussian coordinates with parameter s. Then there exists a universal constant C such that* $\Pr[\|\boldsymbol{x}\|_2 > C \cdot s\sqrt{n}] \leq 2^{-\Omega(n)}$.

To suppress the growth in noise, Gentry et al. [GSW13] made use of a procedure that decomposes a vector in binary representation. Alperin-Sheriff and Peikert [AP14] observed that instead of the decomposition procedure, using the following algorithm $\boldsymbol{G}^{-1}$ that samples a subgaussian random vector allows us to re-randomize errors in ciphertexts and tightly analyze the noise growth in [GSW13]. Lemma 3.2.2 can be extended to matrices in the obvious way. We here let $\boldsymbol{g}^T := (1, 2, 2^2, \ldots, 2^{\lceil \log q \rceil - 1})$ and $\boldsymbol{G} := \boldsymbol{g}^T \otimes \boldsymbol{I}_n$.

Lemma 3.2.2 ([AP14], which is adapted from [MP12]). *There is a randomized, efficiently computable function $\boldsymbol{G}^{-1} : \mathbb{Z}_q^n \to \mathbb{Z}^{n \cdot \lceil \log q \rceil}$ such that for any $\boldsymbol{a} \in \mathbb{Z}_q^n$, $\boldsymbol{x} \xleftarrow{R} \boldsymbol{G}^{-1}(\boldsymbol{a})$ is subgaussian with parameter $O(1)$ and $\boldsymbol{a} = [\boldsymbol{G}\boldsymbol{x}]_q$*

3.2.2 Homomorphic Encryption, Circular Security, and Bootstrapping

We here describe the syntax and security of homomorphic encryption scheme, and introduce a definition of circular security and the Gentry's bootstrapping theorem. Let $\mathcal{M}$ and C be the message and ciphertext space.

Definition 3.2.1 (Homomorphic Encryption)**.** *A homomorphic encryption scheme consists of four algorithms,* $\{\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}\}$.

- $\mathsf{KeyGen}(1^\lambda)$*: output a public encryption key* pk*, a secret decryption key* sk*, and a public evaluation key* evk.
- $\mathsf{Enc}_{\mathsf{pk}}(m)$*: using a public key* pk*, encrypt a plaintext* $m \in \mathcal{M}$ *into a ciphertext* $c \in C$.
- $\mathsf{Dec}_{\mathsf{sk}}(c)$*: using a secret key* sk*, recover the message encrypted in the ciphertext* c.
- $\mathsf{Eval}_{\mathsf{evk}}(f, c_1, \ldots, c_\ell)$*: using the evaluation key* evk*, output a ciphertext* $c_f \in C$ *that is obtained by applying the function* $f : \mathcal{M}^\ell \to \mathcal{M}$ *to* $c_1, \ldots, c_\ell$.

Security of homomorphic encryption is defined as follows:

Definition 3.2.2 (IND-CPA Security)**.** *Let* $\mathsf{HE} = \{\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}\}$ *be a homomorphic encryption scheme and consider the following game between the challenger and adversary:*

1. *The challenger generates* $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk}) \xleftarrow{R} \mathsf{KeyGen}(1^\lambda)$ *and sends* $(\mathsf{pk}, \mathsf{evk})$ *to the adversary.*
2. *The adversary sends a pair of messages* μ_0, μ_1 *to the challenger.*
3. *The challenger randomly samples* $b \xleftarrow{U} \{0, 1\}$ *and computes* $c^* \xleftarrow{R} \mathsf{Enc}_{\mathsf{pk}}(\mu_b)$*. It sends* c^* *to the adversary.*
4. *The adversary outputs* $b' \in \{0, 1\}$.

The advantage of an adversary $\mathcal{A}$ *is* $|\Pr[b' = b] - 1/2|$*, where* b *and* b' *are generated in the above game between the challenger and the adversary.*

To prove the security of our construction, we introduce a special kind of circular security for a homomorphic encryption scheme.

Definition 3.2.3 (Circular security)**.** *Let $\mathcal{K}$ be the key space defined by a security parameter λ. Let f be a function from $\mathcal{K}$ to C. A homomorphic encryption scheme* $\mathsf{HE} = \{\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval}\}$ *is circular secure with respect to f if for all probabilistic polynomial-time adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ in the following game is negligible in λ:*

1. *A challenger computes* $(\mathsf{pk}, \mathsf{sk}, \mathsf{evk}) \xleftarrow{R} \mathsf{KeyGen}(1^\lambda)$*, and chooses a bit* $b \xleftarrow{U} \{0, 1\}$.
2. *Let $f_+ : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ be a function that computes $f_+(x, y) := x + y \in \mathcal{M}$. The challenger computes a challenge ciphertext c^* as follows and sends it to $\mathcal{A}$.*
$$c^* := \begin{cases} \mathsf{Eval}_{\mathsf{evk}}(f_+, \mathsf{Enc}_{\mathsf{pk}}(0), f(\mathsf{sk})) & \text{if } b = 0, \\ \mathsf{Enc}_{\mathsf{pk}}(0) \in C & \text{otherwise.} \end{cases}$$
3. *$\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$.*

The advantage of $\mathcal{A}$ is $\Pr[b = b'] - 1/2$.

In LWE-based FHE schemes, $\mathsf{Eval}_{\mathsf{evk}}(f_+, \mathsf{Enc}_{\mathsf{pk}}(0), f(\mathsf{sk}))$ can be seen as a kind of ciphertexts that encrypt $f(\mathsf{sk})$. This is why we call the above security notion circular security.

3.2.3 Lattices and Learning with Errors Assumption

A *lattice* is a set of points in n-dimensional space with a periodic structure. Formally, the lattice is generated by given n-linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$ as
$$\mathcal{L} = \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\}.$$
Lattices are an extremely useful mathematical structure for cryptography, which provides security against quantum computers, worst-case security guarantee, and richer structure for constructing highly functional cryptosystems such as FHE, functional encryption, and so on. From the introduction of the lattices to cryptography by Ajtai [Ajt96], many cryptosystems are constructed on the lattices, and the most of them are proven secure under the *learning with errors (LWE)* assumption.

The LWE assumption was first introduced by Regev [Reg05]. The construction of this chapter relies on the decisional version of the LWE assumption:

Definition 3.2.4 (DLWE)**.** *For a security parameter λ, let $n := n(\lambda)$ be an integer dimension, let $q := q(\lambda) \geq 2$ be an integer modulus, and let $\chi := \chi(\lambda)$ be an error distribution over $\mathbb{Z}$.* $\mathsf{DLWE}_{n,q,\chi}$ *is the problem to distinguish the following*

two distributions: In the first distribution, a tuple $(\boldsymbol{a}_i, b_i)$ *is sampled from uniform over* $\mathbb{Z}_q^n \times \mathbb{Z}_q$*; In the second distribution,* $\boldsymbol{s} \xleftarrow{U} \mathbb{Z}_q^n$ *and then a tuple* $(\boldsymbol{a}_i, b_i)$ *is sampled by sampling* $\boldsymbol{a}_i \xleftarrow{U} \mathbb{Z}_q^n$*,* $e_i \xleftarrow{R} \chi$*, and setting* $b_i := \langle \boldsymbol{a}_i, \boldsymbol{s} \rangle + e_i \bmod q$*. The* $\mathsf{DLWE}_{n,q,\chi}$ *assumption is that* $\mathsf{DLWE}_{n,q,\chi}$ *is infeasible.*

GapSVP_γ is the problem to distinguish between the case in which a lattice has a vector shorter than $r \in \mathbb{Q}$, and the case in which all the lattice vectors are greater than $\gamma \cdot r$. SIVP_γ is the problem to find the set of short linearly independent vectors in a lattice. $\mathsf{DLWE}_{n,q,\chi}$ has reductions to the standard lattice assumptions as follows. These reductions take χ to be a discrete Gaussian distribution $D_{\mathbb{Z},\alpha q}$ (that is centered around 0 and has parameter αq for some $\alpha < 1$).

Corollary 3.2.1 ([Reg05, Pei09, MM11, MP12])**.** *Let* $q := q(n) \in \mathbb{N}$ *be a power of primes* $q := p^r$ *or a product of distinct prime numbers* $q := \prod_i q_i$ *(*$q_i := \mathrm{poly}(n)$ *for all i), and let* $\alpha \geq \sqrt{n}/q$*. If there exists an efficient algorithm that solves (average-case)* $\mathsf{DLWE}_{n,q,D_{\mathbb{Z},\alpha q}}$*,*

- *there exists an efficient quantum algorithm that can solve* $\mathsf{GapSVP}_{\tilde{O}(n/\alpha)}$ *and* $\mathsf{SIVP}_{\tilde{O}(n/\alpha)}$ *in the worst-case for any n-dimensional lattices.*
- *if in addition it holds that* $q \geq \tilde{O}(2^{n/2})$*, there exists an efficient classical algorithm that can solve* $\mathsf{GapSVP}_{\tilde{O}(n/\alpha)}$ *in the worst-case for any n-dimensional lattices.*

3.3 Matrix GSW-FHE

We translate [GSW13] to be able to encrypt a *matrix* and homomorphically compute *matrix* addition and multiplication. This is a natural extension of packed FHE schemes. In Section 3.3.1, we present our matrix FHE scheme. In Section 3.3.2, we discuss the relationship between our scheme and packed FHE schemes.

3.3.1 Construction

Let λ be the security parameter. Our scheme is parameterized by an integer lattice dimension n, an integer modulus q, and a distribution χ over $\mathbb{Z}$ that is assumed to be subgaussian , all of which depends on λ. We let $\ell := \lceil \log q \rceil$, $m := O((n + r) \log q)$, and $N := (n + r) \cdot \ell$. Let r be the number of bits to be encrypted, which defines the message space $\{0,1\}^{r \times r}$. The ciphertext space is $\mathbb{Z}_q^{(n+r) \times N}$. Our scheme uses the rounding function $\lfloor \cdot \rceil_2$ that for any $x \in \mathbb{Z}_q$, $\lfloor x \rceil_2$ outputs 1 if x is close to $q/4$, and 0 otherwise. Recall that $\boldsymbol{g}^T = (1, 2, \ldots, 2^{\ell-1})$ and $\boldsymbol{G} = \boldsymbol{g}^T \otimes \boldsymbol{I}_{n+r}$.

- $\mathsf{KeyGen}(1^\lambda, r)$: Set the parameters n, q, m, ℓ, N, and χ as described above. Sample a uniformly random matrix $\boldsymbol{A} \xleftarrow{U} \mathbb{Z}_q^{n\times m}$, secret key matrix $\boldsymbol{S}' \xleftarrow{R} \chi^{r\times n}$, and noise matrix $\boldsymbol{E} \xleftarrow{R} \chi^{r\times m}$. Let $\boldsymbol{S} := [\boldsymbol{I}_r \parallel -\boldsymbol{S}'] \in \mathbb{Z}_q^{r\times(n+r)}$. We denote by $\boldsymbol{s}_i^T$ the i-th row of $\boldsymbol{S}$. Set

$$\boldsymbol{B} := \begin{pmatrix} \boldsymbol{S}'\boldsymbol{A} + \boldsymbol{E} \\ \boldsymbol{A} \end{pmatrix} \in \mathbb{Z}_q^{(n+r)\times m}.$$

  Let $\boldsymbol{M}_{(i,j)} \in \{0,1\}^{r\times r}$ $(i, j = 1, \ldots, r)$ be the matrix with 1 in the (i, j)-th position and 0 in the others. For all $i, j = 1, \ldots, r$, first sample $\boldsymbol{R}_{(i,j)} \xleftarrow{U} \{0,1\}^{m\times N}$, and set

$$\boldsymbol{P}_{(i,j)} := \boldsymbol{B}\boldsymbol{R}_{(i,j)} + \begin{pmatrix} \boldsymbol{M}_{(i,j)}\boldsymbol{S} \\ \boldsymbol{0} \end{pmatrix} \boldsymbol{G} \in \mathbb{Z}_q^{(n+r)\times N}.$$

  Output $\mathsf{pk} := (\{\boldsymbol{P}_{(i,j)}\}_{i,j\in[r]}, \boldsymbol{B})$ and $\mathsf{sk} := \boldsymbol{S}$.

- $\mathsf{SecEnc}_{\mathsf{sk}}(\boldsymbol{M} \in \{0,1\}^{r\times r})$: Sample a random matrices $\boldsymbol{A}' \xleftarrow{U} \mathbb{Z}_q^{n\times N}$ and $\boldsymbol{E} \xleftarrow{R} \chi^{r\times N}$, parse $\boldsymbol{S} = [\boldsymbol{I}_r \parallel -\boldsymbol{S}']$, and output the ciphertext

$$\boldsymbol{C} := \left[\begin{pmatrix} \boldsymbol{S}'\boldsymbol{A}' + \boldsymbol{E} \\ \boldsymbol{A}' \end{pmatrix} + \begin{pmatrix} \boldsymbol{M}\boldsymbol{S} \\ \boldsymbol{0} \end{pmatrix} \boldsymbol{G} \right]_q \in \mathbb{Z}_q^{(n+r)\times N}.$$

- $\mathsf{PubEnc}_{\mathsf{pk}}(\boldsymbol{M} \in \{0,1\}^{r\times r})$: Sample a random matrix $\boldsymbol{R} \xleftarrow{U} \{0,1\}^{m\times N}$, and output the ciphertext

$$\boldsymbol{C} := \boldsymbol{B}\boldsymbol{R} + \sum_{i,j\in[r]:\boldsymbol{M}[i,j]=1} \boldsymbol{P}_{(i,j)} \in \mathbb{Z}_q^{(n+r)\times N},$$

  where $\boldsymbol{M}[i, j]$ is the (i, j)-th element of $\boldsymbol{M}$.

- $\mathsf{Dec}_{\mathsf{sk}}(\boldsymbol{C})$: Output the matrix $\boldsymbol{M} = (\lfloor \langle \boldsymbol{s}_i, \boldsymbol{c}_{j\ell-1} \rangle \rceil_2)_{i,j\in[r]} \in \{0,1\}^{r\times r}$.

- $\boldsymbol{C}_1 \oplus \boldsymbol{C}_2$: Output $\boldsymbol{C}_{add} := \boldsymbol{C}_1 + \boldsymbol{C}_2 \in \mathbb{Z}_q^{(n+r)\times N}$ as the result of homomorphic addition between the input ciphertexts.

- $\boldsymbol{C}_1 \odot \boldsymbol{C}_2$: Output $\boldsymbol{C}_{mult} := \boldsymbol{C}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2) \in \mathbb{Z}_q^{(n+r)\times N}$ as the result of homomorphic multiplication between the input ciphertexts.

Definition 3.3.1. *We say that a ciphertext $\boldsymbol{C}$ encrypts a plaintext matrix $\boldsymbol{M}$ with noise matrix $\boldsymbol{E}$ if $\boldsymbol{C}$ is an encryption of $\boldsymbol{M}$ and $\boldsymbol{E} = \boldsymbol{S}\boldsymbol{C} - \boldsymbol{M}\boldsymbol{S}\boldsymbol{G}$* (mod q).

The following lemma states the correctness of our asymmetric encryption. Similar to this, the correctness of our symmetric encryption can be proven immediately.

Lemma 3.3.1. *If a ciphertext $\boldsymbol{C}$ encrypts a plaintext matrix $\boldsymbol{M} \in \{0,1\}^{r\times r}$ with noise matrix $\boldsymbol{E}'$ such that $\|\boldsymbol{E}'\|_\infty < q/8$, then $\mathsf{Dec}_{\mathsf{sk}}(\boldsymbol{C}) = \boldsymbol{M}$.*

Proof. We have

$$\begin{aligned}\boldsymbol{SC} &= \boldsymbol{S}\left(\boldsymbol{BR} + \textstyle\sum_{i,j\in[r]:\boldsymbol{M}[i,j]=1} \boldsymbol{BR}_{(i,j)} + \left(\dfrac{\boldsymbol{MS}}{\boldsymbol{0}}\right)\boldsymbol{G}\right)\\ &= \boldsymbol{ER} + \textstyle\sum_{i,j\in[r]:\boldsymbol{M}[i,j]=1} \boldsymbol{ER}_{(i,j)} + \boldsymbol{MSG}\\ &= \boldsymbol{ER} + \textstyle\sum_{i,j\in[r]:\boldsymbol{M}[i,j]=1} \boldsymbol{ER}_{(i,j)}\\ &\quad + [\boldsymbol{M}(\boldsymbol{g}^T \otimes \boldsymbol{I}_r) \,\|\, -\boldsymbol{MS}'(\boldsymbol{g}^T \otimes \boldsymbol{I}_n)]\end{aligned}$$

Let $\boldsymbol{E}' := \boldsymbol{E}(\boldsymbol{R} + \sum_{i,j\in[r]:\boldsymbol{M}[i,j]=1} \boldsymbol{R}_{(i,j)})$, then $\|\boldsymbol{E}'\|_\infty < q/8$. Because of $2^{\ell-2} \in [q/4, q/2)$, for all $i, j = 1, \ldots, r$, it holds that $\langle \boldsymbol{s}_i, \boldsymbol{c}_{j\ell-1}\rangle \approx q/4$ if $m_{i,j} = 1$, and $\langle \boldsymbol{s}_i, \boldsymbol{c}_{j\ell-1}\rangle \approx 0$ otherwise. □

Security of SecEnc directly holds from $\mathsf{DLWE}_{n,q,\chi}$. For a matrix $\boldsymbol{M} \in \{0,1\}^{r\times r}$, let $f_{\boldsymbol{M}}$ be a function from $\mathbb{Z}_q^{r\times(n+r)}$ to $\mathbb{Z}_q^{(n+r)\times N}$ such that for a matrix $\boldsymbol{S} \in \mathbb{Z}_q^{r\times(n+r)}$,

$$f_{\boldsymbol{M}}(\boldsymbol{S}) = \left(\frac{\boldsymbol{MS}}{\boldsymbol{0}}\right)\boldsymbol{G} \in \mathbb{Z}_q^{(n+r)\times N}.$$

The security of PubEnc holds by $\mathsf{DLWE}_{n,q,\chi}$ and assuming our scheme circular secure with respect to $f_{\boldsymbol{M}_{(i,j)}}$. The IND-CPA security of our scheme is immediately proven from the following lemma:

Lemma 3.3.2. *Let $\boldsymbol{B}, \boldsymbol{M}_{(i,j)}, \boldsymbol{R}_{(i,j)}, \boldsymbol{P}_{(i,j)}$ $(i, j = 1, \ldots, r)$ be the matrices generated in* KeyGen*, and $\boldsymbol{R}$ be the matrix generated in* PubEnc*. For every $i, j = 1, \ldots, r$, if our scheme is circular secure with respect to $f_{\boldsymbol{M}_{(i,j)}}$ and $\mathsf{DLWE}_{n,q,\chi}$ holds, then the joint distribution $(\boldsymbol{B}, \boldsymbol{BR}_{(i,j)}, \boldsymbol{P}_{(i,j)}, \boldsymbol{BR})$ is computationally indistinguishable from uniform over $\mathbb{Z}_q^{(n+r)\times m} \times \mathbb{Z}_q^{(n+r)\times N} \times \mathbb{Z}_q^{(n+r)\times N} \times \mathbb{Z}_q^{(n+r)\times N}$.*

We need to estimate the noise growth by the evaluation of homomorphic matrix addition and multiplication. Similar to [AP14], we employ the properties of subgaussian random variables for tight analysis. We collect the results of the estimation in the following lemma.

Lemma 3.3.3. *Let $\boldsymbol{S} \in \mathbb{Z}^{r\times(n+r)}$ be a secret key matrix. Let $\boldsymbol{C}_1 \in \mathbb{Z}_q^{(n+r)\times N}$ and $\boldsymbol{C}_2 \in \mathbb{Z}_q^{(n+r)\times N}$ be ciphertexts that encrypt $\boldsymbol{M}_1 \in \{0,1\}^{r\times r}$ and $\boldsymbol{M}_2 \in \{0,1\}^{r\times r}$ with noise matrices $\boldsymbol{E}_1 \in \mathbb{Z}^{r\times N}$ and $\boldsymbol{E}_2 \in \mathbb{Z}^{r\times N}$, respectively. Let $\boldsymbol{e}_{1,i}^T \in \mathbb{Z}^{1\times N}$ $(i = 1, \ldots, r)$ be the i-th row vector of $\boldsymbol{E}_1$. Let $\boldsymbol{C}_{add} := \boldsymbol{C}_1 \oplus \boldsymbol{C}_2$ and $\boldsymbol{C}_{mult} \xleftarrow{R} \boldsymbol{C}_1 \odot \boldsymbol{C}_2$. Then, we have*

$$\begin{aligned}\boldsymbol{SC}_{add} &= \boldsymbol{E}_{add} + (\boldsymbol{M}_1 + \boldsymbol{M}_2)\boldsymbol{SG} \in \mathbb{Z}_q^{r\times N},\\ \boldsymbol{SC}_{mult} &= \boldsymbol{E}_{mult} + (\boldsymbol{M}_1\boldsymbol{M}_2)\boldsymbol{SG} \in \mathbb{Z}_q^{r\times N},\end{aligned}$$

where $\boldsymbol{E}_{add} := \boldsymbol{E}_1 + \boldsymbol{E}_2$ and $\boldsymbol{E}_{mult} := \boldsymbol{E} + \boldsymbol{M}_1\boldsymbol{E}_2$. In particular, $\boldsymbol{E}$ has in the i-th row the independent subgaussian entries with parameter $O(\|\boldsymbol{e}_{1,i}\|_2)$.

Proof. We can immediately prove the statements for $\boldsymbol{C}_{add}$. For $\boldsymbol{C}_{mult}$, we have

$$\begin{aligned}\boldsymbol{S}\boldsymbol{C}_{mult} &= \boldsymbol{S}\boldsymbol{C}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)\\ &= (\boldsymbol{E}_1 + \boldsymbol{M}_1\boldsymbol{S}\boldsymbol{G})\boldsymbol{G}^{-1}(\boldsymbol{C}_2)\\ &= \boldsymbol{E}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2) + \boldsymbol{M}_1\boldsymbol{E}_2 + \boldsymbol{M}_1\boldsymbol{M}_2\boldsymbol{S}\boldsymbol{G}.\end{aligned}$$

From the subgaussian properties and Lemma 3.2.2, we can see that the i-th row entries of $\boldsymbol{E} := \boldsymbol{E}_1\boldsymbol{G}^{-1}(\boldsymbol{C}_2)$ are independent subgaussian with parameter $O(\|\boldsymbol{e}_{1,i}\|_2)$. □

Similar to the original GSW scheme, our scheme also has the asymmetric noise growth property, and thereby computing a polynomial length chain of homomorphic multiplications incurs the noise growth by a multiplicative polynomial factor. For ease of analyzing our optimized bootstrapping procedure described in the next section, we set the following corollary immediately proven from Lemma 3.3.3 and the properties of subgaussian random variables. This corollary includes the fixed ciphertext $\boldsymbol{G} \in \mathbb{Z}^{(n+r)\times N}$ of the message $\boldsymbol{I}_r$ with noise $\boldsymbol{0}$. This makes the noise in the output ciphertext subgaussian and independent from the noise in the input ciphertexts.

Corollary 3.3.1. *For $i = 1, \ldots, k$, let $\boldsymbol{C}_i \in \boldsymbol{Z}^{(n+r)\times N}$ be a ciphertext that encrypts a message matrix $\boldsymbol{M}_i \in \{0,1\}^{r\times r}$ such that for a matrix $\boldsymbol{E} \in \mathbb{Z}^{r\times N}$, $\|(\boldsymbol{M}_i\boldsymbol{E})^T\|_2 \leq \|\boldsymbol{E}^T\|_2$ with noise matrix $\boldsymbol{E}_i \in \mathbb{Z}^{r\times N}$. Let*

$$\boldsymbol{C} \xleftarrow{R} \bigodot_{i=1}^{k} \boldsymbol{C}_i \odot \boldsymbol{G} = \boldsymbol{C}_1 \odot (\boldsymbol{C}_2 \odot (\cdots (\boldsymbol{C}_{k-1} \odot (\boldsymbol{C}_k \odot \boldsymbol{G}))) \cdots).$$

For $i = 1, \ldots, k$, let $\boldsymbol{e}_i^T$ be a row vector of $\boldsymbol{E}_i$ whose norm is equal to $\|\boldsymbol{E}_i^T\|_2$, and $\boldsymbol{e}^T := [\boldsymbol{e}_1^T \parallel \boldsymbol{e}_2^T \parallel \cdots \parallel \boldsymbol{e}_k^T] \in \mathbb{Z}^{1\times kN}$. Then the noise matrix of $\boldsymbol{C}$ has in every row the independent subgaussian entries with parameter $O(\|\boldsymbol{e}\|_2)$.

Proof. The ciphertext $\boldsymbol{C}$ encrypts $\prod_{i=1}^{k} \boldsymbol{M}_i$ with noise $\boldsymbol{E}_1\boldsymbol{X}_1 + \sum_{i=2}^{k}(\prod_{j=1}^{i-1} \boldsymbol{M}_j)\boldsymbol{E}_i\boldsymbol{X}_i$, where $\boldsymbol{X}_i$ is the matrix used in the evaluation of each $\odot$. By Lemma 3.3.3, the elements of $\boldsymbol{E}_1\boldsymbol{X}_1$ in every row are independent and subgaussian with parameter $O(\|\boldsymbol{e}_1\|_2)$. Since we have $\|(\boldsymbol{M}_i\boldsymbol{E})^T\|_2 \leq \|\boldsymbol{E}^T\|_2$, $(\prod_{j=1}^{i-1} \boldsymbol{M}_j)\boldsymbol{E}_i\boldsymbol{X}_i$ has in its every row the independent subgaussian entries with parameter $O(\|\boldsymbol{e}_i\|_2)$. By the Pythagorean additivity of subgaussian random variables, $\boldsymbol{E}_1\boldsymbol{X}_1 + \sum_{i=2}^{k}(\prod_{j=1}^{i-1} \boldsymbol{M}_j)\boldsymbol{E}_i\boldsymbol{X}_i$ has in every row the independent subgaussian entries with parameter $O(\|\boldsymbol{e}\|_2)$. □

3.3.2 Relation to Packed FHE

The matrix GSW-FHE above is a natural extension of packed FHE. Plaintext slots in packed FHE correspond to diagonal entries of plaintext matrices in the matrix

GSW-FHE scheme. It is easy to see that we can correctly compute homomorphic slot-wise addition and multiplication. In applications of packed FHE such as in [GHS12], we may want to permute plaintext slots. This can be achieved by multiplying the encryptions of a permutation and its inverse from left and right. Security and correctness of the following algorithms clearly holds from Lemmas 3.3.2 and 3.3.3.

Let $r > 0$ be an integer. For any permutation $\sigma : \{1, \ldots, r\} \to \{1, \ldots, r\}$, its permutation matrix Σ is given as: $\Sigma := [\boldsymbol{e}_{\sigma(1)} \parallel \cdots \parallel \boldsymbol{e}_{\sigma(r)}] \in \{0,1\}^{r \times r}$, where $\boldsymbol{e}_i \in \{0,1\}^r$ ($i \in [r]$) is the standard basis vector with 1 in the i-th position and 0 in the others.

- $\mathsf{SwitchKeyGen}(\boldsymbol{S}, \sigma)$: Given a secret key matrix $\boldsymbol{S} \in \mathbb{Z}_q^{r \times (n+r)}$ and a permutation σ, let $\Sigma \in \{0,1\}^{r \times r}$ be the permutation matrix of σ, and generate

$$\boldsymbol{W}_{\sigma} \xleftarrow{R} \mathsf{SecEnc}_{\boldsymbol{S}}(\Sigma),$$
$$\boldsymbol{W}_{\sigma^{-1}} \xleftarrow{R} \mathsf{SecEnc}_{\boldsymbol{S}}(\Sigma^T).$$

  Output the switch key $\mathsf{ssk}_{\sigma} := (\boldsymbol{W}_{\sigma}, \boldsymbol{W}_{\sigma^{-1}})$.

- $\mathsf{SlotSwitch}_{\mathsf{ssk}_{\sigma}}(\boldsymbol{C})$: Take as input a switch key ssk_{σ} and a ciphertext $\boldsymbol{C}$, output

$$\boldsymbol{C}_{\sigma} \xleftarrow{R} \boldsymbol{W}_{\sigma} \odot (\boldsymbol{C} \odot (\boldsymbol{W}_{\sigma^{-1}} \odot \boldsymbol{G})),$$

  where $\boldsymbol{G} \in \mathbb{Z}^{(n+r) \times N}$ is the fixed encryption of $\boldsymbol{I}_r$ with noise zero.

One nice feature of our plaintext-slot switching is not to suffer from the inconvenience of the security as in [BGH13]: we do not have to use a larger modulus than the matrix GSW-FHE scheme. Brakerski et al. [BGH13] made use of a larger modulus $Q = 2^{\ell} q$ to suppress noise growth when switching decryption keys, so the security of the plaintext-slot switching in [BGH13] must have related to Q. The larger modulus leads the larger modulus-to-noise ratio. To obtain the same security level as the underlying SIMD scheme of [BGH13], it was required to select a larger dimension. As opposed to this, our plaintext-slot switching can use the same modulus as the matrix GSW-FHE scheme.

3.3.3 Discussion

The underlying GSW-FHE has a variant from Ring Learning With Errors (RLWE) problem and ID/attribute-based constructions. According to this, we discuss such variants of our scheme.

A RLWE-based Variant. The RLWE problem was first introduced by Lyubashevsky, Peikert, and Regev [LPR10]. The paper [LPR10] showed that the problem can be reduced to the well-established shortest vector problem (SVP) on ideal lattices.

Definition 3.3.2. *For a security parameter λ, let $f(x) := x^d + 1$ where $d := d(\lambda)$ is a power of 2. Let $q := q(\lambda) \geq 2$ be an integer. Let $R := \mathbb{Z}[X]/(f(x))$ and $R_q := R/qR$. Let $\chi := \chi(\lambda)$ be a distribution over R. The $\mathsf{RLWE}_{n,q,\chi}$ problem is to distinguish the following two distributions: In the first distribution, (a_i, b_i) is sampled from R_q^2 uniformly. In the second distribution, one first samples s from R_q uniformly , and samples (a_i, b_i) by sampling $a_i \xleftarrow{U} R_q$, $e_i \xleftarrow{R} \chi$ and setting $b_i := a_i s + e_i$. The $\mathsf{RLWE}_{n,q,\chi}$ assumption is that the $\mathsf{RLWE}_{n,q,\chi}$ problem is infeasible.*

The RLWE variant of our scheme starts with the LPR encryption [LPR10], specifically with a multibit variant of the LPR encryption. A public key of the encryption is a tuple of RLWE instances for a common ring element $a \xleftarrow{U} R_q$:

$$\boldsymbol{a} := \left(\begin{array}{c} a \cdot s_1 + e_1 \\ a \cdot s_2 + e_2 \\ \vdots \\ a \cdot s_r + e_r \\ \hline a \end{array}\right) \in R_q^{(r+1)},$$

where for all $i \in [r]$ $s_i \xleftarrow{R} \chi$ and $e_i \xleftarrow{R} \chi$. As shown in [LPR10], one can sample s_i from the noise distribution χ. The corresponding secret key is a $r \times (r+1)$ matrix $\boldsymbol{S}$ over R_q:

$$\boldsymbol{S} := \left[\begin{array}{c|c} \boldsymbol{I}_r & \begin{array}{c} -s_1 \\ \vdots \\ -s_r \end{array} \end{array}\right] \in R_q^{r \times (r+1)},$$

where $\boldsymbol{Sa} = \boldsymbol{e}$ is a small vector in R_q^r. To encrypt $(0, \ldots, 0) \in \{0,1\}^r$, one first samples a random short element $r \xleftarrow{R} \chi$ and a short vector $\boldsymbol{e}' \xleftarrow{R} \chi^{(r+1)}$, and outputs $\boldsymbol{c} := \boldsymbol{a} \cdot r + \boldsymbol{e}' \in R_q^{(r+1)}$. To encrypt $(m_1, \ldots, m_r) \in \{0,1\}^r$, one adds $m_1 \cdot \lfloor q/2 \rfloor, \ldots, m_r \cdot \lfloor q/2 \rfloor \in R_q$ to the first r elements of $\boldsymbol{c}$. The decryption computes

$$\boldsymbol{Sc} = \boldsymbol{e} \cdot r + \boldsymbol{Se}' + \left(\begin{array}{c} m_1 \cdot \lfloor q/2 \rfloor \\ \vdots \\ m_r \cdot \lfloor q/2 \rfloor \end{array}\right) \in R_q^r,$$

and for each $i \in [r]$ outputs $m_i = 0$ or $m_i = 1$ depending on whether or not the i-th element of $\boldsymbol{Sc}$ is small.

For an integer $r > 0$, the message space of our RLWE variant is $\{0,1\}^{r\times r}$. Let $\ell := \lceil \log q \rceil$ and $N := (r+1)\cdot\ell$. Let $\boldsymbol{g}^T := (1, 2, \ldots, 2^{\ell-1}) \in R_q^{1\times\ell}$ and $\boldsymbol{G} := \boldsymbol{g}^T \otimes \boldsymbol{I}_{(r+1)} \in R_q^{(r+1)\times N}$. We can define the $\boldsymbol{G}^{-1}(\cdot)$ function for polynomial-ring elements as well as for integer matrices: There exists a deterministic polynomial-time algorithm $\boldsymbol{G}^{-1}(\cdot)$ such that for any integer $k > 0$ and for any $\boldsymbol{C} \in R_q^{(r+1)\times k}$, we have $\boldsymbol{C} = \boldsymbol{G}\boldsymbol{G}^{-1}(\boldsymbol{C}) \in R_q^{(r+1)\times k}$. Similar to our LWE-based construction, we publish as a part of the public key the secret key encryptions of partial plaintext matrices. The partial plaintext matrices are masked by the LPR encryptions. Let $\boldsymbol{C}' \in R_q^{(r+1)\times N}$ be N LPR encryptions. For all $i, j \in [r]$, the public key $\boldsymbol{P}_{(i,j)}$ is

$$\boldsymbol{P}_{(i,j)} := \boldsymbol{C}' + \left[\frac{\boldsymbol{M}_{(i,j)}\boldsymbol{S}}{\boldsymbol{0}}\right]\boldsymbol{G} \in R_q^{(r+1)\times N}.$$

To encrypt a plaintext matrix publicly, we randomize the corresponding public keys by other N LPR encryptions. That is, an encryption of a message $\boldsymbol{M} \in \{0,1\}^{r\times r}$ is

$$\boldsymbol{C} := \boldsymbol{C}'' + \sum_{i,j\in[r]:\boldsymbol{M}[i,j]=1} \boldsymbol{P}_{(i,j)} \in R_q^{(r+1)\times N}.$$

The decryption, homomorphic addition, and homomorphic multiplication are the same as them of the LWE based scheme. Since multiplying the secret key matrix to the LPR encryptions leads a small error matrix in $R_q^{r\times N}$, correctness of the decryption holds as in the LWE case. Since the matrix $\boldsymbol{C}''$ masking the sum of $\boldsymbol{P}_{(i,j)}$ is indistinguishable from a $(r+1)\times N$ random matrix over R_q by the security of the LPR encryption scheme, the ciphertext $\boldsymbol{C}$ is also indistinguishable from a random in $R_q^{(r+1)\times N}$.

Our RLWE variant is more efficient than the LWE-based one, but is not as efficient as the previous RLWE-based SIMD FHE schemes. This is because the previous schemes use the dimension-reduction algorithm [BV11a, BV14], which is much more efficient for RLWE-based FHE schemes than LWE-based ones.

ID/Attribute-based Constructions. For simplicity, we focus only on the ID-based variant. The same argument described here can easily adopted to the attribute-based case.

As the same reason that FHE schemes before GSW-FHE can not be transformed into the ID-based ones, our scheme can not be ID-based. Recall that our scheme publishes as the public key secret key encryptions of partial plaintexts. Since they need to be encryptions under the secret key based on an ID, the public key needs to be user-specific, and so is not ID-based.

3.4 Optimizing Bootstrapping

We describe how to optimize the bootstrapping procedure of [AP14] by using our scheme. In Section3.4.1, we present the optimized bootstrapping procedure outlined in Section 3.1.3, whose correctness and security are discussed in Section 3.4.2.

3.4.1 Optimized Procedure

Let Q be the modulus of the ciphertext to be refreshed. Using the dimension-modulus reduction technique [BV11a,BV14], we can publicly switch the modulus and the dimension to the arbitrary and possibly smaller ones $q, d = \tilde{O}(\lambda)$. Here, q has the form $q := \prod_{i=1}^{t} r_i$, where r_i are small and powers of distinct primes (and hence pairwise coprime). The following lemma allows us to choose a sufficiently large q so that the correctness of the dimension-modulus reduction holds by letting it be the product of all maximal prime powers r_i bounded by $O(\log \lambda)$, and then there exists $t = O(\log \lambda / \log\log \lambda)$.

Lemma 3.4.1 ([AP14])**.** *For all $x \geq 7$, the product of all maximal prime powers $r_i \leq x$ is at least* $\exp(3x/4)$.

By CRT, the additive group $\mathbb{Z}_q^+$ is isomorphic to the direct product $\mathbb{Z}_{r_1}^+ \times \cdots \times \mathbb{Z}_{r_t}^+$. For all $i \in [t]$, $x \in \mathbb{Z}_{r_i}^+$ corresponds to a cyclic permutation that can be represented by an indicator vector with 1 in the x-th position and 0 in the others. The reason is that we can compute permutation matrices (whose concrete definition is described in Section 3.3.2) for elements in $\mathbb{Z}_{r_i}$ from their indicator vectors as described in Section 3.1.3. We write $\phi_i : \mathbb{Z}_q \to \{0,1\}^r$, where $r := \max_i\{r_i\}$, for an embedding from $\mathbb{Z}_q$ to a group of cyclic permutations for the elements in $\mathbb{Z}_{r_i}$.

Our optimized bootstrapping procedure consists of two algorithms, $\mathsf{BootKeyGen}$ and $\mathsf{Bootstrap}$. The procedure can be used to refresh ciphertexts of all known standard LWE-based FHE. We achieve the input ciphertext $\boldsymbol{c} \in \{0,1\}^d$ for $\mathsf{Bootstrap}$ from the dimension-modulus reduction and bit-decomposition of the ciphertext to be refreshed, and let $\boldsymbol{s} \in \mathbb{Z}_q^d$ be a secret key that corresponds to $\boldsymbol{c}$. This pre-processing is the same as that in [AP14], so see for further details.

- $\mathsf{BootKeyGen}(\mathsf{sk}, \boldsymbol{s})$: Given a secret key sk for our matrix GSW-FHE and a secret key $\boldsymbol{s} \in \mathbb{Z}_q^d$ for a ciphertext to be refreshed, output a bootstrapping key. For every $i \in [t]$ and $j \in [d]$, let $\pi_{\phi_i(s_j)}$ be the permutation corresponding to $\phi_i(s_j)$, and compute

$$\tau_{i,j} \xleftarrow{R} \mathsf{SecEnc}_{\mathsf{sk}}(\mathsf{diag}(\phi_i(s_j))),$$
$$\mathsf{ssk}_{i,j} \xleftarrow{R} \mathsf{SwitchKeyGen}(\mathsf{sk}, \pi_{\phi_i(s_j)}),$$

where for a vector $\boldsymbol{x} \in \mathbb{Z}^r$, $\mathsf{diag}(\boldsymbol{x}) \in \mathbb{Z}^{r \times r}$ is the square integer matrix that has $\boldsymbol{x}$ in its diagonal entries and 0 in the others. In addition, we generate hints to check equality on packed indicator vectors. For every $i \in [t]$ and $x \in \mathbb{Z}_q$ such that $\lfloor x \rceil_2 = 1$ [2], generate

$$\mathsf{ssk}_{\phi_i(x)} \xleftarrow{R} \mathsf{SwitchKeyGen}(\mathsf{sk}, \pi_{\phi_i(x)}),$$

where $\pi_{\phi_i(x)}$ is the cyclic permutation that maps the $(x \bmod r_i)$-th row to the first row in the matrix. To mask the first plaintext slot, generate an encryption of $(1, 0, \ldots, 0)$:

$$\boldsymbol{P} \xleftarrow{R} \mathsf{SecEnc}_{\mathsf{sk}}(\mathsf{diag}((1, 0, \ldots, 0))).$$

Output the bootstrapping key

$$\mathsf{bk} := \{(\tau_{i,j}, \mathsf{ssk}_{i,j}, \boldsymbol{P}, \mathsf{ssk}_{\phi_i(x)})\}_{i \in [t], j \in [d], x \in \mathbb{Z}_q : \lfloor x \rceil_2 = 1}.$$

- $\mathsf{Bootstrap}_{\mathsf{bk}}(\boldsymbol{c})$: Given a bootstrapping key bk and a ciphertext $\boldsymbol{c} \in \mathbb{Z}_q^d$, output the refreshed ciphertext $\boldsymbol{C}^*$. The decryption of all FHE based on the standard LWE computes $\lfloor \langle \boldsymbol{c}, \boldsymbol{s} \rangle \rceil_2$. The algorithm $\mathsf{Bootstrap}$ consists of two phases that homomorphically evaluate the inner product and rounding.

  Inner Product: For every $i \in [t]$, homomorphically compute an encryption of $\phi_i(\langle \boldsymbol{c}, \boldsymbol{s} \rangle)$. Let $h := \min\{j \in [d] : c_j = 1\}$. For $i = 1, \ldots, t$, set $\boldsymbol{C}_i^* := \tau_{i,h}$, and iteratively compute

  $$\boldsymbol{C}_i^* \xleftarrow{R} \mathsf{SlotSwitch}_{ssk_{i,j}}(\boldsymbol{C}_i^*)$$

  for $j = h + 1, \ldots, d$ such that $c_j = 1$.

  Rounding: For each $x \in \mathbb{Z}_q$ such that $\lfloor x \rceil_2 = 1$, homomorphically check equality between x and $\langle \boldsymbol{c}, \boldsymbol{s} \rangle$, and sum their results. The refreshed ciphertext is computed as:

  $$\boldsymbol{C}^* \xleftarrow{R} \bigoplus_{x \in \mathbb{Z}_q : \lfloor x \rceil_2 = 1} \left(\bigodot_{i \in [t]} \left(\mathsf{SlotSwitch}_{\mathsf{ssk}_{\phi_i(x)}}(\boldsymbol{C}_i^*) \right) \odot \boldsymbol{P} \right). \tag{3.1}$$

The post-processing is almost the same as that in [AP14] except for the way to extract a matrix ciphertext. When finishing the bootstrapping procedure, we have a ciphertext $\boldsymbol{C}^*$ that encrypts in the first slot the same plaintext as the ciphertext

[2] Obviously, our procedure can work on not only the rounding function $\lfloor \cdot \rceil_2$ but also some arbitrary functions $f : \mathbb{Z}_q \to \{0, 1\}$.

$\boldsymbol{c}$. A vector ciphertext like [BV11a, BGV12, Bra12] can be obtained to just take the $\ell - 1$-th column vector of $\boldsymbol{C}^*$, and a matrix ciphertext like [GSW13] can be obtained by removing from the second row to the r-th row and from the $l + 1$-th column to rl-th column, and aggregating the remainders. We can utilize the key-switching procedure [BV11a, BGV12] for switching from $\boldsymbol{s}_1$ back to the original secret key $\boldsymbol{s}$. This requires us to assume circular security.

Our bootstrapping procedure is more time- and space- efficient than that of [AP14]. The procedure [AP14] encrypts every elements of the permutation matrices corresponding to the secret key elements, and homomorphically evaluates naive matrix multiplications to obtain encryptions of compositions of permutations. In our procedure, a permutation is encrypted in one ciphertext, and a composition is computed by two homomorphic multiplications. This makes our procedure time-efficient by roughly a $O(\log^2 \lambda)$ factor, and space-efficient by a $O(\log \lambda)$ factor.

3.4.2 Correctness and Security

From the security of our scheme, it is easy to see that our bootstrapping procedure can be secure by assuming the circular security and DLWE. Correctness holds as the following lemma.

Lemma 3.4.2. *Let* sk *be the secret key for our scheme. Let* $\boldsymbol{c}$ *and* $\boldsymbol{s}$ *be a ciphertext and secret key described in our bootstrapping procedure. Then, for a bootstrapping key* $\mathsf{bk} \xleftarrow{R} \mathsf{BootKeyGen}(\mathsf{sk}, \boldsymbol{s})$*, the refreshed ciphertext* $\boldsymbol{C}^* \xleftarrow{R} \mathsf{Bootstrap}_{\mathsf{bk}}(\boldsymbol{c})$ *encrypts* $\lfloor \langle \boldsymbol{s}, \boldsymbol{c} \rangle \rceil_2 \in \{0, 1\}$ *in the first slot.*

Proof. From Lemma 3.3.3 and group homomorphism of ϕ_i, $\boldsymbol{C}_i^*$ encrypts $\phi_i([\langle \boldsymbol{s}, \boldsymbol{c} \rangle]_q)$. Since $\mathbb{Z}_q$ is isomorphic to $\mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_t}$ by CRT, $\bigodot_{i \in [t]} (\mathsf{SlotSwitch}_{\mathsf{ssk}_{\phi_i(x)}}(\boldsymbol{C}_i^*)) \odot \boldsymbol{P}$ encrypts 1 in the first slot if and only if $x = \langle \boldsymbol{s}, \boldsymbol{c} \rangle \bmod q$. Finally, $\boldsymbol{C}^*$ encrypts 1 if and only if $\lfloor \langle \boldsymbol{s}, \boldsymbol{c} \rangle \rceil_2 = 1$. □

Here, we let s be the Gaussian parameter. Recall that n is the LWE dimension, r is the number of encrypted bits, $\ell = \lceil \log Q \rceil$, $N = (n + r) \cdot \ell$, $t = O(\log \lambda / \log \log \lambda)$, $d = \tilde{O}(\lambda)$ and $q = \tilde{O}(\lambda)$. We estimate the noise growth by our optimized bootstrapping procedure.

Lemma 3.4.3. *For any ciphertext* $\boldsymbol{c} \in \{0, 1\}^d$ *described in our bootstrapping procedure, the noise in the refreshed ciphertext* $\boldsymbol{C}^* \xleftarrow{R} \mathsf{Bootstrap}_{\mathsf{bk}}(\boldsymbol{c})$ *has independent subgaussian entries with parameter* $O(s\sqrt{n\ell dtq})$*, except with probability* $2^{-\Omega((n+r)ldt)}$ *over the random choice of* bk *and* $\mathsf{Bootstrap}$.

Proof. Since the parenthesized part before the additions in Eq. (3.1) can be broken down into a sequence of $O(dt)$ homomorphic multiplications, Corollary

3.3.1 and Lemma 3.2.1 tell us that the term has subgaussian noise with parameter $O(s\sqrt{Ndt})$, except with probability $2^{-\Omega(Ndt)}$. From the Pythagorean additivity of subgaussian random variables and $N = (n + r) \cdot \ell$, the noise in $\boldsymbol{C}^*$ are subgaussian with parameter $O(s\sqrt{(n + r)\ell dtq})$, and so $O(s\sqrt{n\ell dtq})$ by the fact $n > r$. □

From the above lemma, we can see that our procedure refreshes ciphertexts with error growth by the $O(\sqrt{nldtq})$ factor. Our scheme can evaluate its augmented decryption circuit by choosing a larger modulus than the final noise, and thus be pure FHE by the Gentry's bootstrapping theorem and the circular security assumption.

Theorem 3.4.1. *Our optimized bootstrapping scheme can be correct and secure assuming*

- *the quantum worst-case hardness of approximating* $\mathsf{GapSVP}_{\tilde{O}(n^{1.5}\lambda)}$ *and* $\mathsf{SIVP}_{\tilde{O}(n^{1.5}\lambda)}$,
- *or the classical worst-case hardness of approximating* $\mathsf{GapSVP}_{\tilde{O}(n^2\lambda)}$

on any n dimensional lattice.

Proof. By Lemma 3.2.1, to rely on the quantum worst-case hardness, we choose $s = \Theta(\sqrt{n})$. From Lemma 3.4.3, for correctness we only have to select $Q = \tilde{\Omega}(n\lambda \log Q)$, which satisfies $Q = \tilde{O}(n\lambda)$. Since the LWE inverse error rate is $1/\alpha = Q/s = \tilde{O}(\sqrt{n}\lambda)$, the security of our bootstrapping scheme is reduced to $\mathsf{GapSVP}_{\tilde{O}(n^{1.5}\lambda)}$ and $\mathsf{SIVP}_{\tilde{O}(n^{1.5}\lambda)}$.

In the case of reducing to the classical hardness of the lattice problem, since $1/\alpha = \tilde{\Omega}(\lambda\sqrt{n \log Q})$ and we must take $Q \approx 2^{n/2}$, the LWE inverse error rate satisfies $1/\alpha = \tilde{\Omega}(\lambda \cdot n)$. Therefore, the security of our optimized bootstrapping scheme is reduced to the classical hardness of $\mathsf{GapSVP}_{\tilde{O}(n^2\lambda)}$. □

Since all known algorithms that approximate GapSVP and SIVP on any n dimensional lattices to within a poly(n)-factor run in time $2^{\Omega(n)}$, the 2^λ hardness requires us to choose $n = \Theta(\lambda)$. This makes the problems to which the security is reduced in the quantum case have the approximation factor $\tilde{O}(n^{2.5})$, which is smaller than $\tilde{O}(n^3)$, the one of [AP14]'s bootstrapping scheme. In the classical case, the LWE inverse error rate is $1/\alpha = \tilde{\Omega}(n^2)$ and hence our approximation factor is $\tilde{O}(n^3)$. Furthermore, by selecting a larger dimension $n = \lambda^{1/\epsilon}$ for $\epsilon > 0$ (so at the cost of efficiency), the approximation factor can be $\tilde{O}(n^{1.5+\epsilon})$, which is comparable to the one of [BV14] and so the best known factor of standard lattice-based PKE. Consequently, our optimized bootstrapping scheme can be as secure as any other standard lattice-based PKE *without* successive dimension-modulus reduction, which is essential in all the known bootstrapping procedures [BV14, AP14] provided recently.

3.5 Conclusion of This Chapter

In this chapter, we showed the first construction of FHE that encrypts matrices and supports homomorphic matrix operations. This is a natural extension of packed FHE and supports more complicated homomorphic operations. We then showed that our FHE can be used to optimize the bootstrapping procedure proposed by Alperin-Sheriff and Peikert [AP14].

New SIMD FHE based on the LWE assumption. The proposed FHE is secure under the LWE assumption, and the homomorphic SIMD operation algorithms are very simple: they are just addition and multiplication of two ciphertext matrices. The simplicity of the algorithms leads smaller complexity than the previous SIMD FHE based on the same security assumption. In particular, our homomorphic SIMD multiplication requires $\tilde{O}(n^{2.3727})$ operations while the previous SIMD FHE requires $\tilde{O}(n^3)$ operations. The complexity of our FHE is estimated by the best complexity of the multiplication algorithm between two square matrices. As the study for the matrix multiplication goes forward, the time complexity of our homomorphic evaluations will decrease.

Our FHE is an extension of [GSW13], and the construction of [GSW13] has a great influence on the construction idea of some cryptosystems based on the LWE such as fully homomorphic signatures [GVW15, FMNP16], attribute based encryption [BGG+14, BV16, BCTW16], and multilinear maps [GGH15]. Therefore, our construction idea may also provide a big impact on future cryptographic constructions based on LWE.

Optimizing Bootstrapping of [AP14]. Using our scheme, we fully sequentialized and thus optimized the bootstrapping procedure of [AP14]. We recall the result of the optimization described in the following theorem:

Theorem 3.5.1. *Our optimized bootstrapping scheme can be secure assuming the hardness of approximating the standard lattice problem to within the factor* $\tilde{O}(n^{1.5}\lambda)$ *on any n dimensional lattices.*

For 2^λ hardness, we need to take $n = \Omega(\lambda)$. Asymptotically minimal selection of $n = \tilde{O}(\lambda)$ leads to the approximation factor $\tilde{O}(n^{2.5})$ for the underlying worst-case lattice assumption, which is smaller than $\tilde{O}(n^3)$, the factor of [AP14]. Using a kind of dimension leveraging technique: selecting a larger dimension $n = \lambda^{1/\epsilon}$ for $\epsilon \in (0, 1)$, we can also obtain the best known approximation factor, $\tilde{O}(n^{1.5+\epsilon})$, without successive dimension-modulus reduction, which was essential for achieving the best factor in the prior works on bootstrapping of standard lattice-based FHE.

Relation with the Replacement of the Random Oracle. The construction of

our FHE is so similar to a recent multilinear maps construction of Gorbunov et al [GGH15]. Multilinear maps [GGH13a, CLT13, GGH15] are very powerful cryptographic tools that allow us to obtain highly functional cryptographic primitives such as indistinguishability obfuscation [GGH+13b]. An example application of the indisitinguishability obfuscation is to replace the random oracle in a cryptographic scheme to a concrete hash function [HSW13, FHPS13, HSW14]. From the next chapter, we use the indistinguishability obfuscation to show that the random oracle of our cryptosystems can be directly replaced by a real hash function derived from the indistinguishability obfuscation. This guarantees that our cryptosystems in the following chapters can be at least implemented securely.

Chapter 4

Tightly Secure Signatures from the RSA Assumption

In this chapter, we construct a new RSA-based signature scheme that is tightly secure in the random oracle model. The number of random oracles used in this scheme is less than that of all previous schemes with the same security guarantee. We then show that for any PPT adversary there exists a concrete hash function from indistinguishability obfuscation that can replace the random oracle while maintaining security. The same statement can be proven for the signatures of Coron.

4.1 Efficient Digital Signatures

4.1.1 Background

The security of a cryptosystem is guaranteed by a certain computational hardness assumption. To prove the security of the cryptosystem, we reduce breaking the security to break the assumption (i.e., to solve the problem assumed to be hard in the underlying security assumption). There is a gap, which is called *reduction efficiency*, between the hardnesses of breaking a cryptosystem and solving a security problem. The reduction efficiency is defined as the probability that breaking security of a cryptosystem leads solving a problem on which the security of the cryptosystem is based. We say that a reduction from the security of a cryptosystem to an underlying problem is tight if its reduction efficiency is equal to 1 (i.e., if we can break the cryptosystem, we can solve the underlying problem with probability 1). If a security reduction is tight, breaking the cryptosystem is as hard as solving the underlying problem. Hence, if we can prove the security of a cryptosystem by constructing a tight security reduction, we can see that the cryptosystem can be

implemented with smaller parameter settings (which leads to smaller key size). In this chapter, we particularly focus on tightly secure digital signatures based on the RSA assumption in the random and standard models.

The random oracle model is an idealized paradigm in which hash functions are viewed as an oracle that outputs a random value for every input query. Bellare and Rogaway in [BR96] proposed full domain hash (FDH) signatures that are implemented by the random oracle in the security proof. The reduction efficiency of the FDH signatures was improved by Coron in [Cor02]. Bellare and Rogaway [BR96] also proposed a Probabilistic Signature Scheme (PSS) whose security is tightly reducible to the RSA assumption. Since the PSS is tightly secure only for long random salts, Coron introduced a probabilistic full-domain hash (pFDH) implemented by the random oracle to prove that the PSS also has a tight security reduction for shorter random salts. However, the Coron's signature scheme has a complex construction since it uses the random oracle multiple times. The above signatures are secure in the random oracle model, and their random oracles are replaced by concrete hash functions when implementing their signatures in the real world. A security proof for a cryptographic scheme in the random oracle model does not mean that it is secure in the real world, but it provides some kind of security guarantee, and it is still important in a practical sense to prove the security in the random oracle model.

There are some impossibility results [CGH98, GK03, BBP04] that show that the random oracle of their results cannot be replaced by any concrete hash function. They not only construct *artificial* (namely, unnatural) usages of the random oracle, but also show that there exists an impossibility result in the Fiat-Shamir heuristic, which is widely used in the real world. In such ways in using the random oracle, it may be difficult to show practical importance of the security proofs in the random oracle model. Therefore, a main issue to consider is how to use the random oracle to indicate the practical importance. That is, we should tackle a problem to construct a way to use the random oracle so that it does not lead to impossibility results like the above. In this line of research, recently, there have been some recent studies [HSW13, FHPS13, HSW14] that investigate cases in which the random oracle is directly replaced with a concrete hash function while maintaining security. The goal of these studies is to obtain an understanding or findings regarding the random oracle in the underlying cryptosystems. [1] In particular, it is an important problem to see whether or not a cryptosystem from the random oracle remains secure even if the random oracle is replaced by a concrete hash function. So, we address the following problem:

[1] A goal of this chapter is to show that the random oracle used in our constructions can be replaced by a concrete hash function constructed from indistinguishability obfuscation. It differs from the work by Sahai and Waters [SW14] that shows which cryptographic components can be constructed from indistinguishability obfuscation.

In a digital signature scheme tightly secure based on the RSA assumption in the random oracle model, Can its random oracle be replaced with a concrete hash function while maintaining the same security?

4.1.2 Our Results

In this chapter, we first propose new digital signatures that are tightly secure based on the RSA assumption in the random oracle model. Our signatures have a simpler construction than the previous tightly secure RSA-based signatures [BR96, Cor02], since the number of the random oracle used in our signatures is less than the previous ones. While the Coron's signatures [Cor02] has the security reduction whose efficiency depends on the number of adversarial queries to the signing oracle, the efficiency of the security reduction for our signatures does not depend on the power of the adversary. In other words, while the security of the Coron's signatures depends on the complexity of the adversary, the security of our signatures does not. Additionally, we introduce a new proof technique, called α-β *hiding technique*, which has not been used in security proofs of the previous cryptosystems secure in the random oracle model.

Next, we answer the question mentioned in the above: we show that the random oracle of our signatures can be directly replaced by a concrete hash function from indistinguishability obfuscation while maintaining the same security [2]. We also show that the random oracle of the Coron's signatures [Cor02] can be directly replaced by a concrete hash function. Therefore, we consider that these results give some findings for the use of the random oracle in the Coron's signatures [Cor02].

4.1.3 Our Techniques

Tightly Secure Digital Signatures based on the RSA Assumption in the Random Oracle Model. Let (N, e) be a public key and d be a secret key of RSA encryption [3]. Let h be a collision-resistant hash function, and H be a hash function implemented by the random oracle. Let v_0 and v_1 be two random integers

[2] Indistinguishability obfuscator is thought to be constructed in the standard model (at least there are no negative results to say that it cannot be constructed in the standard model), but it is achieved under a very strong computational assumption, and so it is desirable to show that the indistinguishability obfuscator can be constructed from the weaker assumption. This thesis proves that the random oracle of our signatures is removable under the existence of indistinguishability obfuscation.

[3] For the mathematics of the RSA assumption, we refer the reader to Definition 4.2.7.

chosen from $\mathbb{Z}_N^*$. For random $r \in \{0,1\}^\gamma$ (γ is a polynomial for a security parameter) and $s \in \mathbb{Z}_e$, a signature for plaintext m is a triple (σ, r, s) where σ is computed as

$$\sigma := (v_0 v_1^{h(m)} H(r)^s)^d \bmod N.$$

To verify this signature, we check whether or not the following holds:

$$\sigma^e \equiv v_0 v_1^{h(m)} H(r)^s \pmod{N}.$$

To prove the security of the above signatures, we introduce a new proof technique, which is called *α-β hiding technique*, to simulate responses to oracle queries. Suppose that the reduction algorithm to the RSA assumption receives an instance (N, e, y) of the RSA problem. The reduction algorithm first computes $v_0 := y^{\beta_0} \bmod N$ and $v_1 := y^{\beta_1} \bmod N$ for $\beta_0, \beta_1 \xleftarrow{U} \{0,1\}^{3|N|}$ ($\beta_0, \beta_1 \not\equiv 0 \pmod{e}$), where v_0 and v_1 are uniformly distributed over $\mathbb{Z}_N^*$. To simulate the random oracle, the reduction first chooses random $\alpha_i \xleftarrow{U} \{0,1\}^{3|N|}$ ($\alpha_i \not\equiv 0 \pmod{e}$), stores triple $(r_i, y^{\alpha_i} \bmod N, \alpha_i)$ for random oracle query $r_i \in \{0,1\}^\gamma$, and defines the random oracle output as $H(r_i) := y^{\alpha_i} \bmod N$, where $H(r_i)$ is also uniformly distributed over $\mathbb{Z}_N^*$ as well as v_0 and v_1. To simulate a response for signing oracle query m_j, the reduction chooses $\alpha_j \xleftarrow{U} \{0,1\}^{3|N|}$ and computes $s_j := (-\beta_0 - h(m_j)\beta_1)/\alpha_j \bmod e$, where for every s_j β_0 (and β_1) is hidden by α_j. From $e \mid \beta_0 + h(m_j)\beta_1 + s_j\alpha_j$, the reduction can compute the corresponding signature

$$\sigma_j := y^{(\beta_0 + h(m_j)\beta_1 + s_j\alpha_j)/e} \bmod N.$$

The reduction chooses $r_j \xleftarrow{U} \{0,1\}^\gamma$, stores $(r_j, y^{\alpha_j} \bmod N, \alpha_j)$ to a list for the random oracle, and defines $H(r_j) := y^{\alpha_j} \bmod N$. The simulated signature (σ_j, r_j, s_j) for m_j satisfies

$$\sigma_j^e \equiv v_0 v_1^{h(m_j)} H(r_j)^{s_j} \pmod{N}.$$

The reduction that simulates the random and signing oracles as above can break the RSA assumption by using a successful forger. For forgery $(m^*, \sigma^*, r^*, s^*)$ and integer α^* stored as $(r^*, y^{\alpha^*} \bmod N, \alpha^*)$ in the list for the random oracle, $e \nmid \beta_0 + h(m^*)\beta_1 + \alpha^* s^*$ holds with overwhelming probability. Then it holds that $(\sigma^*)^e \equiv y^{K_0 e + L_0} \pmod{N}$ for some $K_0, L_0 \neq 0 \in \mathbb{Z}$. Since this leads $(\sigma^* y^{-K_0})^e \equiv y^{L_0} \pmod{N}$ and we have $\gcd(e, L_0) = 1$, the reduction can efficiently compute $x \in \mathbb{Z}$ such that $x^e \equiv y \pmod{N}$.

Replacing the Random Oracle with a Concrete Hash Function. We here briefly show that the random oracle H of the above signatures can be directly replaced by a concrete hash function constructed from indistinguishability obfuscation and punctured pseudorandom functions. [4]

[4] For indistinguishability obfuscation and punctured pseudorandom functions, we refer the reader to Section 4.2.3.

The reduction algorithm first chooses a key K for a punctured pseudorandom function $F : \{0,1\}^\gamma \to \{0,1\}^{3|N|}$, chooses $y \xleftarrow{U} \mathbb{Z}_N^*$, and defines a new pseudorandom function F_0 as

$$F_0(K; r) := y^{F(K;r)} \bmod N.$$

The reduction publishes as hash function H an obfuscation of the program that outputs $F_0(K; r)$ for input $r \in \{0,1\}^\gamma$.

To prove the security, we use a sequence of games. In the following, we let q be the number of adversarial signing queries. The first game is the EUF-CMA game. In the second game, the challenger sets q punctures in the domain of the punctured pseudorandom function. This change does not vary the input and output of F_0. In the third game, a challenger sets as the output of F_0 almost uniformly random values $y^{\alpha_j} \bmod N$ ($\alpha_j \xleftarrow{U} \{0,1\}^{3|N|}, \alpha_j \not\equiv 0 \pmod e$). In the last game, the challenger simulates the signing oracle by setting $v_0 := y^{\beta_0} \bmod N, v_1 := y^{\beta_1} \bmod N$ ($\beta_0, \beta_1 \xleftarrow{U} \{0,1\}^{3|N|}, \beta_0, \beta_1 \not\equiv 0 \pmod e$), and computes $s_j := (-\beta_0 - \beta_1 h(m_j))/\alpha_j \bmod e$ and $\sigma_j := y^{(\beta_0+\beta_1 h(m_j)+\alpha_j s_j)/e} \bmod N$ in responses of the signing oracle queries. In this last game, the reduction algorithm can break the RSA assumption as well as in the random oracle model by using a successful forger against our signature scheme.

Replacing the Random Oracle of the Coron's Signature [Cor02] In [Cor02], Coron used a pFDH to construct tightly secure signatures under the RSA assumption. The Coron's scheme generates a signature for m as a triple (σ, r, s) where r and s are random bit strings and σ is computed as

$$\sigma := H(m, r, s)^d \bmod N.$$

We can also show that the pFDH H implemented by the random oracle can also be replaced by a concrete hash function. In particular, the hash function H is constructed as follows: takes as input m, r, s, and outputs $v_0 v_1^{h(m)} (F_0(K; r))^s \bmod N$, where $v_0, v_1 \xleftarrow{U} \mathbb{Z}_N^*$ are uniformly random elements in $\mathbb{Z}_N^*$, h is a collision-resistant hash function, and F_0 is a punctured pseudorandom function defined as $F_0(K; r) := y^{F(K;r)} \bmod N$ for some punctured pseudorandom function $F : \{0,1\}^\gamma \to \{0,1\}^{3|N|}$ and random integer $y \xleftarrow{U} \mathbb{Z}_N^*$. The security of this scheme can be proven as well as the above case when instantiating the random oracle of our signatures.

4.1.4 Organization of This Chapter

In Section 4.2, we introduce mathematical preliminaries for this chapter. In Section 4.3, we propose a new efficient digital signature scheme whose security is tightly reducible to the RSA assumption in the random oracle model. In Section

4.4, we prove that for any PPT adversary, there exists a concrete hash function that is constructed from indistinguishability obfuscation, so that it can directly replace the random oracle of the scheme in Section 4.3. In Section 4.5, we adapt the technique of Section 4.4 to replace the pFDH of the Coron's signatures. In Section 4.6, we summarize the results achieved in this chapter.

4.2 Preliminaries

4.2.1 Digital Signatures

We here introduce the definition of digital signatures and their security.

Definition 4.2.1 (Digital Signatures)**.** *A digital signature scheme is a triple of three algorithms* $\Sigma := (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify})$ *that satisfy the following:*

- $\mathsf{Setup}(1^\lambda)$*: takes as input a security parameter* λ*, and outputs a pair of verification and signing keys* $(\mathsf{vk}, \mathsf{sk})$*.*
- $\mathsf{Sign}_{\mathsf{sk}}(m)$*: takes as input a signing key* sk *and message* m*, and outputs a signature* σ *for* m*.*
- $\mathsf{Verify}_{\mathsf{vk}}(m, \sigma)$*: takes as input a verification key* vk*, message* m*, and signature* σ*, and outputs* 1 *if* σ *is a valid signature for* m*, and* 0 *otherwise.*

A digital signature scheme Σ *is correct if the following holds:*

$$\Pr\left[\mathsf{Verify}_{\mathsf{vk}}(m, \sigma) = 1 : \begin{array}{l}(\mathsf{vk}, \mathsf{sk}) \xleftarrow{R} \mathsf{Setup}(1^\lambda); \\ m \xleftarrow{U} \mathcal{M}; \sigma \xleftarrow{R} \mathsf{Sign}_{\mathsf{sk}}(m)\end{array}\right] = 1,$$

where $\mathcal{M}$ *is the message space defined by* vk*.*

To define the security of digital signatures, we consider the following experiment (game) between a challenger and adversary. Let $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\Sigma,\mathcal{F}}(\lambda)$ be an experiment for a digital signature scheme Σ between a challenger and adversary $\mathcal{F}$.

- **Setup phase**: The challenger receives a security parameter λ, generates a pair of verification and signing keys $(\mathsf{vk}, \mathsf{sk}) \xleftarrow{R} \mathsf{Setup}(1^\lambda)$, and sends vk to the adversary.
- **Query phase**: The adversary can obtain a signature σ_j for m_j by sending a signing query m_j $(1 \leq j \leq q)$ to the challenger.
- **Guess phase**: The adversary outputs a signature forgery (m^*, σ^*). The challenger outputs 1 if $m^* \notin \{m_j\}_{j\in[q]}$ and $\mathsf{Verify}_{\mathsf{vk}}(m^*, \sigma^*) = 1$, and 0 otherwise.

The advantage of the adversary $\mathcal{F}$, $\mathsf{Adv}^{\text{EUF-CMA}}_{\Sigma,\mathcal{F}}(\cdot)$, is defined to be the probability that the above experiment outputs 1. That is,

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\Sigma,\mathcal{F}}(\lambda) := \Pr[\mathsf{Exp}^{\text{EUF-CMA}}_{\Sigma,\mathcal{F}}(\lambda) \to 1].$$

We use the advantage of $\mathcal{F}$ to define the security of the digital signatures:

Definition 4.2.2. *For any PPT algorithm $\mathcal{F}$, a digital signature scheme Σ is EUF-CMA secure if the following holds:*

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\Sigma,\mathcal{F}}(\lambda) = \mathsf{negl}(\lambda).$$

4.2.2 Collision-Resistant Hash Function

In this section, we introduce the definition of collision-resistant hash functions.

Definition 4.2.3 (Hash Function Family)**.** *A function family $\{h_i : \mathcal{D}_i \to \mathcal{R}_i\}_{i\in\mathcal{I}}$ is a hash function family if a tuple of PPT algorithms* $(\mathsf{Gen}, \mathsf{Samp}, h_i)$ *satisfies the following:*

- $\mathsf{Gen}(1^\lambda)$*: takes as input a security parameter λ, and outputs an index $i \in \mathcal{I}$ such that $|i| \geq \lambda$. The index $i \in \mathcal{I}$ determines a domain $\mathcal{D}_i$ with probability distribution and a range $\mathcal{R}_i$. Then it holds that $|\mathcal{D}_i| \geq |\mathcal{R}_i|$.*
- $\mathsf{Samp}(i)$*: takes as input $i \in \mathcal{I}$, and outputs a sample from $\mathcal{D}_i$.*
- $h_i(x)$*: takes as input $i \in \mathcal{I}$ and $x \in \mathcal{D}_i$, and outputs $h_i(x) \in \mathcal{R}_i$.*

Definition 4.2.4 (Collision-Resistant Hash Function)**.** *A hash function family $\{h_i : \mathcal{D}_i \to \mathcal{R}_i\}_{i\in\mathcal{I}}$ is collision-resistant if for any PPT algorithm $\mathcal{A}$ it holds that*

$$\begin{aligned}\mathsf{Adv}^{\text{CRHF}}_{\mathcal{A}}(\lambda) &:= \Pr\left[\begin{array}{l|l}(x, x^*) \in \mathcal{D}_i^2, x \neq x^*, & i \xleftarrow{R} \mathsf{Gen}(1^\lambda), \\ h_i(x) = h_i(x^*) & (x, x^*) \xleftarrow{R} \mathcal{A}(1^\lambda, i)\end{array}\right] \\ &= \mathsf{negl}(\lambda).\end{aligned}$$

4.2.3 Indistinguishability Obfuscation, Punctured Pseudorandom Function

The concept of indistinguishability obfuscation was first introduced by Barak et al. in [BGI+01]. Intuitively, an obfuscator is indistinguishability obfuscation if for all inputs x and two circuits C_1, C_2 with the same size, the obfuscations of C_1 and C_2 are computationally indistinguishable. Garg et al. [GGH+13b] showed that a candidate of the indistinguishability obfuscator can be constructed in the standard model. Formally, indistinguishability obfuscator is defined as follows.

Definition 4.2.5 (Indistinguishability Obfuscator)**.** *A uniform PPT algorithm $i\mathcal{O}$ is an indistinguishability obfuscator for a circuit class $\{C_\lambda\}_{\lambda\in\mathbb{N}}$ if the following holds:*

- *For all $\lambda \in \mathbb{N}$, all $C \in C_\lambda$, all inputs x,*

$$\Pr[C'(x) = C(x) : C' \xleftarrow{R} i\mathcal{O}(\lambda, C)] = 1.$$

- *For any (not necessarily uniform) PPT adversary $(\mathsf{Samp}, \mathcal{D})$, there exists a negligible function $\alpha(\cdot)$ such that the following holds: if $\Pr[C_0(x) = C_1(x)\ \forall x : (C_0, C_1, \tau) \xleftarrow{R} \mathsf{Samp}(1^\lambda)] \geq 1 - \alpha(\lambda)$, then we have*

$$\mathsf{Adv}^{i\mathcal{O}}_{\mathsf{Samp},\mathcal{D}}(\lambda) := \left| \Pr\left[\begin{array}{c} \mathcal{D}(i\mathcal{O}(\lambda, C_0), \tau)=1: \\ (C_0, C_1, \tau) \xleftarrow{R} \mathsf{Samp}(1^\lambda) \end{array} \right] - \Pr\left[\begin{array}{c} \mathcal{D}(i\mathcal{O}(\lambda, C_1), \tau)=1: \\ (C_0, C_1, \tau) \xleftarrow{R} \mathsf{Samp}(1^\lambda) \end{array} \right] \right| = \mathsf{negl}(\lambda).$$

Puncturable pseudorandom functions are pseudorandom functions (PRFs) that can be defined on all bit strings of a certain length, except for any polynomial-size set of inputs. A PRF construction from the pseudorandom generator by Goldreich, Goldwasser and Micali [GGM84] leads a puncturable pseudorandom function that satisfies the following.

Definition 4.2.6 (Puncturable Pseudorandom Functions)**.** *A puncturable family of PRFs F is given by a triple of PPT algorithms $(\mathsf{Key}_F, \mathsf{Puncture}_F, \mathsf{Eval}_F)$ and a pair of computable functions $(n(\cdot), m(\cdot))$, that satisfy the following conditions:*

- ***Functionality preserved under puncturing.*** *For every PPT adversary $(\mathcal{A}_1, \mathcal{A}_2)$ such that $\mathcal{A}_1$ outputs a set $S \subseteq \{0,1\}^{n(\lambda)}$ and all $x \in \{0,1\}^{n(\lambda)} \setminus S$, we have that:*

$$\Pr\left[\begin{array}{l} \mathsf{Eval}_F(K, x) \\ \quad = \mathsf{Eval}_F(K_S, x) \end{array} : \begin{array}{l} K \xleftarrow{R} \mathsf{Key}_F(1^\lambda); \\ K_S = \mathsf{Puncture}_F(K, S) \end{array} \right] = 1.$$

- ***Pseudorandom at punctured points.*** *For every PPT adversary $(\mathcal{A}_1, \mathcal{A}_2)$ such that $\mathcal{A}_1$ outputs a set $S \subseteq \{0,1\}^{n(\lambda)}$ and state τ, consider an experiment where $K \xleftarrow{R} \mathsf{Key}_F(1^\lambda)$ and $K_S = \mathsf{Puncture}_F(K, S)$. Then we have*

$$\mathsf{Adv}^{\mathsf{pPRF}}_{\mathcal{A}_1,\mathcal{A}_2}(\lambda) := \left| \Pr[\mathcal{A}_2(\tau, K_S, S, \mathsf{Eval}_F(K, S))=1] - \Pr[\mathcal{A}_2(\tau, K_S, S, U(\{0,1\}^{m(\lambda)\cdot|S|}))=1] \right| = \mathsf{negl}(\lambda),$$

where $S = \{x_1, \ldots, x_k\}$ is the enumeration of the elements of S in lexicographic order, $\mathsf{Eval}_F(K, S)$ denotes the concatenation of $\mathsf{Eval}_F(K, x_1), \ldots, \mathsf{Eval}_F(K, x_k)$.

For the sake of ease, we write $\mathsf{Eval}_F(K, x)$ by $F(K; x)$

4.2.4 Hardness Assumptions

Here, we give the definition of the RSA assumption [RSA78].

Definition 4.2.7 (RSA [RSA78])**.** *Let* GenRSA *be an algorithm that takes as input security parameter* λ *and outputs* (n, p, q, e)*, where* $n = pq$*,* p *and* q *are primes, and* e *is a random integer such that* $\gcd(e, \phi(n)) = 1$ [5]*. The RSA problem is to compute* $x \in \mathbb{Z}_n$ *such that* $x^e \equiv y \pmod{n}$*, given* (n, e, y) $(y \xleftarrow{U} \mathbb{Z}_n^*)$*. The advantage of adversary* $\mathcal{A}$ *for the RSA problem is defined as*

$$\mathsf{Adv}^{\mathsf{RSA}}_{\mathcal{A}}(\lambda) := \Pr\left[x^e \equiv y \pmod{n} : \begin{array}{r}(n, p, q, e) \xleftarrow{R} \mathsf{GenRSA}(1^\lambda); \\ y \xleftarrow{U} \mathbb{Z}_n^*; x \xleftarrow{R} \mathcal{A}(n, e, y)\end{array}\right].$$

The RSA assumption holds if for any PPT adversary $\mathcal{A}$*, it holds that* $\mathsf{Adv}^{\mathsf{RSA}}_{\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$*.*

4.3 Digital Signature Scheme Σ_{ROM} in the Random Oracle Model

In this chapter, we construct a digital signature scheme whose security is tightly reducible to the RSA assumption. Our proposed scheme has a simpler construction than the previous schemes [BR96, Cor02]. In particular, our scheme can generate signatures via only one random oracle, while the previous PSSs such as [BR96, Cor02] use two random oracles to generate signatures. We show in Section 4.3.1 how to construct our signature scheme whose security is proven in Section 4.3.2.

4.3.1 Construction

Our proposed signature scheme consists of the following three PPT algorithms $\Sigma_{ROM} := (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify})$:

- $\mathsf{Setup}(1^\lambda)$: Generate an instance $(N, P, Q, e) \xleftarrow{R} \mathsf{GenRSA}(1^\lambda)$, where e is a prime number such that $|e| = |N|$ and $\gcd(e, \phi(N)) = 1$. Compute an integer $d \in \mathbb{Z}$ such that $ed \equiv 1 \pmod{\phi(N)}$. Let $\gamma := \gamma(\lambda)$ be a polynomial in λ. Let $H : \{0,1\}^\gamma \to \mathbb{Z}_N^*$ be a hash function modeled as the random oracle, and $h : \{0,1\}^* \to \mathbb{Z}_e$ be a collision-resistant hash function parameterized by security parameter λ. Choose two integers $v_0, v_1 \xleftarrow{U} \mathbb{Z}_N^*$ uniformly at random. Output the verification key $\mathsf{vk} := (H, h, v_0, v_1, N, e)$ and signing key $\mathsf{sk} := d$.

[5] When we use GenRSA after Section 4.3, e is a prime number where $|e| = |n|$.

- $\mathsf{Sign}_{\mathsf{sk}}(m \in \{0,1\}^*)$: Choose a random string $r \xleftarrow{U} \{0,1\}^\gamma$ and integer $s \xleftarrow{U} \mathbb{Z}_e$, and compute
$$\sigma := (v_0 v_1^{h(m)} H(r)^s)^d \bmod N.$$
Output (σ, r, s) as a signature for m.

- $\mathsf{Verify}_{\mathsf{vk}}(m, (\sigma, r, s))$: Output 1 if the following condition holds, and 0 otherwise:
$$\sigma^e \equiv v_0 v_1^{h(m)} H(r)^s \pmod N.$$

The correctness of Σ_{ROM} can immediately be proven from the equation in Verify.

Theorem 4.3.1 (Correctness)**.** *The signature scheme Σ_{ROM} is correct.*

Proof. A signature of a message m is a triple (σ, r, s) for randomness $r \in \{0,1\}^\gamma$ and $s \in \mathbb{Z}_e$ such that
$$\sigma := (v_0 v_1^{h(m)} H(r)^s)^d \bmod N,$$
where v_0, v_1 and N are the public key elements and d is the signing key. Therefore we have
$$\begin{aligned}\sigma^e &\equiv ((v_0 v_1^{h(m)} H(r)^s)^d)^e \\ &\equiv v_0 v_1^{h(m)} H(r)^s \bmod N \ (\because e \cdot d \equiv 1 \bmod \phi(N)).\end{aligned}$$
□

4.3.2 Security

In the following, we prove that our proposed signature scheme is EUF-CMA secure under the RSA assumption in the random oracle model. We can also immediately prove that our scheme is also sEUF-CMA secure.

Theorem 4.3.2. *If the RSA assumption holds and h is a collision-resistant hash function, then the proposed signature scheme Σ_{ROM} is EUF-CMA secure in the random oracle model. In particular, for any PPT adversary $\mathcal{F}$, there exists a PPT algorithm $\mathcal{B}$ such that*
$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Sigma_{ROM},\mathcal{F}}(\lambda) \leq \mathsf{Adv}^{\mathsf{RSA}}_{\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{F}}(\lambda) + \frac{1}{\Theta(2^\lambda)} + \frac{1}{2^\lambda}.$$

We use the sequence of the following games to prove Theorem 4.3.2. For a security parameter $\lambda \in \mathbb{N}$, let $\mathsf{Adv}^{\mathsf{Game}_i}_{\mathcal{F}}(\lambda)$ be an advantage of $\mathcal{F}$ in Game_i $(i = 0, 1, 2)$, respectively.

- Game_0: This is the same as the original EUF-CMA game, so we have $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Sigma_{ROM},\mathcal{F}}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{F}}(\lambda)$.

- Game_1: This is the same as Game_0 except for how to compute the values v_0 and v_1 in the verification key vk and how to reply random and signing oracle queries.

  The challenger of Game_1 first chooses a random integer $y \xleftarrow{U} \mathbb{Z}_N^*$. Instead of choosing v_0 and v_1 uniformly at random, the challenger computes $v_0 := y^{\beta_0} \bmod N, v_1 := y^{\beta_1} \bmod N$ for uniformly random $\beta_0, \beta_1 \xleftarrow{U} \{0,1\}^{3|N|}$ ($\beta_0, \beta_1 \not\equiv 0 \pmod{e}$).

  When the adversary $\mathcal{F}$ queries r_i to the random oracle, the challenger returns $H(r_i)$ computed as follows. Suppose that the challenger manages a list of tuples (r_i, c_i, α_i), which is called the H-List. The challenger returns c_i to $\mathcal{F}$ if the H-List already contains a tuple (r_i, c_i, α_i) for query r_i, and registers $(r_i, c_i := y^{\alpha_i} \bmod N, \alpha_i)$ (for $\alpha_i \xleftarrow{U} \{0,1\}^{3|N|}$ and $\alpha_i \not\equiv 0 \pmod{e}$) to the H-List and returns $H(r_i) := c_i$ to $\mathcal{F}$ otherwise.

  When $\mathcal{F}$ queries m_j to the signing oracle, the challenger simulates the oracle by computing a signature (σ_j, r_j, s_j) as follows. The challenger first chooses $\alpha_j \xleftarrow{U} \{0,1\}^{3|N|}$ ($\alpha_j \not\equiv 0 \pmod{e}$) uniformly at random, and then computes $s_j \in \mathbb{Z}_e$ such that $e \mid \beta_0 + h(m_j)\beta_1 + s_j\alpha_j$, that is $s_j := (-\beta_0 - h(m_j)\beta_1)/\alpha_j \bmod e$. The challenger computes $\sigma_j := y^{K_j} \bmod N$ for $K_j := (\beta_0 + h(m_j)\beta_1 + s_j\alpha_j)/e$, chooses a random string $r_j \xleftarrow{U} \{0,1\}^\gamma$, and registers a tuple $(r_j, c_j := y^{\alpha_j} \bmod N, \alpha_j)$ to the H-List. The challenger returns (σ_j, r_j, s_j) to $\mathcal{F}$ as a signature for m_j. Note that (σ_j, r_j, s_j) is a valid signature for m_j since $\sigma_j^e \equiv y^{\beta_0 + h(m_j)\beta_1 + s_j\alpha_j} \equiv v_0 v_1^{h(m_j)} H(r_j)^{s_j} \pmod{N}$.

- Game_2: This is the same as Game_1 except for aborting the game when $\mathcal{F}$ outputs a forgery $(m^*, (\sigma^*, r^*, s^*))$ such that $m^* \notin \{m_j\}_{j\in[q]}$ and $h(m^*) \in \{h(m_j)\}_{j\in[q]}$.

Theorem 4.3.2 can be proven from the following Lemmas 4.3.1, 4.3.2, and 4.3.3.

Lemma 4.3.1.

$$|\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_0}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda)| \leq \frac{1}{2^\lambda}.$$

Proof. The values v_0 and v_1 in Game_1 are statistically indistinguishable from those values in Game_0. Let $K, L(\neq 0) \in \mathbb{Z}$ be integers such that $2^{3|N|} = K\phi(N) + L$. The statistical distance between the distribution $U(\{0,1\}^{3|N|}) \bmod \phi(N)$ obtained by choosing an element from $\{0,1\}^{3|N|}$ uniformly at random and reducing it mod-

ulo $\phi(N)$, and the uniform distribution $U(\mathbb{Z}_{\phi(N)})$ over $\mathbb{Z}_{\phi(N)}$ satisfies

$$\begin{aligned}
&\Delta(U(\{0,1\}^{3|N|}) \bmod \phi(N), U(\mathbb{Z}_{\phi(N)})) \\
=&\frac{1}{2}L(\frac{K+1}{2^{3|N|}} - \frac{1}{\phi(N)}) + \frac{1}{2}(\phi(N)-L)(\frac{1}{\phi(N)} - \frac{K}{2^{3|N|}}) \\
=&\frac{L(\phi(N)-L)}{2^{3|N|}\phi(N)}
\end{aligned}$$

Therefore, the distributions of v_0 and v_1 in Game_0 and Game_1 are statistically indistinguishable. Similar to the above, we can show that in Game_0 and Game_1 the distributions of replies by $\mathcal{F}$ to random oracle queries are also statistically indistinguishable.

Since the distribution of $\alpha_j \bmod e$ is statistically indistinguishable from the uniform distribution over $\mathbb{Z}_e$, the distributions of s_j in Game_0 and Game_1 are also statistically indistinguishable. Let $K', L' (\neq 0) \in \mathbb{Z}$ be integers such that $2^{3|N|} = K'e + L'$. The statistical distance between the distribution $U(\{0,1\}^{3|N|}) \bmod e$ obtained by choosing an element from $\{0,1\}^{3|N|}$ uniformly at random and reducing it modulo e, and the uniform distribution $U(\mathbb{Z}_e)$ over $\mathbb{Z}_e$ satisfies

$$\begin{aligned}
&\Delta(U(\{0,1\}^{3|N|}) \bmod e, U(\mathbb{Z}_e)) \\
=&\frac{1}{2}L'(\frac{K'+1}{2^{3|N|}} - \frac{1}{e}) + \frac{1}{2}(e-L')(\frac{1}{e} - \frac{K'}{2^{3|N|}}) \\
=&\frac{L'(e-L')}{2^{3|N|}e}.
\end{aligned}$$

Suppose that $\mathcal{F}$ queries to the signing oracle at q times. The advantages of $\mathcal{F}$ in Game_0 and Game_1 satisfy

$$|\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_0}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda)| \leq 2 \cdot \frac{L(\phi(N)-L)}{2^{3|N|}\phi(N)} + q \cdot \frac{L'(e-L')}{2^{3|N|}e} \leq \frac{1}{2^{\lambda}}.$$

□

Lemma 4.3.2.

$$|\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_2}(\lambda)| \leq \mathsf{Adv}_{\mathcal{F}}^{\mathsf{CRHF}}(\lambda).$$

Proof. Let F_1 and F_2 be the events where Game_1 and Game_2 output 1, respectively. Let E_2 be an event in which the challenger aborts in Game_2. Since Game_2 equals Game_1 if the challenger does not abort in Game_2, we have $\Pr[F_2 \wedge \neg E_2] = \Pr[F_1 \wedge \neg E_2]$. From the difference lemma, we have $|\Pr[F_1] - \Pr[F_2]| < \Pr[E_2]$, and so

$$|\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_2}(\lambda)| \leq \mathsf{Adv}_{\mathcal{F}}^{\mathsf{CRHF}}(\lambda).$$

□

Lemma 4.3.3.

$$\mathsf{Adv}^{\mathsf{Game}_2}_{\mathcal{F}}(\lambda) \leq \mathsf{Adv}^{\mathsf{RSA}}_{\mathcal{B}}(\lambda) + \frac{1}{\Theta(2^{\lambda})}.$$

Proof. We prove this lemma by constructing a PPT algorithm $\mathcal{B}$ that uses $\mathcal{F}$ to solve the RSA problem when given the RSA instance (N, e, y).

Suppose that $\mathcal{F}$ finally outputs a forgery $(m^*, (\sigma^*, r^*, s^*))$ $(m^* \notin \{m_j\}_{j\in[q]})$ in Game_2. We now consider the following $(q+1) \times (q+3)$ matrix $\mathbf{A}$ for tuples $(m_j, (\sigma_j, r_j, s_j))$ (where $j = 1, \ldots, q$, m_j is a queried message, and (σ_j, r_j, s_j) is a reply from the signing oracle for m_j) and the forgery $(m^*, (\sigma^*, r^*, s^*))$ of $\mathcal{F}$:

$$\mathbf{A} := \begin{pmatrix} 1 & h(m_1) & s_1 & 0 & \cdots & & & 0 \\ 1 & h(m_2) & 0 & s_2 & 0 & \cdots & & 0 \\ & \vdots & & & & & & \\ 1 & h(m_q) & 0 & \cdots & & & 0 \; s_q & 0 \\ 1 & h(m^*) & 0 & \cdots & & & 0 & s^* \end{pmatrix}.$$

Then $\mathbf{A}$ satisfies the following equation:

$$\mathbf{A}\begin{pmatrix} \beta_0 \\ \beta_1 \\ \alpha_1 \\ \vdots \\ \alpha_q \\ \alpha^* \end{pmatrix} \equiv \begin{pmatrix} 0 \\ \vdots \\ 0 \\ S := \beta_0 + h(m^*)\beta_1 + s^*\alpha^* \bmod e \end{pmatrix} \pmod{e}, \tag{4.1}$$

where α^* is obtained from $(r^*, y^{\alpha^*} \bmod N, \alpha^*)$ in the H-List. Because of $\gcd(e, \phi(N)) = 1$, the values in $\mathbb{Z}_e$ are hidden from the view of $\mathcal{F}$. We want to show that $S := \beta_0 + h(m^*)\beta_1 + s^*\alpha^* \bmod e$ is distributed uniformly over $\mathbb{Z}_e$ from the view of $\mathcal{F}$. To do so, we consider the following two cases:

1. When $r^* \notin \{r_j\}_{j\in[q]}$, S is (almost) uniformly distributed over $\mathbb{Z}_e$ since $\mathrm{rank}(\mathbf{A}) = q + 1$ and α^* is (almost) uniformly distributed over $\mathbb{Z}_e$.

2. When $r^* \in \{r_j\}_{j\in[q]}$ (that is, $r^* = r_j$ for some $j \in [q]$) , the equation (4.1) leads to

$$\underbrace{\begin{pmatrix} 1 & h(m_1) & s_1 & 0 & \cdots & & & 0 \\ 1 & h(m_2) & 0 & s_2 & 0 & \cdots & & 0 \\ & \vdots & & & & & & \\ 1 & h(m_i) & 0 & \cdots & 0 & s_i & 0 \cdots & 0 \\ & \vdots & & & & & & \\ 1 & h(m_q) & 0 & \cdots & & & 0 & s_q \\ 1 & h(m^*) & 0 & \cdots & 0 & s^* & 0 \cdots & 0 \end{pmatrix}}_{=:\mathbf{A}'} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \alpha_1 \\ \vdots \\ \alpha_q \end{pmatrix} \equiv \begin{pmatrix} 0 \\ \vdots \\ 0 \\ S \end{pmatrix} \pmod{e}.$$

Let $\mathbf{A}^*$ be the $q \times (q+2)$ matrix obtained by extracting q rows from the top of $\mathbf{A}'$, and $(\beta_0, \beta_1, \alpha_1, \ldots, \alpha_q)$ is distributed uniformly over the kernel $\mathrm{Ker}(f_{\mathbf{A}^*})$ of the linear transformation $f_{\mathbf{A}^*}(\mathbf{x}) := \mathbf{A}^*\mathbf{x} \bmod e$. Since $\mathrm{Ker}(f_{\mathbf{A}^*})$ is the dimension of 2, for vectors $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{Z}_e^{q+2}$ and integers $c_1, c_2 \in \mathbb{Z}_e$, we have $(\beta_0, \beta_1, \alpha_1, \ldots, \alpha_q) = c_1 \cdot \mathbf{b}_1 + c_2 \cdot \mathbf{b}_2 \in \mathbb{Z}_e^{q+2}$, where c_1 and c_2 are distributed uniformly since $(\beta_0, \beta_1, \alpha_1, \ldots, \alpha_q)$ are (almost) uniformly distributed over $\mathrm{Ker}(f_{\mathbf{A}^*})$. Let $\mathbf{a}$ be the $q+1$-th column of $\mathbf{A}'$. For any output m^*, s^* ($m^* \notin \{m_j\}_{j\in[q]}$) of $\mathcal{F}$, $\mathbf{a}$ is linearly independent of all column vectors of $\mathbf{A}^*$ because of $h(m^*) \notin \{h(m_j)\}_{j\in[q]}$. Since $S = \langle \mathbf{a}, (\beta_0, \beta_1, \alpha_1, \ldots, \alpha_q)\rangle = c_1 \cdot \langle \mathbf{a}, \mathbf{b}_1\rangle + c_2 \cdot \langle \mathbf{a}, \mathbf{b}_2\rangle \bmod e$, $\langle \mathbf{a}, \mathbf{b}_1\rangle$ and $\langle \mathbf{a}, \mathbf{b}_2\rangle$ are not equal to 0, and c_1 and c_2 are (almost) uniformly distributed over $\mathbb{Z}_e$, S is (almost) uniformly distributed over $\mathbb{Z}_e$ from the view of $\mathcal{F}$.

For any forgery $(m^*, (\sigma^*, r^*, s^*))$ of $\mathcal{F}$, the probability of $e \mid \beta_0 + h(m^*)\beta_1 + s^*\alpha^*$ is at most $1/e$. If we have $e \nmid \beta_0 + h(m^*)\beta_1 + s^*\alpha^*$, then there exist $K_0, L_0 \neq 0 \in \mathbb{Z}$ such that $\beta_0 + h(m^*)\beta_1 + s^*\alpha^* = K_0 e + L_0$. Since we have $(\sigma^*)^e \equiv v_0 v_1^{h(m^*)} H(r^*)^{s^*} \equiv y^{K_0 e + L_0} \pmod N$, it holds that $(\sigma^* y^{-K_0})^e \equiv y^{L_0} \pmod N$. $\mathcal{B}$ can efficiently compute $a, b \in \mathbb{Z}$ such that $ae + bL_0 = 1$ because of $\gcd(e, L_0) = 1$. Since we have $(\sigma^* y^{-K_0})^{be} y^{ae} \equiv y^{ae+bL_0} \equiv y \pmod N$ (that is, $((\sigma^* y^{-K_0})^b y^a)^e \equiv y \pmod N$), $x := (\sigma^* y^{-K_0})^b y^a$ satisfies $x^e \equiv y \pmod N$. $\mathcal{B}$ outputs x as a solution to the given RSA problem.

Then the following equation holds:

$$\mathsf{Adv}_{\mathcal{B}}^{\mathsf{RSA}}(\lambda) \geq \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_2}(\lambda)\left(1 - \frac{1}{e}\right),$$

and so we can obtain

$$\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_2}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{RSA}}(\lambda) + \frac{1}{\Theta(2^\lambda)}.$$

□

4.4 Digital Signature Scheme Σ_{SM} in the Standard Model

In this chapter, we show that for any PPT adversary there exists a concrete hash function that can replace the random oracle of the signature scheme Σ_{ROM} proposed in Section 4.3. Such a hash function is built from indistinguishability obfuscation [GGH+13b, GGH15]. The statement proven in this chapter gives a guarantee that the construction of Σ_{ROM} does not lead to the impossibility results such as in [CGH98, GK03, BBP04]. The construction of Σ_{SM} is given in Section 4.4.1, and its security is proven in Section 4.4.2.

Figure 4.1: Hash

Constants: RSA modulus N, key K of pseudorandom function F, and $y \in \mathbb{Z}_N^*$.
Input: $r \in \{0,1\}^\gamma$。

1. Output $F_0(K; r)$.

4.4.1 Construction

We show that there exists a concrete hash function that can replace the random oracle of Σ_{ROM} while maintaining security. To do this, we construct a program Hash as described in Figure 4.1, where if we let $K \xleftarrow{R} \mathsf{Key}_F(1^\lambda)$ be a key for the PRF $F : \{0,1\}^\gamma \to \{0,1\}^{3|N|}$, and let $y \xleftarrow{U} \mathbb{Z}_N^*$, then the PRF $F_0 : \{0,1\}^\gamma \to \mathbb{Z}_N^*$ is defined as:

$$F_0(K; r) := y^{F(K;r)} \bmod N.$$

We set an obfuscation of the program Hash, $H := i\mathcal{O}(\mathsf{Hash})$, as the hash function H in the standard model. To prove that the signature scheme with H instead of the random oracle is EUF-CMA secure, we use obfuscations of the programs described in Figure 4.2 and 4.3, where q is the number of signing queries made by the adversary.

4.4.2 Security

In Theorem 4.3.2, we prove that the security of the proposed scheme Σ_{ROM} is tightly reducible to the RSA assumption in the random oracle model. In the following theorem, we show that even if the random oracle of Σ_{ROM} is replaced by a concrete hash function, the signature scheme remains secure. Particularly, we show that for any PPT adversary $\mathcal{F}$, there exists a concrete hash function H (whose construction depends on the runtime of F) such that it can replace the random oracle of Σ_{ROM} while maintaining the security. In the following, we call Σ_{SM} the signature scheme in which the random oracle of Σ_{ROM} is replaced by the hash function H.

Theorem 4.4.1. *If the RSA assumption holds, h is a collision-resistant hash function, $i\mathcal{O}$ is an indistinguishability obfuscator, and F_0 is an puncturable pseudorandom function, for any PPT algorithm $\mathcal{F}$ there exists a hash function H that satisfies the following: for some PPT algorithms* Samp, $\mathcal{D}, \mathcal{A}_1, \mathcal{A}_2$ *and* $\mathcal{B}$ *it holds*

Constants: RSA modulus N, set $S := \{r_j\}_{j\in[q]}$ of punctured points, punctured key K_S, random integer $y \in \mathbb{Z}_N^*$, and $\{c_j := F_0(K; r_j)\}_{j\in[q]}$.
Input $r \in \{0,1\}^\gamma$.

1. If $r = r_j$ for some $j \in [q]$, output c_j,
2. else output $F_0(K_S; r)$.

Figure 4.2: Hash functionHash^*

Constants: RSA modulus N, set $S := \{r_j\}_{j\in[q]}$ of punctured points, punctured key K_S, random integer $y \in \mathbb{Z}_N^*$, and $\{c_j := y^{\alpha_j} \bmod N\}_{j\in[q]}$.
Input: $r \in \{0,1\}^\gamma$.

1. If $r = r_j$ for some $j \in [q]$, output c_j,
2. else output $F_0(K_S; r)$.

Figure 4.3: Hash function Hash^{**}

that

$$\mathsf{Adv}^{\text{EUF-CMA}}_{\Sigma_2,\mathcal{F}}(\lambda) \le \mathsf{Adv}^{i\mathcal{O}}_{\mathsf{Samp},\mathcal{D}}(\lambda)+\mathsf{Adv}^{\text{pPRF}}_{\mathcal{A}_1,\mathcal{A}_2}(\lambda)+\mathsf{Adv}^{\text{RSA}}_{\mathcal{B}}(\lambda)+\mathsf{Adv}^{\text{CRHF}}_{\mathcal{F}}(\lambda)+\frac{1}{\Theta(2^\lambda)}+\frac{1}{2^\lambda},$$

where we have $e = \Theta(2^\lambda)$.

We use the sequence of the following games to prove Theorem 4.4.1. For security parameter $\lambda \in \mathbb{N}$, let $\mathsf{Adv}^{\mathsf{Game}_i}_{\mathcal{F}}(\lambda)$ be the advantage of $\mathcal{F}$ in Game_i ($i = 0, 1, 2, 3, 4$).

- Game_0: This is the same as the original EUF-CMA game, so we have $\mathsf{Adv}^{\text{EUF-CMA}}_{\Sigma_2,\mathcal{F}}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{A}}(\lambda)$.
- Game_1: This is the same as Game_0 except that the challenger generates an obfuscated program of Hash^* instead of Hash.
- Game_2: This is the same as Game_1 except that the challenger generates an obfuscated program of Hash^{**} instead of Hash^*. Here the challenger chooses random integers $\alpha_j \xleftarrow{U} \{0,1\}^{3|N|}$ ($j \in [q]$) and sets $\{y^{\alpha_j} \bmod N\}_{j\in[q]}$ as constants of Hash^{**}.

- Game_3: This is the same as Game_2 except for the way to compute v_0 and v_1 and to reply to signing queries. The challenger chooses a random integer $y \xleftarrow{U} \mathbb{Z}_N^*$. Instead of choosing v_0 and v_1 uniformly at random, the challenger chooses $\beta_0, \beta_1 \xleftarrow{U} \{0,1\}^{3|N|}$ ($\beta_0, \beta_1 \not\equiv 0 \pmod{e}$) uniformly at random, and sets $v_0 := y^{\beta_0} \bmod N$, $v_1 := y^{\beta_1} \bmod N$.

  Let $m_1, \ldots, m_q$ be messages queried by $\mathcal{F}$. For the j-th query m_j, the challenger computes $s_j \in \mathbb{Z}_e$ such that $e \mid \beta_0 + \beta_1 h(m_j) + s_j \alpha_j$, that is, $s_j := (-\beta_0 - \beta_1 h(m_j))/\alpha_j \bmod e$. Let $K_j := (\beta_0 + \beta_1 h(m_j) + s_j \alpha_j)/e$. The challenger sets $\sigma_j := y^{K_j} \bmod N$, and returns (σ_j, r_j, s_j) (where r_j is a punctured point of F_0) as a signature for m_j. The triple (σ_j, r_j, s_j) satisfies

  $$\sigma_j^e \equiv v_0 v_1^{h(m_j)} H(r_j)^{s_j} \pmod{N},$$

  and so it is a valid signature for m_j.

- Game_4: This is the same as Game_3 except that the challenger aborts if the forgery $(m^*, (\sigma^*, r^*, s^*))$ of $\mathcal{F}$ satisfies $m^* \notin \{m_j\}_{j \in [q]}$ and $h(m^*) \in \{h(m_j)\}_{j \in [q]}$.

To prove Theorem 4.4.1, we must prove the following four lemmas.

Lemma 4.4.1.

$$|\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_0}(\lambda)| = \mathsf{Adv}_{\mathsf{Samp},\mathcal{D}}^{i\mathcal{O}}(\lambda).$$

Proof. To prove this lemma, we use $\mathcal{F}$ to construct an adversary $(\mathsf{Samp}, \mathcal{D})$ for an indistinguishability obfuscator $i\mathcal{O}$.

$\mathsf{Samp}(1^\lambda)$ first generates $(N, P, Q, e) \xleftarrow{R} \mathsf{GenRSA}(1^\lambda)$ and computes an integer $d \in \mathbb{Z}$ such that $ed \equiv 1 \pmod{\phi(N)}$. It then builds the programs of Hash and Hash^*, where the programs are padded so that both are the same size. Let $h : \{0,1\}^* \to \mathbb{Z}_e$ be a collision-resistant hash function parameterized by λ. $\mathsf{Samp}(1^\lambda)$ chooses random integers $v_0, v_1 \xleftarrow{U} \mathbb{Z}_N^*$, and outputs $C_0 := \mathsf{Hash}$, $C_1 := \mathsf{Hash}^*$, and $\tau := (h, v_0, v_1, N, e, d)$.

By construction, C_0 and C_1 always behave identically for every input. With suitable padding, both C_0 and C_1 are the same size.

The algorithm $\mathcal{D}$ takes as input τ and either the obfuscation H of C_0 or C_1. It sets the verification key $\mathsf{vk} := (H, h, v_0, v_1, N, e)$, and invokes the adversary $\mathcal{F}$ by giving vk to him. For a signing query m_j ($j \in [q]$) made by $\mathcal{F}$, $\mathcal{D}$ uses $\mathsf{sk} := d$ to generate a valid signature (σ_j, r_j, s_j) for m_j. $\mathcal{F}$ finally outputs $(m^*, (\sigma^*, r^*, s^*))$, and wins if $\mathsf{Verify}_{\mathsf{vk}}(m^*, (\sigma^*, r^*, s^*)) = 1$. $\mathcal{D}$ outputs 1 if $\mathcal{F}$ wins, and 0 otherwise.

By the construction of $\mathcal{D}$, it holds that $\Pr[\mathcal{D}(H, \tau) \to 1] = \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_0}(\lambda)$ if $H := i\mathcal{O}(C_0)$, and $\Pr[\mathcal{D}(H, \tau) \to 1] = \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda)$ if $H := i\mathcal{O}(C_1)$. Therefore,

$$\mathsf{Adv}_{\mathsf{Samp},\mathcal{D}}^{i\mathcal{O}}(\lambda) = |\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_0}(\lambda)|.$$

□

Lemma 4.4.2.

$$|\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_2}(\lambda)| = \mathsf{Adv}_{\mathcal{A}_1,\mathcal{A}_2}^{\mathsf{pPRF}}(\lambda).$$

Proof. To prove this lemma, we use the forger $\mathcal{F}$ to construct an adversary $(\mathcal{A}_1, \mathcal{A}_2)$ that breaks the pseudorandom property at punctured points of F_0.

The algorithm $\mathcal{A}_1$ simply chooses $r_j \xleftarrow{U} \{0,1\}^\gamma$ $(j = 1, \ldots, q)$ uniformly at random, and outputs $S := \{r_j\}_{j\in[q]}$.

The algorithm $\mathcal{A}_2$ takes as input a set of punctured points S, punctured key K_S, and either $\{c_j := F_0(K_S; r_j)\}_{j\in[q]}$ or $\{c_j \xleftarrow{U} \mathbb{Z}_N^*\}_{j\in[q]}$. It follows the algorithm Setup of Σ_{SM} to generate keys $\mathsf{vk} := (H, h, v_0, v_1, N, e)$ and $\mathsf{sk} := d$, where the obfuscated program H is obtained by using $i\mathcal{O}$ to obfuscate the program of the hash function constructed from the input $\{c_j\}_{j\in[q]}$. $\mathcal{A}_2$ invokes $\mathcal{F}$, which makes signing queries m_j $(j \in [q])$. It uses sk to make a valid signature (σ_j, r_j, s_j) for m_j.

The forger $\mathcal{F}$ finally outputs $(m^*, \sigma^*, r^*, s^*)$, and wins if $\mathsf{Verify}_{\mathsf{vk}}(m^*, \sigma^*, r^*, s^*) = 1$. $\mathcal{A}_2$ outputs 1 if $\mathcal{F}$ wins, and 0 otherwise.

In the case where the adversary $\mathcal{A}_2$ is given $\{c_j := F_0(K_S; r_j)\}_{j\in[q]}$, it holds that $\Pr[\mathcal{A}_2(S, K_S, \mathsf{Eval}_{F_0}(K, S)) \to 1] = \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda)$ since the hash function H is the obfuscation of Hash^*, that is, $H = i\mathcal{O}(\mathsf{Hash}^*)$. For $\alpha_j \xleftarrow{U} \{0,1\}^{3|N|}$, the distribution of $y^{\alpha_j} \bmod N$ is statistically indistinguishable from uniform on $\mathbb{Z}_N^*$. In the case that $\mathcal{A}_2$ is given $\{c_j \xleftarrow{U} \mathbb{Z}_N^*\}_{j\in[q]}$, we have $\Pr[\mathcal{A}_2(S, K_S, U(\{0,1\}^{q\cdot|N|})) \to 1] = \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_2}(\lambda)$ because of $H = i\mathcal{O}(\mathsf{Hash}^{**})$. Therefore, we have

$$\mathsf{Adv}_{\mathcal{A}_1,\mathcal{A}_2}^{\mathsf{pPRF}}(\lambda) = |\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_2}(\lambda)|.$$

□

Lemma 4.4.3.

$$|\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_2}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_3}(\lambda)| \leq \frac{1}{2^\lambda}.$$

Proof. We can prove this lemma in a way similar to Lemma 4.3.1. □

Lemma 4.4.4.

$$|\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_3}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_4}(\lambda)| \leq \mathsf{Adv}_{\mathcal{F}}^{\mathsf{CRHF}}(\lambda).$$

Proof. We can prove this lemma in a way similar to Lemma 4.3.2. □

Lemma 4.4.5.

$$\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_4}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{RSA}}(\lambda) + \frac{1}{\Theta(2^\lambda)}.$$

Proof. We use the adversary $\mathcal{F}$ in Game_4 to construct a PPT algorithm $\mathcal{B}$ that breaks the RSA assumption. $\mathcal{B}$ takes as input an instance of the RSA problem (N, e, y), where e is a prime number such that $\gcd(\phi(N), e) = 1$ and $y \in \mathbb{Z}_N^*$.

$\mathcal{F}$ finally outputs the forgery $(m^*, (\sigma^*, r^*, s^*))$. $\mathcal{B}$ first computes $\alpha^* := F(K; r^*)$. Similar to the proof of Lemma 4.3.3, $(\beta_0 + h(m^*)\beta_1 + s^*\alpha^*) \bmod e$ is (almost) uniformly distributed over $\mathbb{Z}_e$, so the probability of $e \mid \beta_0 + h(m^*)\beta_1 + s^*\alpha^*$ is at most $1/e$. The fact that $e \nmid \beta_0 + h(m^*)\beta_1 + s^*\alpha^*$ leads to the fact that there exist $K_0, L_0 (\neq 0) \in \mathbb{Z}$ such that $\beta_0 + h(m^*)\beta_1 + s^*\alpha^* = K_0 e + L_0$. Since $(\sigma^*)^e \equiv v_0 v_1^{h(m^*)} H(r^*)^{s^*} \equiv y^{K_0 e + L_0} \pmod{N}$, we have $(\sigma^* y^{-K_0})^e \equiv y^{L_0} \pmod{N}$. By $\gcd(e, L_0) = 1$, $\mathcal{B}$ can efficiently compute $a, b \in \mathbb{Z}$ such that $ae + bL_0 = 1$. This leads to $(\sigma^* y^{-K_0})^{be} y^{ae} \equiv y \pmod{N}$, and so we have $((\sigma^* y^{-K_0})^b y^a)^e \equiv y \pmod{N}$. $\mathcal{B}$ can efficiently obtain a solution to the given RSA problem $x = (\sigma^* y^{-K_0})^b y^a$ that satisfies $x^e \equiv y \pmod{N}$.

From the above, the advantages of $\mathcal{F}$ and $\mathcal{B}$ satisfy

$$\mathsf{Adv}_{\mathcal{B}}^{\mathsf{RSA}}(\lambda) \geq \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_4}(\lambda)\left(1 - \frac{1}{e}\right),$$

and so we can obtain

$$\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_4}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{RSA}}(\lambda) + \frac{1}{\Theta(2^\lambda)}.$$

□

4.5 Instantiating pFDH with a Concrete Hash Function

In this chapter, we show that for any PPT adversary there exists a concrete hash function that can directly replace the pFDH of [Cor02] (modeled as the random oracle). We give the construction of the proposed signatures in Section 4.5.1. We then present the construction of the concrete hash function and prove the security in Section 4.5.2.

4.5.1 Construction

The proposed signature scheme $\Sigma_{Coron} := (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Verify})$ consists of the following three algorithms:

- $\mathsf{Setup}(1^\lambda)$: Generate an RSA instance $(N, P, Q, e) \xleftarrow{R} \mathsf{GenRSA}(1^\lambda)$, where e is a prime number such that $|e| = |N|$ and $\gcd(e, \phi(N)) = 1$. Compute an integer $d \in \mathbb{Z}$ such that $ed \equiv 1 \pmod{\phi(N)}$. Let $\gamma = \gamma(\lambda)$ be a polynomial in λ. Generate a PRF key $K \xleftarrow{R} \mathsf{Key}_F(1^\lambda)$ for $F : \{0,1\}^\gamma \to \{0,1\}^{3|N|}$. Choose

a random integer $y \xleftarrow{U} \mathbb{Z}_N^*$, and define the following pseudorandom function $F_0 : \{0,1\}^\gamma \to \mathbb{Z}_N^*$:

$$F_0(K; r) := y^{F(K;r)} \bmod N.$$

Let $h : \{0,1\}^* \to \mathbb{Z}_e$ be a collision-resistant hash function parameterized by λ. Choose random integers $v_0, v_1 \xleftarrow{U} \mathbb{Z}_N^*$. Create an obfuscation $H := i\mathcal{O}(\mathsf{pFDH})$ of the full-domain hash pFDH of Figure 4.4. The obfuscated program H maps elements in $\{0,1\}^\gamma$ to elements in $\mathbb{Z}_N^*$. Output the verification key $\mathsf{vk} := (H, N, e)$ and signing key $\mathsf{sk} := d$.

- $\mathsf{Sign}_{\mathsf{sk}}(m \in \{0,1\}^*)$: Choose $r \xleftarrow{U} \{0,1\}^\gamma$ and $s \xleftarrow{U} \mathbb{Z}_e$ uniformly at random, and compute

  $$\sigma := (H(m, r, s))^d \bmod N.$$

  Output (σ, r, s) as a signature for m.

- $\mathsf{Verify}_{\mathsf{vk}}(m, (\sigma, r, s))$: Output 1 if the following holds, and 0 otherwise:

  $$\sigma^e \equiv H(m, r, s) \pmod N.$$

The correctness of Σ_{Coron} can immediately be proven from the equation in Verify.

Theorem 4.5.1 (Correctness)**.** *The above signature scheme Σ_{Coron} is correct.*

Proof. A signature of a message m is a triple (σ, r, s) for randomnesses $r \in \{0,1\}^\gamma$ and $s \in \mathbb{Z}$ such that

$$\sigma := (H(m, r, s))^d \bmod N,$$

where d is the signing key. Therefore, we have

$$\begin{aligned}\sigma^e &\equiv ((H(m, r, s))^d)^e \\ &\equiv H(m, r, s) \bmod N \ (\because e \cdot d \equiv 1 \bmod \phi(N)).\end{aligned}$$

□

4.5.2 Security

In this section, we show that the signature scheme Σ_{Coron} described in Section 4.5.1 is unforgeable. To prove that, we use the three full-domain hashes of Figure 4.4, 4.5, and 4.6.

Constants: RSA modulus N, PRF key K, and random integers $v_0, v_1, y \in \mathbb{Z}_N^*$.
Inputs: $m \in \{0,1\}^*$, $r \in \{0,1\}^\gamma$ and $s \in \mathbb{Z}_e$.

1. Output $v_0 v_1^{h(m)} (F_0(K; r))^s \bmod N$.

Figure 4.4: pFDH pFDH

Constants: RSA modulus N, set of punctured points $S := \{r_j\}_{j\in[q]}$, punctured PRF key K_S, random integers $v_0, v_1, y \in \mathbb{Z}_N^*$, and $\{c_j := F_0(K; r_j)\}_{j\in[q]}$.
Inputs: $m \in \{0,1\}^*$, $r \in \{0,1\}^\gamma$ and $s \in \mathbb{Z}_e$.

1. If $r = r_j$ for some $j \in [q]$, output $v_0 v_1^{h(m)} (c_j)^s \bmod N$ and exit.
2. Output $v_0 v_1^{h(m)} (F_0(K_S; r))^s \bmod N$.

Figure 4.5: pFDH pFDH^*

Theorem 4.5.2. *If the RSA assumption holds, h is a collision-resistant hash function, $i\mathcal{O}$ is an indistinguishability obfuscator, and F_0 is a puncturable pseudorandom function, for any PPT algorithm $\mathcal{F}$ there exists a hash function that satisfies the following: for some PPT algorithms $\mathsf{Samp}, \mathcal{D}, \mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{B}$ it holds that*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Sigma_{Coron},\mathcal{F}}(\lambda) \leq \mathsf{Adv}^{i\mathcal{O}}_{\mathsf{Samp},\mathcal{D}}(\lambda) + \mathsf{Adv}^{\mathsf{pPRF}}_{\mathcal{A}_1,\mathcal{A}_2}(\lambda) + \mathsf{Adv}^{\mathsf{RSA}}_{\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{F}}(\lambda) + \frac{1}{\Theta(2^\lambda)} + \frac{1}{2^\lambda},$$

where the public key e in the RSA assumption satisfies $e = \Theta(2^\lambda)$.

We use a sequence of games to prove this theorem. The proof is almost the same as in Section 4.4.2. For a security parameter λ, let $\mathsf{Adv}^{\mathsf{Game}_i}_{\mathcal{F}}(\lambda)$ be an advantage of $\mathcal{F}$ in Game_i ($i = 0, 1, 2, 3, 4$).

- Game_0: This is the same as the original EUF-CMA game. That is, we have $\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\Sigma_{Coron},\mathcal{F}}(\lambda) = \mathsf{Adv}^{\mathsf{Game}_0}_{\mathcal{F}}(\lambda)$.
- Game_1: This is the same as Game_0 except that the challenger creates an obfuscation of pFDH^* instead of pFDH.
- Game_2: This is the same as Game_1 except that the challenger creates an obfuscation of pFDH^{**} instead of pFDH^*, where the challenger chooses $\alpha_j \xleftarrow{U} \{0,1\}^{3|N|}$ uniformly at random and sets $\{y^{\alpha_j} \bmod N\}_{j\in[q]}$ as constants of pFDH^{**}.

Constants: RSA modulus N, set of punctured points $S := \{r_j\}_{j\in[q]}$, punctured PRF key K_S, random integers $v_0, v_1, y \in \mathbb{Z}_N^*$, and $\{c_j := y^{\alpha_j} \bmod N\}_{j\in[q]}$.
Inputs: $m \in \{0,1\}^*$, $r \in \{0,1\}^\gamma$ and $s \in \mathbb{Z}_e$.

1. If $r = r_j$ for some $j \in [q]$, output $v_0 v_1^{h(m)} (c_j)^s \bmod N$ and exit.
2. Output $v_0 v_1^{h(m)} (F_0(K_S; r))^s \bmod N$.

Figure 4.6: pFDH pFDH^{**}

- Game_3: This is the same as Game_2 except for the ways to compute v_0 and v_1 in vk and to make replies to the signing queries. The challenger first chooses a random integer $y \xleftarrow{U} \mathbb{Z}_N^*$. It then chooses $\beta_0, \beta_1 \xleftarrow{U} \{0,1\}^{3|N|}$ ($\beta_0, \beta_1 \not\equiv 0 \pmod{e}$), and computes $v_0 := y^{\beta_0} \bmod N$ and $v_1 := y^{\beta_1} \bmod N$.

  Suppose that $\mathcal{F}$ makes signing queries $m_1, \ldots, m_q$. For the j-th signing query m_j, the challenger computes $s_j \in \mathbb{Z}_e$ such that $e \mid \beta_0 + \beta_1 h(m_j) + s_j \alpha_j$, that is, the challenger computes $s_j := (-\beta_0 - \beta h(m_j))/\alpha_j \bmod e$. The challenger computes $\sigma_j := y^{K_j} \bmod N$ for $K_j := (\beta_0 + \beta_1 h(m_j) + s_j \alpha_j)/e$, and returns (σ_j, r_j, s_j) to $\mathcal{F}$ where r_j is a punctured point of F_0.

- Game_4: This is the same as Game_3 except that the challenger aborts if the forgery $(m^*, (\sigma^*, r^*, s^*))$ of $\mathcal{F}$ satisfies $m^* \notin \{m_j\}_{j\in[q]}$ and $h(m^*) \in \{h(m_j)\}_{j\in[q]}$.

To prove Theorem 4.5.2, we must prove the following four lemmas. They can be proven in a way similar to Lemmas 4.4.1, 4.4.2, 4.4.3, 4.4.4, and 4.4.5, respectively.

Lemma 4.5.1.

$$|\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_0}(\lambda)| = \mathsf{Adv}_{\mathcal{D}}^{i\mathcal{O}}(\lambda).$$

Proof. To prove this lemma, we use $\mathcal{F}$ to construct an adversary $(\mathsf{Samp}, \mathcal{D})$ for an indistinguishability obfuscator $i\mathcal{O}$.

$\mathsf{Samp}(1^\lambda)$ first generates $(N, P, Q, e) \xleftarrow{R} \mathsf{GenRSA}(1^\lambda)$ and computes an integer $d \in \mathbb{Z}$ such that $ed \equiv 1 \pmod{\phi(N)}$. It then builds the programs of pFDH and pFDH^*, where the programs are padded so that both are the same size. Let $h : \{0,1\}^* \to \mathbb{Z}_e$ be a collision-resistant hash function parameterized by λ. $\mathsf{Samp}(1^\lambda)$ chooses random integers $v_0, v_1 \xleftarrow{U} \mathbb{Z}_N^*$, and outputs $C_0 := \mathsf{pFDH}$, $C_1 := \mathsf{pFDH}^*$, and $\tau := (N, e, d)$.

By construction, C_0 and C_1 always behave identically on every input. With suitable padding, both C_0 and C_1 are the same size.

The algorithm $\mathcal{D}$ takes as input τ and either the obfuscation H of C_0 or C_1. It sets the verification key $\mathsf{vk} := (H, N, e)$, and invokes the adversary $\mathcal{F}$ by giving vk to him. For a signing query m_j ($j \in [q]$) made by $\mathcal{F}$, $\mathcal{D}$ uses $\mathsf{sk} := d$ to generate a valid signature (σ_j, r_j, s_j) for m_j. $\mathcal{F}$ finally outputs $(m^*, (\sigma^*, r^*, s^*))$, and wins if $\mathsf{Verify}_{\mathsf{vk}}(m^*, (\sigma^*, r^*, s^*)) = 1$. $\mathcal{D}$ outputs 1 if $\mathcal{F}$ wins, and 0 otherwise.

By the construction of $\mathcal{D}$, it holds that $\Pr[\mathcal{D}(H, \tau) \to 1] = \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_0}(\lambda)$ if $H := i\mathcal{O}(C_0)$, and $\Pr[\mathcal{D}(H, \tau) \to 1] = \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda)$ if $H := i\mathcal{O}(C_1)$. Therefore,

$$\mathsf{Adv}^{i\mathcal{O}}_{\mathsf{Samp},\mathcal{D}}(\lambda) = |\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_0}(\lambda)|.$$

□

Lemma 4.5.2.

$$|\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_2}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda)| = \mathsf{Adv}^{\mathsf{pPRF}}_{\mathcal{A}_1,\mathcal{A}_2}(\lambda).$$

Proof. To prove this lemma, we use the forger $\mathcal{F}$ to construct an adversary $(\mathcal{A}_1, \mathcal{A}_2)$ that breaks the pseudorandom property at punctured points of F_0.

The algorithm $\mathcal{A}_1$ simply chooses $r_j \xleftarrow{U} \{0,1\}^\gamma$ ($j = 1, \ldots, q$) uniformly at random, and outputs $S := \{r_j\}_{j\in[q]}$.

The algorithm $\mathcal{A}_2$ takes as input a set of punctured points S, punctured key K_S, and either $\{c_j := F_0(K_S; r_j)\}_{j\in[q]}$ or $\{c_j \xleftarrow{U} \mathbb{Z}_N^*\}_{j\in[q]}$. It follows the algorithm Setup of Σ_{Coron} to generate keys $\mathsf{vk} := (H, N, e)$ and $\mathsf{sk} := d$, where the obfuscated program H is obtained by using $i\mathcal{O}$ to obfuscate the program of the hash function constructed from the input $\{c_j\}_{j\in[q]}$. $\mathcal{A}_2$ invokes $\mathcal{F}$, which makes signing queries m_j ($j \in [q]$). It uses sk to make a valid signature (σ_j, r_j, s_j) for m_j.

The forger $\mathcal{F}$ finally outputs $(m^*, \sigma^*, r^*, s^*)$, and wins if $\mathsf{Verify}_{\mathsf{vk}}(m^*, \sigma^*, r^*, s^*) = 1$. $\mathcal{A}_2$ outputs 1 if $\mathcal{F}$ wins, and 0 otherwise.

In the case that the adversary $\mathcal{A}_2$ is given $\{c_j := F_0(K_S; r_j)\}_{j\in[q]}$, it holds that $\Pr[\mathcal{A}_2(S, K_S, \mathsf{Eval}_{F_0}(K, S)) \to 1] = \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda)$ since the hash function H is the obfuscation of pFDH^*, that is, $H = i\mathcal{O}(\mathsf{pFDH}^*)$. For $\alpha_j \xleftarrow{U} \{0,1\}^{3|N|}$, the distribution of $y^{\alpha_j} \bmod N$ is statistically indistinguishable from uniform on $\mathbb{Z}_N^*$. In the case that $\mathcal{A}_2$ is given $\{c_j \xleftarrow{U} \mathbb{Z}_N^*\}_{j\in[q]}$, we have $\Pr[\mathcal{A}_2(S, K_S, U(\{0,1\}^{q\cdot|N|})) \to 1] = \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_2}(\lambda)$ because of $H = i\mathcal{O}(\mathsf{pFDH}^{**})$. Therefore, we have

$$\mathsf{Adv}^{\mathsf{pPRF}}_{\mathcal{A}_1,\mathcal{A}_2}(\lambda) = |\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_1}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_2}(\lambda)|.$$

□

Lemma 4.5.3.

$$|\mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_2}(\lambda) - \mathsf{Adv}_{\mathcal{F}}^{\mathsf{Game}_3}(\lambda)| \leq \frac{1}{2^\lambda}.$$

Proof. We can prove this lemma in a way similar to Lemma 4.3.1. □

Lemma 4.5.4.

$$|\mathsf{Adv}^{\mathsf{Game}_3}_{\mathcal{F}}(\lambda) - \mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{F}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{F}}(\lambda).$$

Proof. We can prove this lemma in a way similar to Lemma 4.3.2. □

Lemma 4.5.5.

$$\mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{F}}(\lambda) \leq \mathsf{Adv}^{RSA}_{\mathcal{B}}(\lambda) + \frac{1}{\Theta(2^{\lambda})}.$$

Proof. We use the adversary $\mathcal{F}$ in Game_4 to construct a PPT algorithm $\mathcal{B}$ that breaks the RSA assumption. $\mathcal{B}$ takes as input an instance of the RSA problem (N, e, y), where e is a prime number such that $\gcd(\phi(N), e) = 1$ and $y \in \mathbb{Z}^*_N$.

$\mathcal{F}$ finally outputs the forgery $(m^*, (\sigma^*, r^*, s^*))$. $\mathcal{B}$ first computes $\alpha^* := F(K; r^*)$. Similar to the proof of Lemma 4.3.3, $(\beta_0 + h(m^*)\beta_1 + s^*\alpha^*) \bmod e$ is (almost) uniformly distributed over $\mathbb{Z}_e$, so the probability of $e \mid \beta_0 + h(m^*)\beta_1 + s^*\alpha^*$ is at most $1/e$. The fact that $e \nmid \beta_0 + h(m^*)\beta_1 + s^*\alpha^*$ leads to the fact that there exist $K_0, L_0 (\neq 0) \in \mathbb{Z}$ such that $\beta_0 + h(m^*)\beta_1 + s^*\alpha^* = K_0 e + L_0$. Since $(\sigma^*)^e \equiv H(m^*, r^*, s^*) \equiv y^{K_0 e + L_0} \pmod{N}$, we have $(\sigma^* y^{-K_0})^e \equiv y^{L_0} \pmod{N}$. By $\gcd(e, L_0) = 1$, $\mathcal{B}$ can efficiently compute $a, b \in \mathbb{Z}$ such that $ae + bL_0 = 1$. This leads $(\sigma^* y^{-K_0})^{be} y^{ae} \equiv y \pmod{N}$, and so we have $((\sigma^* y^{-K_0})^b y^a)^e \equiv y \pmod{N}$. $\mathcal{B}$ can efficiently obtain a solution to the given RSA problem $x = (\sigma^* y^{-K_0})^b y^a$ that satisfies $x^e \equiv y \pmod{N}$.

From the above, the advantages of $\mathcal{F}$ and $\mathcal{B}$ satisfy

$$\mathsf{Adv}^{\mathsf{RSA}}_{\mathcal{B}}(\lambda) \geq \mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{F}}(\lambda)(1 - \frac{1}{e}),$$

and so we can obtain

$$\mathsf{Adv}^{\mathsf{Game}_4}_{\mathcal{F}}(\lambda) \leq \mathsf{Adv}^{\mathsf{RSA}}_{\mathcal{B}}(\lambda) + \frac{1}{\Theta(2^{\lambda})}.$$

□

4.6 Conclusion of This Chapter

In this chapter, we proposed new digital signatures that are tightly secure based on the RSA assumption in the random oracle model. The proposed signatures have a simpler construction than the previous tightly secure RSA-based signatures [BR96,Cor02], since fewer random oracles are used in our signatures. While Coron's signatures [Cor02] has the security reduction whose efficiency depends on the number of adversarial queries to the signing oracle, the security reduction

of our signatures does not. In other words, while the security of the Coron's signatures depends on the complexity of the adversary, the security of our signatures is independent of the adversary. Additionally, we introduced a new proof technique called the *α-β hiding technique*, which is not used in the security proofs of previous cryptosystems.

There are some impossibility results [CGH98, GK03, BBP04] to show that the random oracle of their constructions cannot be replaced by any concrete hash function. It is important to understand how we should use the random oracle so that cryptographic constructions do not lead to such impossibility results. There are some studies [HSW13, FHPS13, HSW14] that investigate the case where we can directly replace the random oracle with a concrete hash function while maintaining the same security. The goal of these studies is to obtain understandings or findings regarding the random oracle in the underlying cryptosystems. In particular, it is an important problem to see whether or not a cryptosystem from the random oracle remains secure even if the random oracle is replaced by a concrete hash function.

We showed that the random oracle of our signatures can be directly replaced by a concrete hash function from indistinguishability obfuscation while maintaining the security. We also showed that the random oracle of the Coron's signatures [Cor02] can also be directly replaced by a concrete hash function. Therefore, we consider that these results provide some findings about the use of the random oracle in the Coron's signatures [Cor02].

Chapter 5

Efficient Blind Message Signatures from the RSA Assumption

In this chapter, we introduce a new notion *blind message signatures*, which has the following features. A signer $\mathcal{S}$ executes a blind signature protocol, $\mathcal{P}$, with an user $\mathcal{U}$ and $\mathcal{S}$ is divided into two parts, $\mathcal{S}_0$ and $\mathcal{S}_1$. $\mathcal{S}_0$ accepts a request from the user $\mathcal{U}$ and knows the identity of $\mathcal{U}$. $\mathcal{S}_0$ then runs the sub-protocol of $\mathcal{P}$ with $\mathcal{U}$ which is $\mathcal{P}$ excepting the final round. $\mathcal{S}_1$ executes the final round of $\mathcal{P}$, i.e., $\mathcal{S}_1$ just sends a value to $\mathcal{U}$.

We construct efficient blind message signatures from the RSA assumption in the random oracle model. We then show that for any PPT adversary there exists a concrete hash function from indistinguishability obfuscation that replaces the random oracle with maintaining security. Based on our blind message signatures, we can derive partially blind message signatures, and concurrently secure blind message signatures by combining with the Pailler encryption.

5.1 Efficient Blind Message Signatures

5.1.1 Background

A blind signature protocol is a cryptographic protocol between two parties (user and signer) first introduced by Chaum [Cha82]. In this protocol, the user requests a signature for his message, and receives the signature from the signer, where the signed message is hidden from the signer (blindness), and the number of signatures generated by the user is not larger than the number of runs of the blind signature protocol (unforgeablity). In particular, because of the blindness, blind signatures have an important role in applications such as the electronic cash and electronic voting.

Chaum's blind signatures [Cha82] based on the RSA signatures [RSA78] was not provably secure. In [BNPS03], Bellare et al. showed that the Chaum's blind signature scheme is provably secure, but the underlying assumption is not standard.

Secure blind signatures from the standard assumptions in the random oracle model were proposed in [PS96, Poi98, AO00, Abe01, AO01], the most efficient blind signatures among these studies are the ones by Abe [Abe01].

In ordinary blind signatures, a signer cannot control attributes (e.g., the date of issue or expiration) of the signatures. For instance, if the signer wants to generate signatures that are valid during a certain period, the signer must change the public key every term. A way to deal with this problem is to use the partially blind signature scheme that was proposed by Abe and Fujisaki [AF96]. Partially blind signatures are blind signatures in which a signer can explicitly contain common public information. Since ordinary blind signatures are partially blind signatures where the common information is the null string, the notion of partially blind signatures can be seen as a generalization of blind signatures. Some partially blind signatures under the standard assumptions were proposed in [AO00, Abe01, Fis06, SC12, BPV12], and the most efficient ones are those by Abe [Abe01].

When we implement blind signatures in an application such as an electronic-cash or electronic voting system, it is important to consider concurrent executions of the blind signature protocol. Suppose that a signer is a server that concurrently runs signature generation protocols with multiple users. A malicious signer may break the blindness of the participant users (and know the secret messages of the users), or malicious users may cooperate to extract information regarding the secret key of the signing server. Concurrently-secure blind signatures under the standard assumptions were proposed in [BFPV11, BPV12, SC12].

The above blind signatures from the standard assumptions are proved under the discrete-log type assumptions. There are no known efficient blind signatures that are secure based on the RSA assumption, which is the most widely used assumption (so it can be thought of as the most reliable assumption), except for the blind signatures obtained by applying the transformation of [Poi98] to the Okamoto-Guillou-Quisquater blind signatures [PS96].

5.1.2 Our Results

As seen in the last section, various blind signature schemes were proposed until now, but there are no known efficient constructions based on the RSA assumption. Since the RSA assumption, which is thought as the most reliable assumption, guarantees securities of various cryptosystems, constructing a cryptosystem under the RSA assumption is a significantly important task in cryptography.

In this chapter, we introduce a new notion *blind message signatures*, which

has the following features. A signer $\mathcal{S}$ executes a blind signature protocol, $\mathcal{P}$, with an user $\mathcal{U}$ and $\mathcal{S}$ is divided into two parts,$\mathcal{S}_0$ and $\mathcal{S}_1$. $\mathcal{S}_0$ accepts a request from the user $\mathcal{U}$ and knows the identity of $\mathcal{U}$. $\mathcal{S}_0$ then runs the sub-protocol of $\mathcal{P}$ with $\mathcal{U}$ (say $\mathcal{P}_0$) which is $\mathcal{P}$ excepting the final round. $\mathcal{S}_1$ executes the final round of $\mathcal{P}$ (say $\mathcal{P}_1$), i.e., $\mathcal{S}_1$ just sends a value to $\mathcal{U}$.

Here, unless $\mathcal{S}_0$ and $\mathcal{S}_1$ collaborate, the protocol satisfies the requirements of blind signatures. Message m is hidden before use $\mathcal{U}$ releases the message m with a signature σ even if $\mathcal{S}_0$ and $\mathcal{S}_1$ collude.

We now assumes that the link between the message m and the user $\mathcal{U}$ revealed if $\mathcal{S}_0$ and $\mathcal{S}_1$ collude. Then we show an application of this concept, blind message signature. First, we assume that $\mathcal{S}_0$ and $\mathcal{S}_1$ do not collude usually. For example, in this application, $\mathcal{S}_0$ knows the identity of user, $\mathcal{U}$, and receives a signing request with some value B from $\mathcal{U}$. $\mathcal{S}_0$ then runs sub-protocol $\mathcal{P}_0$. After completing $\mathcal{P}_0$ with $\mathcal{U}$, $\mathcal{S}_0$ gives a string t to $\mathcal{S}_1$. Here, $\mathcal{S}_0$ keeps $(\mathcal{U}, t)$. $\mathcal{S}_1$, given t from $\mathcal{S}_0$, executes $\mathcal{P}_1$, i.e., computes Y and sends it to $\mathcal{U}$ (without knowing the identity of $\mathcal{U}$). Here, $\mathcal{S}_1$ keeps (t, Y). User $\mathcal{U}$, given Y, computes signature σ for message m. If $\mathcal{U}$ keeps message m in secret for a certain period, (for example, m is a secret patent document), message m is kept secret even if $\mathcal{S}_0$ and $\mathcal{S}_1$ collude. After a period, $\mathcal{U}$ releases m along with a signature σ. Since $\mathcal{S}_0$ and $\mathcal{S}_1$ do not collude usually, the privacy of (m, σ) is preserved, i.e., it is a blind signature. If a warrant of arrest is given to user $\mathcal{U}$ under suspicion of e.g., money laundering and illegal dealing of drugs, the police orders $\mathcal{S}_0$ and $\mathcal{S}_1$ to provide the record on $\mathcal{U}$. Given $(\mathcal{U}, t)$ and (t, Y) from $\mathcal{S}_0$ and $\mathcal{S}_1$,, the police traces (m, σ) to be the signature message of $\mathcal{U}$ from information $(\mathcal{U}, Y)$.

Our blind message signatures are proven secure under the RSA assumption in the random oracle model. Particularly, we present blind message signatures, partially blind message signatures, and concurrently secure blind message signatures.

We first construct blind message signatures from the signatures described in Chapter 4, so this blind message signature scheme is secure in the random oracle model. There are known results [CGH98, GK03, BBP04] that show artificial constructions that are secure in the random oracle model but there are no concrete functions that can replace the random oracle while maintaining the scheme secure. As well as the signatures of Chapter 4, the random oracle of our blind message signatures can be replaced by a concrete hash function. We can easily construct a partially blind message signatures from our blind message signatures, and make our blind message signatures concurrently secure by using the Paillier encryption in a similar way to [Oka06].

5.1.3 Our Techniques

Blind Message Signatures BS_{ROM} based on the RSA Assumption. The scheme BS_{ROM} is constructed from the digital signatures Σ_{ROM} described in Chapter 4. Let n be a RSA modulus, (e, d) be integers such that $ed \equiv 1 \pmod{\phi(n)}$, h be a collision-resistant hash function, and H be a random oracle. The verification key is a tuple of (H, h, v_0, v_1, n, e) where $v_0, v_1 \xleftarrow{U} \mathbb{Z}_n^*$ are random integers, and d is the signing key. A signature for m of [HAO16b] is a triple (σ, r, s) where $r \in \{0,1\}^\gamma$ ($\gamma := \gamma(\lambda)$ for security parameter λ) and $s \in \mathbb{Z}_e$ are randomly chosen, and σ is computed as

$$\sigma := (v_0 v_1^{h(m)} H(r)^s)^d \bmod n.$$

To verify the signatures, check whether or not it holds that

$$\sigma^e \equiv v_0 v_1^{h(m)} H(r)^s \pmod{n}.$$

We construct the blind message signatures from the above signatures. The user chooses a random integer $R \xleftarrow{U} \mathbb{Z}_n^*$, computes $B := v_0 v_1^{h(m)} R^e \bmod n$, and sends B to the signer. The user also proves that he knows $(R, h(m))$ for B by the witness indistinguishable proofs like [Oka92]. In particular, the user sends with B a commitment $x := v_1^{r_1} r_2^e \bmod n$ ($r_1 \xleftarrow{U} \mathbb{Z}_e, r_2 \xleftarrow{U} \mathbb{Z}_n^*$) of the witness indistinguishable proof, and receives a challenge $k \xleftarrow{U} \mathbb{Z}_e$ from the signer. The user sends a response ($y_1 := r_1 + kh(m) \bmod e, y_2 := r_2 R^k \bmod n$) to the signer. The signer checks whether it holds that $xB^k \equiv v_0^k v_1^{y_1} y_2^e \pmod{n}$. If the signer accepts this witness indistinguishable proof, the signer computes $Y := (BH(r)^s)^d \bmod n$ for random $r \xleftarrow{U} \{0,1\}^\gamma$ and $s \xleftarrow{U} \mathbb{Z}_e$, and sends a tuple (Y, r, s) to the user. The user computes $\sigma := Y/R \bmod n$ and obtains a signature (σ, r, s) for m. Correctness of this scheme is immediate from the underlying signatures.

We here consider only about the unforgeability of this scheme since blindness of this scheme is immediate from the perfect witness indistinguishability of [Oka92]. The unforgeability of BS_{ROM} is reduced to the EUF-CMA security of the underlying signatures. The reduction $\mathcal{F}$ simulates the signature generation protocol with the adversarial user $\mathcal{U}^*$. The reduction first executes the signature generation protocol as the signer, and obtains a witness $(R, h(m))$ for B by using the witness extractor against the witness indistinguishable proof of [Oka92]. The reduction obtains a signature (σ, r, s) by querying $h(m)$ to the signing oracle , computes $Y := \sigma R \bmod n$, and sends (Y, r, s) to $\mathcal{U}^*$. The protocol messages by $\mathcal{F}$ and $\mathcal{U}^*$ are equally distributed as the real ones, so $\mathcal{F}$ completely simulates the signature generation protocol with $\mathcal{U}^*$. Let q_S be a number of queries to the signing oracle. The adversary $\mathcal{U}^*$ finally outputs signature forgeries $\{(m_j^*, (\sigma_j^*, r_j^*, s_j^*))\}_{j \in [q_S+1]}$. The reduction $\mathcal{F}$ outputs $(m_j^*, (\sigma_j^*, r_j^*, s_j^*))$ for $j \in [q_S + 1]$ such that $h(m_j^*)$ is not queried to the signing oracle. Therefore, $\mathcal{F}$ can break the unforgeability of the

underlying signatures at least with probability that $\mathcal{U}^*$ breaks the unforgeability of our blind message signatures.

Partially Blind Message Signatures PBS_{ROM} based on the RSA Assumption. We convert the blind message signature scheme BS_{ROM} to the partially blind message signature scheme PBS_{ROM}. Let info be a common information between the user and signer. The differences between PBS_{ROM} and BS_{ROM} are as follows. The partially blind message signature scheme PBS_{ROM} additionally outputs a random integer $v_2 \xleftarrow{U} \mathbb{Z}_n^*$. In the signature generation protocol, the user sends $B := v_0 v_1^{h(m)} v_2^{h(\mathsf{info})} R^e \bmod n$ and proves that he knows $(R, h(m), h(\mathsf{info}))$ for B by using a witness indistinguishable proof. Particularly, the user sends $x := v_1^{r_1} v_2^{r_2} r_3^e \bmod n$ $(r_1, r_2 \xleftarrow{U} \mathbb{Z}_e, r_3 \xleftarrow{U} \mathbb{Z}_n^*)$ as a commitment of the witness indistinguishable proof, and receives a challenge $k \xleftarrow{U} \mathbb{Z}_e$ from the signer. The user sends a response $(y_1 := r_1 + k \cdot h(m) \bmod e, y_2 := r_2 + k \cdot h(\mathsf{info}) \bmod e, y_3 := r_3 R^k \bmod n)$ to a signer, and checks whether it holds that $xB^k \equiv v_0^k v_1^{y_1} v_2^{y_2} y_3^e \pmod n$. The other processes of PBS_{ROM} are the same as BS_{ROM}.

The blindness of this partially blind message signature scheme is also immediate from the perfect witness indistinguishability of the internal witness indistinguishable proof system. The unforgeability of PBS_{ROM} is proven by reducing it to the unforgeability of the above BS_{ROM}. When the reduction algorithm $\mathcal{F}$ receives a public key of BS_{ROM}, (H, h, v_0, v_1, n, e), as its input, $\mathcal{F}$ gives the adversary $\mathcal{U}^*$ for PBS_{ROM} $(H, h, v_0, v_1, v_2, n, e)$ as a public key where $v_2 := v_1^e \bmod n$. To simulate the signature generation protocol of PBS_{ROM} with $\mathcal{U}^*$, the reduction $\mathcal{F}$ executes the signature generation protocol of BS_{ROM} with the challenger. Here, $\mathcal{F}$ passes to $\mathcal{U}^*$ the protocol messages from the challenger, but $\mathcal{F}$ modifies the messages from $\mathcal{U}^*$ and sends them to the challenger. In particular, from the response (y_1, y_2, y_3) of $\mathcal{U}^*$ for the challenge k, $\mathcal{F}$ computes $y_3' := v_1^{y_2} y_3 \bmod n$ and gives (y_1, y_3') to the challenger. Since it holds that $v_0^k v_1^{y_1} (y_3')^e \equiv xB^k \pmod n$, the challenger accepts $\mathcal{F}$ whenever the response from $\mathcal{U}^*$ is valid. Therefore, the reduction can correctly simulate the signature generation protocol with $\mathcal{U}^*$. If $\mathcal{F}$ runs the signature generation protocol q_S times, $\mathcal{U}^*$ finally outputs q_S+1 forgeries $\{(\hat{m}_i^*, \mathsf{info}_i^*, (\sigma_i^*, r_i^*, s_i^*))\}_{i \in [q_S+1]}$. For every $i \in [q_S+1]$, let $m_i^* := eh(\mathsf{info}_i^*) + h(\hat{m}_i^*)$. If there exists $i, j \in [q_S + 1]$ $(i \neq j)$ such that $m_i^* = m_j^*$, $\mathcal{F}$ computes $d := (h(\mathsf{info}_i^*) - h(\mathsf{info}_j^*))/(h(\hat{m}_j^*) - h(\hat{m}_i^*))$, and uses d as the secret key of BS_{ROM} to generate $q_S + 1$ signatures. Otherwise, $\mathcal{F}$ outputs $\{(m_i^*, (\sigma_i^*, r_i^*, s_i^*))\}_{i \in [q_S+1]}$ as forgeries. Hence, $\mathcal{F}$ can break the security of BS_{ROM} at least with the probability that $\mathcal{U}^*$ breaks the security of PBS_{ROM}.

Concurrently Secure Blind Message Signatures CBS_{ROM} based on the RSA Assumption. We use the Paillier encryption to make our blind message signatures

concurrently secure in the CRS model in the same way as [Oka06]. In particular, we change the witness indistinguishable proof of BS_{ROM}. In BS_{ROM}, a commitment of a witness is computed as $B := v_0 v_1^{h(m)} R^e \bmod n$, but in the concurrently secure scheme CBS_{ROM}, we set $g := v_1^e \bmod n$ and $B := v_0 v_1^{h(m)} g^R \bmod n$ for a random integer $R \xleftarrow{U} \mathbb{Z}_e$. This allows us to encrypt R by the Paillier encryption, and in the proof of unforgeability the reduction can obtain R by decrypting a Paillier encryption with its secret key contained in the trapdoor information of the CRS. Other processes of the reduction algorithm are the same as in [Oka06].

5.1.4 Organization of This Chapter

In Section 5.2, we introduce a definition of (partially) blind message signatures . In Section 5.3, we construct the blind message signatures secure based on the RSA assumption in the random oracle model. The blind message signatures are constructed from the digital signatures proposed in [HAO16b]. We also show that the random oracle of this blind message signatures can be replaced by a concrete hash function constructed by using indistinguishability obfuscation. In Section 5.4, we convert the proposed blind message signatures to the partially blind ones, and prove their security. In Section 5.5, we also convert in the same way as in [Oka06] the blind message signatures described in 5.3 to concurrently secure blind message signatures. In Section 5.6, we summarize the results described in this chapter.

5.2 Preliminaries

5.2.1 Partially Blind Message Signatures

In this section, we introduce the definition of partially blind message signatures. Suppose that a signer and user in advance agree on a common information when issuing partially blind message signatures. When the information info is the null string $\perp$, then the scheme defined in the following is an ordinal blind message signature scheme. Therefore, the definition of partially blind message signatures includes the definition of ordinal blind message signatures.

Definition 5.2.1 (Partially Blind Message Signatures)**.** *Partially blind message signature scheme* PBS *consists of four (interactive) algorithms (Turing machines)* $(\mathsf{KeyGen}, \mathcal{S}, \mathcal{U}, \mathsf{Verify})$.

- KeyGen *is a PPT algorithm that takes as input security parameter* λ*, and outputs a pair of public and secret keys* $(\mathsf{vk}, \mathsf{sk})$.

- *$(\mathcal{S}, \mathcal{U})$ is a pair of probabilistic interactive Turing machines. Each of them has a public information tape, private input tape, private random tape, private work tape, private output tape, public output tape, and input/output communication tape. Here, the random tape and input tape are read-only, the output tape is write-only, the private work tape can be read-write. The public input tape of $\mathcal{U}$ contains* vk *generated by* $\mathsf{KeyGen}(1^\lambda)$ *and* info*, and the public input tape of $\mathcal{S}$ contains* info*. The private input tape of $\mathcal{S}$ contains* sk*, and that of $\mathcal{U}$ contains message m. $\mathcal{U}$ and $\mathcal{S}$ engage in the signature issuing protocol and stop in polynomial time in λ. When $\mathcal{U}$ and $\mathcal{S}$ stop, the public output tape of $\mathcal{S}$ contains either* $\mathsf{completed}$ *or* $\mathsf{not\text{-}completed}$*. Similarly, the private output tape of $\mathcal{U}$ contains either $\perp$ or (m, σ).*

- Verify *is a PPT algorithm that takes as input* vk *and* $(\mathsf{info}, m, \sigma)$*, and outputs either* 0 *or* 1.

Definition 5.2.2 (Correctness)**.** *When $\mathcal{S}$ and $\mathcal{U}$ execute a signature issuing protocol for common input* $(\mathsf{vk}, \mathsf{info})$*, with the probability of at least $1 - 1/\lambda^c$ for sufficiently large n and some constant c, $\mathcal{S}$ outputs* $\mathsf{completed}$ *and $\mathcal{U}$ outputs (m, σ) such that* $\mathsf{Verify}(\mathsf{vk}, \mathsf{info}, m, \sigma) = 1$*, where the probability is taken over the randomness of* KeyGen*, $\mathcal{S}$ and $\mathcal{U}$.*

Partially Message Blindness. Without loss of generality, we assume that in the signature issuing protocol, $\mathcal{P}$, between $\mathcal{S}$ and $\mathcal{U}$, the final round is from $\mathcal{S}$ to $\mathcal{U}$. Then, let $\mathcal{P}_0$ be the sub-protocol of $\mathcal{P}$ except the final round and $\mathcal{P}_1$ be the final round. Let $\mathcal{S}$ be $(\mathcal{S}_0, \mathcal{S}_1)$ such that $\mathcal{S}_0$ executes $\mathcal{P}_0$ with $\mathcal{U}$, and $\mathcal{S}_1$ executes $\mathcal{P}_1$ with $\mathcal{U}$, where $\mathcal{S}_0$ sends string t to $\mathcal{S}_1$ after executing $\mathcal{P}_0$ (i.e., $\mathcal{S}_1$, given t, sends a value to $\mathcal{U}$).

To define message blindness for a blind message signature scheme, we consider the following experiment $\mathsf{Exp}^{blind}_{\mathcal{S}_0^*, \mathsf{PBS}}(\lambda)$ between adversary $\mathcal{S}_0^*$, (honest) $\mathcal{S}_1$ and users $\mathcal{U}_0, \mathcal{U}_1$.

1. The adversary $\mathcal{S}_0^*(\mathsf{sk}, \mathsf{info})$ outputs $\mathsf{vk}, (m_0, m_1)$.

2. Set up the input tapes of $\mathcal{U}_0$ and $\mathcal{U}_1$ as follows:

   - Choose $b \in \{0, 1\}$ at random, and put m_b and m_{1-b} on the private input tapes of $\mathcal{U}_0$ and $\mathcal{U}_1$, respectively.
   - Put $(\mathsf{info}, \mathsf{vk})$ on the public input tapes of $\mathcal{U}_0$ and $\mathcal{U}_1$.
   - Randomly choose the contents of the private random tapes of $\mathcal{U}_0$ and $\mathcal{U}_1$.

3. $\mathcal{S}_0^*$ engages in protocol $\mathcal{P}_0$ with $\mathcal{U}_0$ and $\mathcal{U}_1$, and $\mathcal{S}_1$ engages in protocol $\mathcal{P}_1$ with $\mathcal{U}_0$ and $\mathcal{U}_1$.

4. If $\mathcal{U}_0$ and $\mathcal{U}_1$ output valid signatures $(\mathsf{info}, m_b, \sigma_b)$ and $(\mathsf{info}, m_{1-b}, \sigma_{1-b})$, give 1 to $\mathcal{S}_0^*$. Give $\perp$ to $\mathcal{S}_0^*$ otherwise.

5. $\mathcal{S}_0^*$ outputs $b' \in \{0, 1\}$.

6. Output 1 if $b = b'$, and 0 otherwise.

We define the advantage of the adversary $\mathcal{S}_0^*$ in the message blindness experiment as follows:

$$\mathsf{Adv}^{blind}_{\mathcal{S}_0^*,\mathsf{PBS}}(\lambda) := 2 \cdot \Pr[\mathsf{Exp}^{blind}_{\mathcal{S}_0^*,\mathsf{PBS}}(\lambda) \to 1] - 1,$$

where the probability is taken over randomness of $\mathcal{S}_0^*, \mathcal{S}_1, \mathcal{U}_0, \mathcal{U}_1$.

Definition 5.2.3 (Partial Message Blindness)**.** *If it holds that* $\mathsf{Adv}^{n}blind_{\mathcal{S}^*,\mathsf{PBS}}(\lambda) = \mathsf{negl}(\lambda)$ *for any PPT algorithm* $\mathcal{S}_0^*$*, the partially blind message signature scheme* PBS *is partially message blind. In addition, if it holds that* $\mathsf{Adv}^{blind}_{\mathcal{S}_0^*,\mathsf{PBS}}(\lambda) = 0$ *for any algorithm* $\mathcal{S}_0^*$*, the partially blind message signature scheme* PBS *is perfect message blind.*

Unforgeability To define unforgeability of blind message signature, we consider the following experiment $\mathsf{Exp}^{unforge}_{\mathcal{U}^*,\mathsf{PBS}}(\lambda)$ between adversary $\mathcal{U}^*$ and a signer $\mathcal{S}$.

1. $(\mathsf{vk}, \mathsf{sk})$ is generated by $\mathsf{KeyGen}(1^\lambda)$, vk is put on the public input tapes of $\mathcal{U}^*$ and $\mathcal{S}$, and sk is put on the private input tape of $\mathcal{S}$.

2. For each engagement of the signature issuing protocol with $\mathcal{S}$, $\mathcal{U}^*$ outputs common information info, which is put on the public input tape of $\mathcal{S}$. Then, $\mathcal{U}^*$ engages in the signature issuing protocol with $\mathcal{S}$ in a concurrent and interleaving way.

3. For each info, let ℓ_{info} be the number of executions of the signature issuing protocol in which $\mathcal{S}$ outputs completed when info is given on the input tape. (We define $\ell_{\mathsf{info}} = 0$ for info that does not appear in the input tape of $\mathcal{S}$.) For $\mathsf{info} = \perp$, we also define $\ell_\perp$ in the same way.

4. $\mathcal{U}^*$ outputs ℓ signatures $(\mathsf{info}, m_1, \sigma_1), \ldots, (\mathsf{info}, m_\ell, \sigma_\ell)$.

5. Output 1 if the following hold, and 0 otherwise.

   - $\ell > \ell_{\mathsf{info}}$.
   - For any (i, j) $(i \neq j, i, j \in \{1, \ldots, \ell\})$, $m_i \neq m_j$.

- For any $i \in [\ell]$, $\mathsf{Verify}_{\mathsf{vk}}(\mathsf{info}, m_i, \sigma_i) = 1$.

The advantage of the adversary $\mathcal{U}^*$ is defined as

$$\mathsf{Adv}^{unforge}_{\mathcal{U}^*,\mathsf{PBS}}(\lambda) := \Pr[\mathsf{Exp}^{unforge}_{\mathcal{U}^*,\mathsf{PBS}}(\lambda) \to 1],$$

where the probability is taken over the randomness of $\mathcal{U}^*, \mathcal{S}$ and $\mathcal{G}$.

Definition 5.2.4 (Unforgeability)**.** *For any PPT algorithm $\mathcal{U}^*$, if it holds that $\mathsf{Adv}^{unforge}_{\mathcal{U}^*,\mathsf{PBS}}(\lambda) = \mathsf{negl}(\lambda)$, then a partially blind message signature scheme* PBS *is unforgeable.*

5.2.2 DCR Assumption

We introduce the decisional composite residuosity (DCR) assumption. To do that, we first define the notion of the *n-th residue*.

Definition 5.2.5 (n-th residue)**.** *An integer $z \in \mathbb{Z}$ is said to be a n-th residue modulo n^2 if there exists an integer $r \in \mathbb{Z}^*_{n^2}$ such that $z \equiv r^n \pmod{n^2}$.*

The DCR assumption is the following computational hardness assumption.

Definition 5.2.6 (DCR assumption)**.** *Let $\mathsf{nRes}(n^2)$ be the set of n-th residue modulo n^2, and* GenMod *be an algorithm that takes security parameter λ and outputs (n, p, q) where $n = pq$ and p, q are primes. Given n and $z \in \mathbb{Z}^*_{n^2}$ where $(n, p, q) \xleftarrow{R} \mathsf{GenMod}(1^\lambda)$, the DCR problem is to distinguish whether z is a random element of $\mathbb{Z}^*_{n^2}$ or $\mathsf{nRes}(n^2) \subseteq \mathbb{Z}^*_{n^2}$. The advantage of adversary $\mathcal{A}$ for the DCR problem is defined as follows.*

$$\mathsf{Adv}^{\mathsf{DCR}}_{\mathcal{A}}(\lambda) := \left| \begin{array}{l} \Pr\left[\mathcal{A}(n,z) \leftarrow 1 \left| \begin{array}{l}(n,p,q) \xleftarrow{R} \mathsf{GenMod}(1^\lambda); \\ z \xleftarrow{U} \mathsf{nRes}(n^2)\end{array}\right.\right] \\ - \Pr\left[\mathcal{A}(n,z) \leftarrow 1 \left| \begin{array}{l}(n,p,q) \xleftarrow{R} \mathsf{GenMod}(1^\lambda); \\ z \xleftarrow{U} \mathbb{Z}_{n^2}\end{array}\right.\right] \end{array} \right|$$

The DCR assumption holds if for any PPT adversary $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathsf{DCR}}_{\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$.

5.3 RSA-based Blind Message Signatures in the Random Oracle Model

In this chapter, we first build the four-move blind message signature scheme based on the RSA assumption in the random oracle model. We then show that for any

PPT adversary there exists a concrete hash function, which is constructed from indistinguishability obfuscation, that can replace the random oracle as well as in Section 4.4. One can find in Section 5.3.1 the construction in the random oracle model, and Section 5.3.2 describes how to instantiate the random oracle.

5.3.1 The Blind Message Signature Scheme BS_{ROM} in the Random Oracle Model

Construction. Our proposed blind message signature scheme $\mathsf{BS}_{ROM} = (\mathsf{KeyGen}, \mathcal{U}, \mathcal{S}, \mathsf{Verify})$ consists of four algorithms. The key generation and verification algorithms, KeyGen and Verify, are the same as those in Σ_{ROM} except that KeyGen outputs a collision-resistant hash function $h : \{0,1\}^* \to \mathbb{Z}_e$ as an additional verification key. We here describe only the signature generation protocol.

1. $\mathcal{U}$ chooses a random integer $R \xleftarrow{U} \mathbb{Z}_n^*$ and computes B as follows:

$$B := v_0 v_1^{h(m)} R^e \bmod n.$$

   $\mathcal{U}$ proves to $\mathcal{S}$ that $\mathcal{U}$ knows $(R, h(m))$ for B using the witness indistinguishable proof as follows:

   (a) $\mathcal{U}$ chooses random integer $r_1 \xleftarrow{U} \mathbb{Z}_e$ and $r_2 \xleftarrow{U} \mathbb{Z}_n^*$, and computes

$$x := v_1^{r_1} r_2^e \bmod n.$$

   $\mathcal{U}$ sends x to $\mathcal{S}$.

   (b) $\mathcal{S}$ chooses a random integer $k \xleftarrow{U} \mathbb{Z}_e$, and sends it to U.

   (c) $\mathcal{U}$ computes

$$y_1 := r_1 + kh(m) \bmod e,$$
$$y_2 := r_2 \cdot R^k \bmod n,$$

   and sends (y_1, y_2) to $\mathcal{S}$.

   (d) $\mathcal{S}$ accepts $\mathcal{U}$ if the following holds, and rejects it otherwise:

$$xB^k \equiv v_0^k v_1^{y_1} y_2^e \pmod n.$$

2. If $\mathcal{S}$ accepts $\mathcal{U}$, $\mathcal{S}$ chooses $r \xleftarrow{U} \{0,1\}^\gamma$ and $s \xleftarrow{U} \mathbb{Z}_e$ at random, computes $Y := (B \cdot H(r)^s)^d \bmod n$, and sends (Y, r, s) to $\mathcal{U}$.

3. $\mathcal{U}$ computes $\sigma := Y/R \bmod n$, and outputs (σ, r, s).

From the correctness of Σ_{ROM}, it is immediate that BS_{ROM} also holds correctness.

Security. In the following, we prove the security of BS_{ROM}.

Theorem 5.3.1. *The proposed scheme BS_{ROM} is perfectly blind.*

Proof. We show that the view of the adversary $\mathcal{S}_0^*$ is independent of $b \in \{0,1\}$ in the experiment of message blindness $\mathsf{Exp}^{blind}_{\mathcal{S}_0^*,\mathsf{BS}_{ROM}}(\lambda)$.

B, x, y_1, and y_2 are independent of the message m since $B := v_0 v_1^{h(m)} R^e \bmod n$, $x := v_1^{r_1} r_2^e \bmod n$, $y_1 := r_1 + k \cdot h(m) \bmod e$, and $y_2 := r_2 R^k \bmod n$ for $R \xleftarrow{U} \mathbb{Z}_n^*$, $r_1 \xleftarrow{U} \mathbb{Z}_e$ and $r_2 \xleftarrow{U} \mathbb{Z}_n^*$. From the independence of B, we can see that $Y := (B \cdot H(r)^s)^d \bmod n$ is also independent of m, which leads us to the signature $\sigma := Y/R \bmod n$ is also independent of m.

From the above, the view of $\mathcal{S}_0^*$ in the signature generation protocol with either $\mathcal{U}_0$ or $\mathcal{U}_1$ is independent of b in $\mathsf{Exp}^{blind}_{\mathcal{S}_0^*,\mathsf{BS}_{ROM}}(\lambda)$. Therefore, we have $\mathsf{Adv}^{blind}_{\mathcal{S}_0^*,\mathsf{BS}}(\lambda) = 0$ for any (not necessarily PPT) $\mathcal{S}_0^*$. □

To prove the unforgeability of BS_{ROM}, we introduce the definition of a synchronized run of cryptographic protocols. Clearly the synchronized run is a generalization of the parallel and sequential runs.

Definition 5.3.1 (Synchronized run)**.** *Suppose a protocol between two parties, Alice and Bob. In a round of the protocol, Alice and Bob exchange messages $a, b, c, \ldots, d$, where the first move is sent from Alice (i.e., Alice sends a and Bob returns b etc.). We now consider q rounds of the protocol execution. Here a tuple $(a_i, b_i, c_i, \ldots, d_i)$ is the exchanged messages in the i-th round ($i = 1, \ldots, q$). We say that a protocol between Alice and Bob is executed in a synchronized run of q rounds of the protocol, if the q rounds of the protocol consist of L sequential intervals and each interval, or the j-th interval ($j = 1, \ldots, L$), consists of the parallel run of q_j ($q_j \in \{1, \ldots, q\}$) rounds of the protocol, $q = q_1 + q_2 + \cdots + q_L$. Therefore, the first interval consists of: the first move from Alice is $(a_1, a_2, \ldots, a_{q_1})$, the second move from Bob is $(b_1, b_2, \ldots, b_{q_1})$, and so on. After completing the first interval, the second interval starts and consists of: the first move from Alice is $(a_{q_1+1}, a_{q_1+2}, \ldots, a_{q_1+q_2})$, the second move from Bob is $(b_{q_1+1}, b_{q_1+2}, \ldots, b_{q_1+q_2})$, and so on.*

In the following, we prove the unforgeability of the proposed blind message signature scheme BS_{ROM} in a synchronized run.

Theorem 5.3.2. *If the signature scheme Σ_{ROM} is EUF-CMA secure in the random oracle model, and h is a collision-resistant hash function, then proposed blind*

message signature BS_{ROM} *is unforgeable against an L-interval synchronized run of adversaries in the random oracle model. In particular, for any PPT adversary* $\mathcal{U}^*$*, there exists a PPT algorithm* $\mathcal{F}$ *such that*

$$\mathsf{Adv}^{unforge}_{\mathcal{U}^*,\mathsf{BS}_{ROM}} \leq \frac{8(L+1)}{L}\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{F},\Sigma_{ROM}}(\lambda) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda).$$

Proof. To prove this theorem, we give the algorithm $\mathcal{F}$ that uses the adversarial forger $\mathcal{U}^*$ for BS_{ROM} to forge signatures of Σ_{ROM}. Suppose that $\mathcal{F}$ is given $\mathsf{vk}' = (H, v_0, v_1, n, e)$ from the challenger of the EUF-CMA game for Σ_{ROM} and makes signing queries q_S times, then $\mathcal{U}^*$ and $\mathcal{F}$ run the signature generation protocol q_S times.

In the signature generation protocol with $\mathcal{B}$, the reduction $\mathcal{F}$ can use the knowledge extractor for the witness indistinguishable proof to compute the witness $(R, \hat{h} := h(m))$ (h is a collision) for B. In particular, $\mathcal{F}$ first obtains (B, x) by invoking $\mathcal{U}^*$ with inputs $\mathsf{vk} := (\mathsf{vk}', h)$ and a randomness. $\mathcal{F}$ then gives a random challenge $k \xleftarrow{U} \mathbb{Z}_e$ to $\mathcal{U}^*$, which returns a response (y_1, y_2). $\mathcal{F}$ executes the signature generation protocol with $\mathcal{U}^*$ up to the output of the commitment (B, x), and gives a new challenge $k' \xleftarrow{U} \mathbb{Z}_e$ to $\mathcal{U}^*$, which returns a new response (y'_1, y'_2). Then, it holds that

$$v_0^k v_1^{y_1} y_2^e (B^{-1})^k \equiv x \equiv v_0^{k'} v_1^{y'_1} (y'_2)^e (B^{-1})^{k'} \pmod{n}.$$

If we let $\Delta y_1 := y_1 - y'_1$, $\Delta y_2 := y_2 / y'_2 \bmod n$, $\Delta k := k - k'$, then we have

$$B^{\Delta k} \equiv v_0^{\Delta k} v_1^{\Delta y_1} (\Delta y_2)^e \pmod{n}.$$

Here, we let $z := 1/\Delta k \bmod e$ (i.e., $z\Delta k = 1 + K \cdot e$ for some $K \in \mathbb{Z}$) and $\hat{h} := z\Delta y_1 \bmod e$ (i.e., $z\Delta y_1 = \hat{h} + K' \cdot e$ for some $K' \in \mathbb{Z}$). Then we have

$$B \equiv v_0 v_1^{\hat{h}} (v_0^K v_1^{K'} B^{-K} \Delta y_2)^e \pmod{n}.$$

Therefore, $\mathcal{F}$ can compute $R := v_0^K v_1^{K'} B^{-K} \Delta y_2 \bmod n$, $\hat{h} := z\Delta y_1 \bmod e$ to obtain $(R, \hat{h})$ such that $B \equiv v_0 v_1^{\hat{h}} R^e \pmod{n}$.

Suppose that $\mathcal{F}$ and $\mathcal{U}^*$ perform an L-interval synchronized run of the blind message signature generation protocol, and the j-th interval consists of a parallel run of q_j rounds of the protocol, where $q_S = q_1 + \cdots + q_L$.

$\mathcal{F}$ works as follows in the synchronized run of the signature generation protocol with $\mathcal{U}^*$.

1. The randomness of KeyGen, $\mathcal{U}^*$, and $\mathcal{F}$ is determined at random, where random q_S challenges $(k_1, \ldots, k_{q_S})$ of $\mathcal{F}$ are also determined.

2. In the j-th interval of the synchronized run, $\mathcal{F}$ uses $(k_{Q_j+1}, \ldots, k_{Q_j+q_j})$ ($Q_j := q_1 + \cdots + q_{j-1}$ and $Q_1 := 0$) to run the signature generation protocol, and obtains the responses $(y_1, y_2)_{Q_j+1}, \ldots, (y_1, y_2)_{Q_j+q_j}$ of $\mathcal{U}^*$ for $(k_{Q_j+1}, \ldots, k_{Q_j+q_j})$.

3. $\mathcal{F}$ checks the validity of the responses from $\mathcal{U}^*$. If they are valid, then $\mathcal{F}$ rewinds the protocol to the beginning of the j-th interval (the point where $\mathcal{U}^*$ outputs the commitments $(B, x)_{Q_j+1}, \ldots, (B, x)_{Q_j+q_j}$ of the witness indistinguishable proof). Otherwise, $\mathcal{F}$ halts.

   $\mathcal{F}$ sends random challenges $(k'_{Q_j+1}, \ldots, k'_{Q_j+q_j})$ ($k_j \neq k'_j, \forall j \in [Q_j + 1, Q_j + q_j]$), and obtains the responses $(y_1, y_2)'_{Q_j+1}, \ldots, (y_1, y_2)'_{Q_j+q_j}$. $\mathcal{F}$ checks their validity. If they are valid, then $\mathcal{F}$ computes the witnesses $\{(\hat{R}_i, \hat{h}_i)\}_{i\in[Q_j+1,Q_j+q_j]}$. Otherwise, $\mathcal{F}$ returns to the beginning of this step.

4. For every $i \in [Q_j + 1, Q_j + q_j]$, $\mathcal{F}$ makes queries $\hat{h}_i$ to the signing oracle, and obtains (σ_i, r_i, s_i). $\mathcal{F}$ computes $Y_i := \hat{R}_i\sigma_i \bmod N$, and gives (Y_i, r_i, s_i) to $\mathcal{U}^*$.

5. $\mathcal{U}^*$ finally outputs q_S+1 forgeries $((m^*_1, \sigma^*_1, r^*_1, s^*_1), \ldots, (m^*_{q_S+1}, \sigma^*_{q_S+1}, r^*_{q_S+1}, s^*_{q_S+1}))$. $\mathcal{U}^*$ outputs $m^*_i, m^*_j \in \{0,1\}^*$ such that $h(m^*_i) = h(m^*_j)$ for some $i \neq j \in [q_S+1]$ with a probability of at most $\mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda)$.

6. $\mathcal{F}$ outputs $(h(m^*_j), \sigma^*_j, r^*_j, s^*_j)$ for $j \in [q_S + 1]$ such that $h(m^*_j) \notin \{h_i\}_{i\in[q_S]}$.

We can estimate the probability that $\mathcal{F}$ can forge signatures of Σ_{ROM} similar to [Oka06, Theorem 4]:

$$\mathsf{Adv}^{unforge}_{\mathcal{U}^*,\mathsf{BS}_{ROM}}(\lambda) \leq \frac{8(L+1)}{L} \cdot \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{F},\Sigma_{ROM}}(\lambda) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda).$$

□

The following corollary can immediately be proven from Theorems 5.3.2 and 4.3.2.

Corollary 5.3.1. *If the RSA assumption holds, and h is a collision-resistant hash function, then the blind message signature scheme BS_{ROM} is unforgeable against an L-interval synchronized run of adversaries in the random oracle model. In particular, for any PPT adversary $\mathcal{U}^*$, there exists a PPT algorithm $\mathcal{B}$ such that*

$$\mathsf{Adv}^{unforge}_{\mathcal{U}^*,\mathsf{BS}_{ROM}}(\lambda) \leq \frac{8(L+1)}{L}\left(\mathsf{Adv}^{\mathsf{RSA}}_{\mathcal{B}}(\lambda) + \frac{1}{\Theta(2^\lambda)} + \frac{1}{2^\lambda}\right) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda),$$

where $e = \Theta(2^\lambda)$.

5.3.2 Instantiating the Random Oracle of BS_{ROM}

Construction. A concrete hash function that can replace the random oracle of BS_{ROM} is constructed in the same way as in Section 4.4.

Security. The security of BS_{SM} can be proven in the same way as in Theorems 5.3.1 and 5.3.2.

Theorem 5.3.3. *The proposed blind message signature BS_{SM} is perfectly message blind.*

Theorem 5.3.4. *If the signature scheme BS_{SM} is EUF-CMA secure in the standard model, and h is a collision-resistant hash function, then the proposed blind message signature scheme BS_{SM} is unforgeable against an L-interval synchronized run of adversaries in the standard model. In particular, for any PPT adversary $\mathcal{U}^*$, there exists a PPT algorithm $\mathcal{F}$ such that*

$$\mathsf{Adv}^{unforge}_{\mathcal{U}^*,\mathsf{BS}_{SM}}(\lambda) \leq \frac{8(L+1)}{L} \cdot \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{F},\Sigma_{SM}}(\lambda) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda).$$

The following corollary can immediately be proven from Theorems 5.3.4 and 4.4.1.

Corollary 5.3.2. *If the RSA assumption holds, h is a collision-resistant hash function, iO is an indistinguishability obfuscator, and F_0 is a puncturable pseudorandom function, then the proposed blind message signature scheme BS_{SM} is unforgeable against an L-interval synchronized run of adversaries in the standard model. In particular, for any PPT adversary $\mathcal{U}^*$, there exists a hash function that satisfies the following: for some PPT algorithms $\mathsf{Samp}, \mathcal{D}, \mathcal{A}_1, \mathcal{A}_2, \mathcal{B}$ such that*

$$\begin{aligned}\mathsf{Adv}^{unforge}_{\mathcal{U}^*,\mathsf{BS}_{SM}}(\lambda) \leq \frac{8(L+1)}{L}\Big(&\mathsf{Adv}^{iO}_{\mathsf{Samp},\mathcal{D}}(\lambda) + \mathsf{Adv}^{\mathsf{pPRF}}_{\mathcal{A}_1,\mathcal{A}_2}(\lambda) \\ &+ \mathsf{Adv}^{\mathsf{RSA}}_{\mathcal{B}}(\lambda) + \frac{1}{\Theta(2^\lambda)} + \frac{1}{2^\lambda}\Big) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda),\end{aligned}$$

where $e = \Theta(2^\lambda)$.

5.4 The Partially Blind Message Signature Scheme PBS_{ROM}

In this chapter, we build a partially blind message signature scheme based on the RSA assumption in the random oracle model. We can also modify the proposed scheme to be secure in the standard model as in Section 5.3.2. We show in Section 5.4.1 how to construct the partially blind message signature scheme whose security is proven in Section 5.4.2.

5.4.1 Construction

The proposed partially blind message signature scheme $\mathsf{PBS}_{ROM} = (\mathsf{KeyGen}, \mathcal{U}, \mathcal{S}, \mathsf{Verify})$ consists of the following four algorithms:

- $\mathsf{KeyGen}(1^\lambda)$: Generate an RSA instance $(n, p, q, e) \xleftarrow{R} \mathsf{GenRSA}(1^\lambda)$, where e is a prime number such that $|e| = |n|$ and $\gcd(e, \phi(n)) = 1$. Compute an integer $d \in \mathbb{Z}$ such that $ed \equiv 1 \pmod{\phi(n)}$. Let H be a hash function modeled as the random oracle, and $h : \{0,1\}^* \to \mathbb{Z}_e$ be a collision-resistant hash function parameterized by λ. Let $\gamma := \gamma(\lambda)$ be a polynomial in λ. Choose random integers $v_0, v_1, v_2 \xleftarrow{U} \mathbb{Z}_n^*$. Output the verification key $\mathsf{vk} := (H, h, v_0, v_1, v_2, n, e)$ and signing key $\mathsf{sk} := d$.

- **Signature generation protocol**:

  1. $\mathcal{U}$ chooses a random integer $R \xleftarrow{U} \mathbb{Z}_n^*$ and computes B as follows:

     $$B := v_0 v_1^{h(m)} v_2^{h(\mathsf{info})} R^e \bmod n.$$

     $\mathcal{U}$ proves to $\mathcal{S}$ that $\mathcal{U}$ knows $(R, h(m), h(\mathsf{info}))$ for B using the following witness indistinguishable proof.

     (a) $\mathcal{U}$ chooses random integers $r_1, r_2 \xleftarrow{U} \mathbb{Z}_e$, and $r_3 \xleftarrow{U} \mathbb{Z}_n^*$, computes

     $$x := v_1^{r_1} v_2^{r_2} r_3^e \bmod n,$$

     and sends x to $\mathcal{S}$.

     (b) $\mathcal{S}$ chooses a random integer $k \xleftarrow{U} \mathbb{Z}_e$, and sends it to $\mathcal{U}$.

     (c) $\mathcal{U}$ computes

     $$y_1 := r_1 + k \cdot h(m) \bmod e,$$
     $$y_2 := r_2 + k \cdot h(\mathsf{info}) \bmod e,$$
     $$y_3 := r_3 R^k \bmod n,$$

     and sends (y_1, y_2, y_3) to $\mathcal{S}$.

     (d) $\mathcal{S}$ accepts $\mathcal{U}$ if the following holds, and rejects it otherwise:

     $$xB^k \equiv v_0^k v_1^{y_1} v_2^{y_2} y_3^e \pmod n.$$

  2. If $\mathcal{S}$ accepts $\mathcal{U}$, then $\mathcal{S}$ chooses $r \xleftarrow{U} \{0,1\}^\gamma$ and $s \xleftarrow{U} \mathbb{Z}_e$ randomly, computes $Y := (B \cdot H(r)^s)^d \bmod n$, and sends (Y, r, s) to $\mathcal{U}$.

  3. $\mathcal{U}$ computes $\sigma := Y/R \bmod n$ and outputs (σ, r, s).

- $\mathsf{Verify}_{\mathsf{vk}}(m, \mathsf{info}, (\sigma, r, s))$: Output 1 if the following holds, and 0 otherwise:

  $$\sigma^e \equiv v_0 v_1^{h(m)} v_2^{h(\mathsf{info})} H(r)^s \pmod n.$$

5.4.2 Security

Theorem 5.4.1. *The proposed partially blind message signature scheme* PBS_{ROM} *is perfectly message blind.*

Proof. This can be proven in the same way as in Theorem 5.3.1. □

Theorem 5.4.2. *If the blind message signature scheme* BS_{ROM} *is unforgeable against an L-interval synchronized run of adversaries in the random oracle model, and h is a collision-resistant hash function, then the proposed partially blind message signature scheme* PBS_{ROM} *is unforgeable against an L-interval synchronized run of adversaries in the random oracle model. In particular, for any PPT algorithm* $\mathcal{U}^*$*, there exists a PPT algorithm* $\mathcal{F}$ *such that*

$$\mathsf{Adv}^{unforge}_{\mathcal{U}^*,\mathsf{PBS}_{ROM}}(\lambda) \leq \mathsf{Adv}^{unforge}_{\mathcal{F},\mathsf{BS}_{ROM}}(\lambda) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda).$$

Proof. To prove this theorem, we construct an adversary $\mathcal{F}$ for BS_{ROM} that uses an adversary $\mathcal{U}^*$ that breaks unforgeability of PBS_{ROM}. Suppose that $\mathcal{F}$ is given the verification key $\mathsf{vk}' = (H, v_0, v_1, n, e)$ of BS_{ROM}, and $\mathcal{U}^*$ and $\mathcal{F}$ run the signature generation protocol at q_S times.

The adversary $\mathcal{F}$ simulates the signature generation protocol of PBS_{ROM} with $\mathcal{U}^*$ q_S times. To do this, $\mathcal{F}$ computes $v_2 := v_1^e \bmod n$ and invokes $\mathcal{U}^*$ with input $\mathsf{vk} := (H, h, v_0, v_1, v_2, n, e)$ (h is a collision resistant hash function). In the signature generation protocol, $\mathcal{U}^*$ outputs info and (B, x). $\mathcal{F}$ just sends (B, x) to $\mathcal{S}$, and receives a challenge k. $\mathcal{F}$ gives the challenge k to $\mathcal{U}^*$, receives a response (y_1, y_2, y_3), computes $y_3' := v_1^{y_2} \cdot y_3 \bmod n$, and gives (y_1, y_3') to $\mathcal{S}$. $\mathcal{S}$ always accepts $\mathcal{F}$ since the following holds for some $r_1 \in \mathbb{Z}_e, r_2 \in \mathbb{Z}_n^*$:

$$\begin{aligned} v_0^k v_1^{y_1} (y_3')^e &\equiv v_0^k v_1^{r_1 + k \cdot h(m)} (v_2^{r_2 + k \cdot h(\mathsf{info})} r_3^e R^{ek}) \\ &\equiv v_1^{r_1} v_2^{r_2} r_3^e (v_0 v_1^{h(m)} v_2^{h(\mathsf{info})})^k \\ &\equiv x B^k \pmod{n}. \end{aligned}$$

$\mathcal{F}$ just gives to $\mathcal{U}^*$ (Y, r, s) received from $\mathcal{S}$. $\mathcal{U}^*$ finally outputs $q_S + 1$ forgeries $\{\hat{m}_i^*, \mathsf{info}_i^*, (\sigma_i^*, r_i^*, s_i^*)\}_{i \in [Q_S + 1]}$, where $\mathcal{U}^*$ outputs $\hat{m}_i^*, \hat{m}_j^*$ such that $h(\hat{m}_i^*) = h(\hat{m}_j^*)$ for $i \neq j \in [q_S + 1]$ with a probability of at most $\mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda)$.

Let $m_i^* := e \cdot h(\mathsf{info}_i^*) + h(\hat{m}_i^*)$ for every $i \in [q_S + 1]$. If there exists a pair $i, j \in [q_S + 1]$ $(i \neq j)$ such that $m_i^* = m_j^*$, then $\mathcal{F}$ computes $d := (h(\mathsf{info}_i^*) - h(\mathsf{info}_j^*))/(h(\hat{m}_j^*) - h(\hat{m}_i^*))$, and uses d as a signing key for BS_{ROM} to generate $q_S + 1$ valid signatures. Otherwise (that is, if $m_i^* \neq m_j^*$ for any pair $i, j \in [q_S + 1]$ $(i \neq j)$), $\mathcal{F}$ outputs a forgery $\{m_i^*, (\sigma_i, r_i, s_i)\}_{i \in [q_S + 1]}$.

From the above, we can obtain the following inequation between the advantages of $\mathcal{U}^*$ and $\mathcal{F}$:

$$\mathsf{Adv}^{unforge}_{\mathcal{U}^*,\mathsf{PBS}_{ROM}}(\lambda) \left(1 - \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda)\right) \leq \mathsf{Adv}^{unforge}_{\mathcal{F},\Sigma_{ROM}}(\lambda).$$

Therefore, we have

$$\mathsf{Adv}^{unforge}_{\mathcal{U}^*,\mathsf{PBS}_{ROM}}(\lambda) \leq \mathsf{Adv}^{unforge}_{\mathcal{F},\Sigma_{ROM}}(\lambda) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda).$$

□

The following corollary can immediately be proven from Theorems 5.4.2 and 5.3.2.

Corollary 5.4.1. *If the RSA assumption holds, and h is a collision-resistant hash function, then the proposed blind message signature scheme* PBS_{ROM} *is unforgeable against an L-interval synchronized run of adversaries in the random oracle model. In particular, for any PPT algorithm* $\mathcal{U}^*$*, there exists a PPT algorithm* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{unforge}_{\mathcal{U}^*,\mathsf{PBS}_{ROM}}(\lambda) \leq \frac{8(L+1)}{L}\left(\mathsf{Adv}^{\mathsf{RSA}}_{\mathcal{B}}(\lambda) + \frac{1}{\Theta(2^\lambda)} + \frac{1}{2^\lambda}\right) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda),$$

where $e = \Theta(2^\lambda)$.

5.5 Concurrently Secure Blind Message Signatures CBS_{ROM}

In this chapter, we propose a concurrently secure blind message signature scheme whose security is proven from the RSA and DCR assumptions in the CRS model. We show in Section 5.5.1 the construction of CBS_{ROM}, and prove its security in Section5.5.2.

5.5.1 Construction

- $\mathsf{KeyGen}(1^\lambda)$: Generate an RSA instance $(n, p, q, e) \xleftarrow{R} \mathsf{GenRSA}(1^\lambda)$, where e is a prime number such that $|e| = |n|$ and $\gcd(e, \phi(n)) = 1$. Compute an integer $d \in \mathbb{Z}$ such that $ed \equiv 1 \pmod{\phi(n)}$. Let H be a hash function modeled as a random oracle, and $h : \{0,1\}^* \to \mathbb{Z}_e$ be a collision-resistant hash function parameterized by security parameter λ. Let $\gamma := \gamma(\lambda)$ be a polynomial in λ. Choose a random integers $v_0, v_1 \xleftarrow{U} \mathbb{Z}^*_n$. Generate a secret and public key for the Paillier encryption, (P, Q) and $(N = PQ, G)$, and a public and secret key for the trapdoor commitment Com of [Dam00] $(\mathsf{pk}_{com}, \mathsf{sk}_{com})$, where $|N| := (4 + 2c_0)|e|$ $(0 < c_0 < 1$ is a constant.). Output a verification and signing keys, $\mathsf{vk} := (H, h, v_0, v_1, n, e)$ and $\mathsf{sk} := d$, and a CRS $(N, G, \mathsf{pk}_{com})$. Here, the trapdoor for CRS is $((P, Q), \mathsf{sk}_{com})$.

- **Signature Generation Protocol:**

  1. $\mathcal{U}$ chooses random integers $R \xleftarrow{U} \mathbb{Z}_e$ and $U \xleftarrow{U} \mathbb{Z}_{N^2}$, sets $g := v_1^e \bmod n$ and $\hat{R} := g^R \bmod n$, and computes

  $$B := v_0 v_1^{h(m)} \hat{R} \bmod n,$$
  $$D := G^{h(m)+R2^K} U^N \bmod N^2,$$

  where $K := (2 + c_0)|e|$. $\mathcal{U}$ sends (B, D) to $\mathcal{S}$. $\mathcal{U}$ proves that he knows $(\hat{R}, h(m))$ for B.

  (a) $\mathcal{U}$ chooses $r_1, r_2 \xleftarrow{U} \{0,1\}^{(2+c_1)|e|}$ (c_1 is a constant $0 < c_1 < c_0 < 1$), $X \xleftarrow{U} \mathbb{Z}_{N^2}$, and a randomness r^* for Com, computes

  $$x := v^{r_1} g^{r_2} \bmod n,$$
  $$E := G^{r_1+r_2 2^K} X^N \bmod N^2,$$
  $$C := \mathsf{Com}_{\mathsf{pk}_{com}}(E, r^*),$$

  and sends (x, C) to $\mathcal{S}$.

  (b) $\mathcal{S}$ sends $k \xleftarrow{U} \mathbb{Z}_e$ to $\mathcal{U}$.

  (c) $\mathcal{U}$ computes the following for the challenge k from $\mathcal{S}$:

  $$y_1 := r_1 + k \cdot h(m),$$
  $$y_2 := r_2 + kR \bmod n,$$
  $$F := XU^k \bmod N^2.$$

  $\mathcal{U}$ sends (y_1, y_2, F, E, r^*) to $\mathcal{S}$.

  (d) $\mathcal{S}$ accepts $\mathcal{U}$ if the following holds, and rejects it otherwise.

  $$|y_1|, |y_2| \leq (2 + c_1) \cdot |e|,$$
  $$v_0^k v_1^{y_1} g^{y_2} \equiv x B^k \pmod{n},$$
  $$c = \mathsf{Com}_{\mathsf{pk}_{com}}(E, r^*),$$
  $$G^{y_1+y_2 \cdot 2^K} \cdot F^N \equiv E \cdot D^k \pmod{N^2}.$$

  2. If $\mathcal{S}$ accepts $\mathcal{U}$, then $\mathcal{S}$ chooses $r \xleftarrow{U} \{0,1\}^\gamma$ and $s \xleftarrow{U} \mathbb{Z}_e$, computes

  $$Y := (B \cdot H(r)^s)^d \bmod n,$$

  and sends (Y, r, s) to $\mathcal{U}$.

  3. $\mathcal{U}$ computes $\sigma := Y/\hat{R} \bmod n$, and outputs (σ, r, s).

5.5.2 Security

Theorem 5.5.1. *If the DCR assumption holds, then the proposed blind message signature scheme* CBS_{ROM} *is message blind against a concurrent run of adversaries.*

Proof. Paillier shows in [Pai99] that the Paillier encryption is IND-CPA secure if the DCR assumption holds. To prove this Theorem, we only have to show that CBS_{ROM} is blind if the Paillier encryption is IND-CPA secure.

We use the adversary $\mathcal{S}_0^*$ that predicts b in $\mathsf{Exp}^{blind}_{\mathsf{CBS}_{ROM},\mathcal{S}_0^*}$ with noticeable probability ϵ to construct an algorithm $\mathcal{B}$ that breaks the IND-CPA security of the Paillier encryption with noticeable advantage.

1. Given a public key (N, G) for the Paillier encryption, $\mathcal{B}$ generates a public and secret key for Com, $(\mathsf{pk}_{com}, \mathsf{sk}_{com})$, and gives to $\mathcal{S}_0^*$ $((N, G), \mathsf{pk}_{com})$ as the CRS.

2. $\mathcal{S}_0^*$ gives to $\mathcal{B}$ the verification key $\mathsf{vk} = (H, h, v_0, v_1, n, e)$ and two messages $m_0, m_1 \in \{0,1\}^*$.

3. $\mathcal{B}$ chooses random integers $R_0, R_1 \xleftarrow{U} \mathbb{Z}_e$, and sets $M_0 := h(m_0) + R_0 \cdot 2^K$, $M_1 := h(m_1) + R_1 \cdot 2^K$.

4. When $\mathcal{B}$ gives (M_0, M_1) to the challenger in the IND-CPA game for the Paillier encryption, the challenger chooses $\beta \xleftarrow{U} \{0,1\}$ and returns $D := G^{M_\beta} \cdot A_0^N \bmod N^2$ to $\mathcal{B}$.

5. $\mathcal{B}$ computes $g := v_1^e \bmod n$ and $\hat{R}_i := g^{R_i} \bmod n$ $(i = 0, 1)$, chooses $b \xleftarrow{U} \{0,1\}$ and $A_1 \xleftarrow{U} \mathbb{Z}_{N^2}$, and computes

$$B_0 := v_0 v_1^{h(m_0)} \hat{R}_0 \bmod n, D_0 := D,$$
$$B_1 := v_0 v_1^{h(m_1)} \hat{R}_1 \bmod n, D_1 := G^{M_1} \cdot A_1^N \bmod N^2.$$

   $\mathcal{B}$ then invokes the signature generation protocols by sending (B_b, D_b) to $\mathcal{S}_0^*$ as $\mathcal{U}_0$ and (B_{1-b}, D_{1-b}) as $\mathcal{U}_1$.

6. $\mathcal{B}$ proves as $\mathcal{U}_i$ $(i = 0, 1)$ to $\mathcal{S}^*$ that he knows $(h(m_i), \hat{R}_i)$.

7. The pair (B_1, D_1) sent by $\mathcal{B}$ as $\mathcal{U}_{1-b}$ is computed as well as the real one by $\mathcal{U}_{1-b}$. The computation for B_0 by $\mathcal{B}$ as $\mathcal{U}_b$ is equal to the real one by $\mathcal{U}_b$, but the one for D_0 is not. $\mathcal{B}$ chooses a random integer $F_0 \xleftarrow{U} \mathbb{Z}_{N^2}$ after obtaining the challenge k from $\mathcal{S}^*$, and computes an Paillier encryption $E_0 := G^{y_1 + y_2 \cdot 2^K} \cdot F_0^N / D_0^k \bmod N^2$ as accepted by $\mathcal{S}^*$, where the pair (y_1, y_2) is determined by the process regarding B_0. $\mathcal{B}$ uses the trapdoor sk to open C to (E_0, r^*), sends $(y_1, y_2, F_0, E_0, r^*)$ to $\mathcal{S}^*$.

8. All the processes after the above are the same as the real ones.

9. After the signature generation protocols for $\mathcal{U}_0$ and $\mathcal{U}_1$ finish, $\mathcal{B}$ checks whether or not the obtained signatures are valid, and outputs $\perp$ if one of the two signatures is not valid. If $\mathcal{B}$ accepts both of the signatures, and then gives 1 to $\mathcal{S}_0^*$. $\mathcal{B}$ obtains the output b' from $\mathcal{S}_0^*$, and outputs $\beta' = 0$ if $b = b'$, and $\beta' = 1$ otherwise.

If $\beta = 0$, then the distribution of the view of $\mathcal{S}_0^*$ regarding $\mathcal{U}_0$ and $\mathcal{U}_1$ in the above protocol execution of $\mathcal{B}$ is the same as that for the blindness experiment for CBS_{ROM} scheme. So, we have $\Pr[b = b' \mid \beta = 0] = (\epsilon + 1)/2$.

If $\beta = 1$, the distribution of D_0 and D_1 are the same. The distributions of the view of $\mathcal{S}_0^*$ regarding B_0 and B_1 in the signature generation protocol are statistically indistinguishable, and the distribution of the view of $\mathcal{S}_0^*$ on the fake protocol regarding D_0 and the real protocol regarding D_1 are also statistically indistinguishable. Whether $\mathcal{B}$ gives $\mathcal{S}_0^*$ $\perp$ or two valid signatures depends only on whether the response (Y, r, s) of $\mathcal{S}_0^*$ satisfies $Y^e \equiv B_i \cdot H(r)^s \pmod{n}$, but does not depend on the value of b since the distribution of B_0 and B_1 are equivalent. From the above, we have $|\Pr[b \neq b' \mid \beta = 1] - 1/2| < \mu$ for negligible μ in λ.

So, we have

$$\begin{aligned}
\Pr[\beta = \beta'] &= \Pr[\beta = \beta' = 0 \vee \beta = \beta' = 1] \\
&= \Pr[\beta' = 0 \mid \beta = 0] \cdot \Pr[\beta = 0] \\
&\quad + \Pr[\beta' = 1 \mid \beta = 1] \cdot \Pr[\beta = 1] \\
&= \frac{1}{2}(\Pr[b = b' \mid \beta = 0] + \Pr[b \neq b' \mid \beta = 1]) \\
&> \frac{1}{2}\left(\frac{1+\epsilon}{2} + \frac{1}{2} - \mu\right) \\
&= \frac{1}{2} + \frac{\epsilon}{4} + \frac{\mu}{2}.
\end{aligned}$$

Therefore, the probability that $\mathcal{B}$ breaks the IND-CPA security (namely the advantage of $\mathcal{B}$ in the IND-CPA game), $2 \cdot \Pr[\beta = \beta'] - 1$, is $\epsilon/2 - \mu$, which is non-negligible in λ. □

Theorem 5.5.2. *If the underlying signature scheme Σ_{ROM} is EUF-CMA secure and the trapdoor commitment* Com *satisfies the binding condition, then the proposed signature scheme CBS_{ROM} is unforgeable against a concurrent run of adversaries in the CRS model. In particular, for any PPT adversary $\mathcal{U}^*$, there exists a PPT algorithm $\mathcal{F}$ such that*

$$\begin{aligned}
\mathsf{Adv}^{unforge}_{\mathcal{U}^*,\mathsf{CBS}_{ROM}}(\lambda) \leq \frac{8(q_S+1)}{q_S} \cdot \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{F},\Sigma_{ROM}}(\lambda) \\
+ \mathsf{Adv}^{binding}_{\mathcal{U}^*,\mathsf{Com}}(\lambda) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda),
\end{aligned}$$

where q_S is the number of queries to the signing oracle by $\mathcal{F}$ (or the number of rounds of the signature generation protocol by $\mathcal{U}^*$), and $\mathsf{Adv}^{binding}_{\mathcal{U}^*,\mathsf{Com}}(\lambda)$ is the probability that $\mathcal{U}^*$ breaks the binding condition of Com.

Proof. We use the adversary $\mathcal{U}^*$ that forges signatures of the blind message signature scheme CBS_{ROM} with non-negligible probability to construct an algorithm that forges signatures of the signature scheme Σ_{ROM}. Let $\mathsf{Adv}^{binding}_{\mathcal{U}^*,\mathsf{Com}}(\lambda)$ be the probability that $\mathcal{U}^*$ breaks the binding condition of the underlying trapdoor commitment Com. By the theorem statement, we can assume that $\mathsf{Adv}^{binding}_{\mathcal{U}^*,\mathsf{Com}}(\lambda) = \mathsf{negl}(\lambda)$.

1. Given a verification key (H, v_0, v_1, n, e) of Σ_{ROM}, $\mathcal{F}$ chooses a secret and public key for the Paillier encryption, (P, Q) and $(N = PQ, G)$, and a secret and public key for Com, $(\mathsf{pk}_{com}, \mathsf{sk}_{com})$. Let h be a collision-resistant hash function determined by security parameter λ. $\mathcal{F}$ gives $\mathcal{U}^*$ (H, h, v_0, v_1, n, e) as the verification key and $(N, G, \mathsf{pk}_{com})$ as the CRS model.

2. $\mathcal{F}$ executes a signature generation protocol with $\mathcal{U}^*$. Receiving the j-th signature generation request, (B, D) and (x, C), $\mathcal{F}$ returns a random challenge $k \xleftarrow{U} \mathbb{Z}_e$ to $\mathcal{U}^*$. $\mathcal{F}$ receives a response (y_1, y_2, F, E, r^*) from $\mathcal{U}^*$ for the challenge k. $\mathcal{F}$ checks whether or not the following hold:

$$
\begin{aligned}
&|y_i| \leq (2 + c_1)|e| \ (i = 1, 2),\\
&C = \mathsf{Com}_{\mathsf{pk}_{com}}(E, r^*),\\
&xB^k \equiv v_0^k v_1^{y_1} g^{y_2} \pmod{n},\\
&G^{y_1 + y_2 2^K} F^N \equiv ED^k \pmod{N^2}.
\end{aligned}
$$

   If they do not hold, $\mathcal{F}$ writes not-completed into his public output tape, and halts.

3. If the above relations hold, $\mathcal{F}$ chooses random $k' \xleftarrow{U} \mathbb{Z}_e$, and uses the secret key for the Paillier encryption, (P, Q), to decrypt $ED^{k'} \bmod N^2$ and obtain ξ. Here, ξ satisfies $G^{\xi} U^N \equiv ED^{k'} \pmod{N}^2$ for some $U \in \mathbb{Z}_{N^2}$. If there exists (y'_1, y'_2) such that $\xi = y'_1 + y'_2 2^K$ and $|y'_i| \leq (2 + c_1)|e|$ $(i = 1, 2)$, $\mathcal{F}$ checks whether or not $xB^{k'} \equiv v_0^{k'} v_1^{y'_1} g^{y'_2} \pmod{n}$ holds. If it does not hold, return to the beginning of this step. The number of iterations of this step is bounded by a polynomial in λ. If it holds, the following equation also holds:

$$
v_0^k v_1^{y_1} g^{y_2} (B^{-1})^k \equiv x \equiv v_0^{k'} v_1^{y'_1} g^{y'_2} (B^{-1})^{k'} \pmod{n}.
$$

   Let $\Delta k := k - k'$ and $\Delta y_1 := y_1 - y'_1$ and $\Delta y_2 := y_2 - y'_2$, then we have $B^{\Delta k} \equiv v_0^{\Delta k} v_1^{\Delta y_1} g^{\Delta y_2} \pmod{n}$. Hence if we let $\Delta y_1 = Le + \hat{h}$ (namely, $\hat{h} := \Delta y_1 \bmod e$

for some $L, \hat{h} \in \mathbb{Z}$, we have

$$B \equiv v_0 v_1^{\hat{h}} (v_0^{\Delta k-1} v_1^{Le} g^{\Delta y_2} (B^{-1})^{\Delta k-1}) \pmod{n}.$$

$\mathcal{F}$ computes the pair $(\hat{h} := \Delta y_1 \bmod e, \hat{R} := v_0^{\Delta k-1} v_1^{Le} g^{\Delta y_2} (B^{-1})^{\Delta k-1} \bmod n)$, which satisfies $B \equiv v_0 v_1^{\hat{h}} \hat{R} \pmod{n}$.

4. $\mathcal{F}$ queries $\hat{h}$ to the signing oracle and obtains (σ, r, s), computes $Y := \sigma \hat{R} \bmod n$, and sends (Y, r, s) to $\mathcal{U}^*$.

5. If the whole signing procedure of the q_S rounds is completed successfully, $\mathcal{U}^*$ outputs $q_S + 1$ forgeries $\{(m_j^*, (\sigma_j^*, r_j^*, s_j^*))\}_{j \in [q_S+1]}$. Here, the probability that $\mathcal{U}^*$ outputs $m_i^*, m_j^* \in \{0,1\}^*$ such that $h(m_i^*) = h(m_j^*)$ for some $i \neq j \in [q_S + 1]$ is at most $\mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda)$.

6. $\mathcal{F}$ outputs $(h(m_j^*), (\sigma_j^*, r_j^*, s_j^*))$ for $j \in [q_S + 1]$ such that $h(m_j^*)$ is not queried to the signing oracle.

We can obtain the probability that $\mathcal{F}$ forges signatures of Σ_{ROM} in a similar way to [Oka06, Theorem 8], and so have

$$\mathsf{Adv}^{unforge}_{\mathcal{U}^*, \mathsf{CBS}_{ROM}}(\lambda) \leq \frac{8(q_S+1)}{q_S} \cdot \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{F}, \Sigma_{ROM}}(\lambda) + \mathsf{Adv}^{binding}_{\mathcal{U}^*, \mathsf{Com}}(\lambda) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda).$$

□

Theorem 5.5.2 and Theorem 4.3.2 immediately lead to the following corollary.

Corollary 5.5.1. *If the RSA assumption holds, and h is a collision-resistant hash function, then the blind message signature scheme* CBS_{ROM} *is unforgeable against a concurrent run of adversaries in the CRS model. In particular, for any PPT adversary* $\mathcal{U}^*$*, there exists a PPT algorithm* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{unforge}_{\mathcal{U}^*, \mathsf{CBS}_{ROM}}(\lambda) \leq \frac{8(q_S+1)}{q_S} \cdot \left(\mathsf{Adv}^{\mathsf{RSA}}_{\mathcal{B}}(\lambda) + \frac{1}{\Theta(2^\lambda)} + \frac{1}{2^\lambda} \right) + \mathsf{Adv}^{binding}_{\mathcal{U}^*, \mathsf{Com}}(\lambda) + \mathsf{Adv}^{\mathsf{CRHF}}_{\mathcal{U}^*}(\lambda),$$

where $e = \Theta(2^\lambda)$.

5.6 Conclusion of This Chapter

In this chapter, we proposed three types of four-move efficient blind message signature secure based on the RSA assumption, particularly blind message signatures, partially blind message signatures, and concurrently secure blind message signatures.

The proposed (partially) blind message signatures are the most efficient blind message signatures secure under the RSA assumption in the random oracle model. Previously, the most efficient blind signature scheme in the random oracle model is the one proposed by Abe that is from the discrete log like assumptions, and there are no known constructions that is secure under the RSA assumption and its efficiency is comparable to the Abe's blind signatures. Our proposed blind message signatures are the first whose efficiency is comparable to the Abe's blind signatures. The blind signatures obtained by applying the transformation [Poi98] to the Okamoto-Guillou-Quisquater blind signatures [PS96] is also secure under the RSA assumption, but this protocol is a five-move protocol.

We first constructed blind message signatures from the signatures proposed in [HAO16b], so this blind message signature protocol is secure in the random oracle model. Since the existence of the random oracle is a significantly strong assumption, there are known results that show artificial constructions secure in the random oracle model but not in the case where the random oracle is replaced with a concrete hash function. Similar to the signatures of [HAO16b], the random oracle of our blind message signatures can also be replaced by a concrete hash function depending on the power of the adversaries. We then showed that it is easily to construct a partially blind message signatures from our blind message signatures, and make our blind message signatures concurrently secure by using the Paillier encryption as in [Oka06].

Chapter 6

Conclusion

Cryptography focuses on securing communications in the presence of an adversary. To communicate securely with each other, the data from a sender are required to be concealed from the adversary, and the receiver of the data must be able to verify that the received data are not tampered with and certainly sent from the sender. Two important objectives of cryptography are to construct cryptosystemes that *hide the secret data* and *authenticate the received data*.

The first objective is to construct a cryptosystem that hides secret information from an adversary. Suppose that Alice wants to send her secret message to Bob over an insecure communication channel. Then the adversary, Eve, may wiretap the message and know what Alice told to him. Public key encryption (PKE) is a way to deal with this problem. The advanced encryption method is a cryptosystem that provides a certain functionality in addition to PKE. A typical example of this method is FHE, which is a variant of PKE and allows us to evaluate any function over encrypted data using only public information. The second objective is to construct a way to authenticate data. We want to ensure that the received data are from the sender, or the data are not tampered. Suppose that Bob receives a message addressed from Alice. Then how does Bob ensure that the received message is the same as the intended message from Alice? In particular, how does he ensure that the received message is not tampered with by an adversary? Digital Signatures are a way to achive this.

The importance of the above cryptosystems will grow in the future as more everyday tasks, processes, and communications are computationalized. The main goals in cryptography are to contrive *efficient* ways for achieving such important cryptosystems, and provide a *theoretical proof* for ensuring their security. Constructing more efficient cryptosystems that implement important cryptographic functions allows us to apply their functions in wider areas. To prove formally the security of a cryptosystem, we construct a reduction (algorithm) that uses an adversary against the cryptosystem to break the assumption (namely, to solve the

problem assumed to be intractable in the assumption). In cryptography, the security assumptions are classified into the following five types of assumptions: the standard assumptions, the non-standard, falsifiable, non-interactive, and static-size assumptions, the falsifiable and non-interactive, but dynamic-size (q-type) assumptions, the falsifiable and interactive assumptions, and the unfalsifiable assumptions. The most desirable assumptions are the standard assumptions.

To implant cryptography as a foundation block of information security that supports our networked society, cryptography must provides important cryptosystems that can be implemented efficiently with their security guaranteed theoretically. For achieving cryptosystems that become one of such foundations, this thesis proposed the efficient and theoretically secure schemes (under the standard assumptions) to implement FHE and signatures (digital signatures and blind signatures) that are now considered as the most important functions.

Chapter 3: Efficient FHE based on the LWE Assumption. FHE allows us to evaluate any function over encrypted data by using only public information. A natural example of its applications is searching on encrypted data. Since the breakthrough work by Gentry [Gen09a, Gen09b], many different varieties of FHE have been proposed [DGHV10, BV11a, BV11b, BGV12, Bra12, GSW13, CLT14]. To date, the fastest (and simplest) FHE based on the *standard* assumption is the one proposed by Gentry, Sahai, and Waters [GSW13] (hereafter, referred to as GSW-FHE). However, it is required to take heavy cost for evaluating a large number of ciphertexts. A way to deal with this issue is to *pack* multiple messages into one ciphertext. Packing messages allows us to apply SIMD homomorphic operations to all encrypted messages. In the case where a remote server stores encrypted data and we want to retrieve certain data from that server, we first apply the equality function to every encrypted data set. If the stored data have been packed into one ciphertext, we can retrieve the desired data by only one homomorphic evaluation of the equality function.

In Chapter 3, we proposed FHE with more efficient homomorphic operation algorithms than the previous FHE based on the standard assumptions. In particular, we constructed an FHE scheme that encrypts *matrices* and supports homomorphic *matrix* addition and multiplication. Homomorphic operations of our FHE are just the matrix multiplication between two square matrices, so its time complexity is estimated based on the best complexity of the multiplication algorithm of two square matrices. It is expected that the complexity of our FHE will decrease as the studies for the matrix multiplication algorithm progress. Our FHE is a natural extension of SIMD FHE and thus supports more complicated homomorphic operations. We optimized the bootstrapping procedure of Alperin-Sheriff and Peikert (CRYPTO 2014) by applying the proposed FHE. Our optimization decreases the lattice approximation factor from $\tilde{O}(n^3)$ to $\tilde{O}(n^{2.5})$. By taking a lattice dimen-

sion as a larger polynomial in a security parameter, we can also obtain the same approximation factor as the best known one of standard lattice-based public-key encryption *without* successive dimension-modulus reduction, which was essential for achieving the best factor in prior studies on bootstrapping of standard lattice-based FHE. Our construction is an extension of the GSW-FHE scheme [GSW13], which has a great influence on the construction ideas of some cryptosystems based on the LWE such as fully homomorphic signatures [GVW15, FMNP16], attribute based encryption [BGG+14, BV16, BCTW16], and multilinear maps [GGH15]. Hence, the idea behind our FHE may also influence future cryptographic constructions based on the LWE assumption.

Chapter 4: Efficient Digital Signatures based on the RSA Assumption. There is a gap, which is called *reduction efficiency*, between the hardnesses of breaking a cryptosystem and solving a security problem. If a security reduction is tight, breaking the cryptosystem is as hard as solving the underlying problem. Hence, if we can prove the security of a cryptosystem with highly efficient security reduction, the cryptosystem can be implemented with smaller parameter settings (that is, a smaller key size). We particularly focus on tightly secure digital signatures in the *random oracle* model.

The random oracle model, which was first introduced by Bellare and Rogaway in 1993 [BR93], is an idealized paradigm in which a hash function is viewed as an oracle that outputs a random value for every input query. A security proof for a cryptographic scheme in the random oracle model does not mean that it is secure in the real world, but it provides some kind of security guarantee, and it is still important in a practical sense to prove the security in the random oracle model.

In Chapter 4, we showed tightly secure efficient digital signatures based on the RSA assumption, which is thought as the most reliable assumption. Our new RSA-based signature scheme is proven secure in the random oracle model. The number of random oracles used in this scheme is less than that of all previous schemes with the same security guarantee, so our signatures are simpler than the previous tightly secure signatures from the RSA assumption. We then showed that for any PPT adversary there exists a concrete hash function from indistinguishability obfuscation that can replace the random oracle while maintaining security. The same statement can be proven for the Coron's signatures. To prove the security of our signatures, we introduced a new proof technique called the $\alpha - \beta$ *hiding technique*. This technique relies on the mathematics of the RSA, so we believe that this technique will become a useful tool to prove security of other cryptosystems based on the RSA assumption. Our proposed signatures are tightly secure as well as the PSS [BR96] that is a foundation of PKCS #1 standard [RSA93], and have simpler construction than the PSS since the number of random oracles (implemented by a hash function) is optimal. The simplicity of the construction

of our signature scheme also leads to lower implementation costs of the secure digital signatures. Therefore, our signatures may be an effective alternative of the PSS.

Chapter 5: Efficient Blind Message Signatures based on the RSA Assumption. Blind signatures are a variant of digital signatures first introduced by Chaum [Cha82]. They are a cryptographic protocol between two parties (user and signer) in which the user requests a signature for his message and obtains a signature from the signer, where the signed message is hidden from the signer (blindness), and the number of signatures generated by the user is not larger than the number of runs of the blind signature protocol (unforgeability). In particular, because of the blindness, blind signatures have an important role in applications such as the electronic cash and electronic voting.

The Chaum's blind signatures [Cha82] from the RSA signatures [RSA78] was not provable secure. In [BNPS03], Bellare et al. showed that the Chaum's blind signature scheme is provable secure, but the underlying assumption is not standard. Secure blind signatures from the standard assumptions in the random oracle model were proposed in [PS96, Poi98, AO00, Abe01, AO01], the most efficient blind signatures among these are the ones by Abe [Abe01].

In Chapter 5, we introduce a new notion *blind message signatures*, which has the following features. A signer $\mathcal{S}$ executes a blind signature protocol, $\mathcal{P}$, with an user $\mathcal{U}$ and $\mathcal{S}$ is divided into two parts,$\mathcal{S}_0$ and $\mathcal{S}_1$. $\mathcal{S}_0$ accepts a request from the user $\mathcal{U}$ and knows the identity of $\mathcal{U}$. $\mathcal{S}_0$ then runs the sub-protocol of $\mathcal{P}$ with $\mathcal{U}$ (say $\mathcal{P}_0$) which is $\mathcal{P}$ excepting the final round. $\mathcal{S}_1$ executes the final round of $\mathcal{P}$ (say $\mathcal{P}_1$), i.e., $\mathcal{S}_1$ just sends a value to $\mathcal{U}$. Here, unless $\mathcal{S}_0$ and $\mathcal{S}_1$ collaborate, the protocol satisfies the requirements of blind signatures. Message m is hidden before use $\mathcal{U}$ releases the message m with signature σ even if $\mathcal{S}_0$ and $\mathcal{S}_1$ collude.

we constructed the first efficient blind message signatures from the RSA assumption in the random oracle model. We also showed that for any PPT adversary there exists a concrete hash function from indistinguishability obfuscation that directly replaces the random oracle while maintaining the security. From our blind message signatures, we can derive partially blind message signatures and concurrently secure blind message signatures from the RSA assumption (the concurrently secure variant requires an additional but also standard assumption). The key generation and verification algorithms are the same as the signature scheme proposed in Chapter 4. If the proposed digital signature is implemented into some information systems instead of the PSS, we can use the signing and verification key, and also sign and verify signatures without changing the algorithms.

Conclusion of This Thesis. All of the cryptosystems proposed in this thesis are proven secure under the standard assumptions, and they are more efficient than

the previous constructions under the same security assumptions. In addition, the cryptographic functions implemented in our cryptosystems (encryption, advanced encryption, and authentication) are considered as the central and important functions in cryptography. We hope that this thesis will contribute to establishing a foundation for information security.

Acknowledgements

I express my sincere thanks to my supervisors, Tatsuaki Okamoto and Masayuki Abe for their helpful discussions and thought-provoking advice. This thesis would not have been possible without their support. I also thank Toru Ishida, Yasuo Okabe, and Yoshimasa Nakamura for their participation as members of my thesis committee and for providing me useful advice.

I would like to thank to my advisors when I was in Kyoto University, Shigeo Matsubara and Keishi Tajima for their useful comments and suggestions. Their comments always elicited deep consideration of the practical use of my studies.

I thank all the members of the Okamoto-Abe laboratory. I have received a lot of inputs from them and could have a great time with them. I also thank all the members of the Algorithm Research and Development group in Mitsubishi Electric for their continuous encouragement and support without which I would not have been able to finish this thesis.

Finally, I wish to thank my family for their supports and encouragements.

Bibliography

[Abe01] Masayuki Abe. A Secure Three-move Blind Signature Scheme for Polynomially Many Signatures. In *Advances in Cryptology - EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 135–150, 2001.

[AF96] Masayuki Abe and Eiichiro Fujisaki. How to Date Blind Signatures. In *Advances in Cryptology - ASIACRYPT '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 244–251, 1996.

[Ajt96] Miklós Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 99–108, 1996.

[AO00] Masayuki Abe and Tatsuaki Okamoto. Provably Secure Partially Blind Signatures. In *Advances in Cryptology - CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 271–286, 2000.

[AO01] Masayuki Abe and Miyako Ohkubo. Provably Secure Fair Blind Signatures with Tight Revocation. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 583–601, 2001.

[AP14] Jacob Alperin-Sheriff and Chris Peikert. Faster Bootstrapping with Polynomial Error. In *Advances in Cryptology - CRYPTO 2014*, volume 8616 of *Lecture Notes in Computer Science*, pages 297–314, 2014.

[Bar86] David A. Mix Barrington. Bounded-Width Polynomial-Size Branching Programs Recognize Exactly Those Languages in NC^1. In *Proceedings of the 18th Annual ACM Symposium on the Theory of Computing*, pages 1–5, 1986.

[Bar10] Boaz Barak. Cryptography course - Lecture Notes, COS 433. Princeton University, Computer Science Department, 2010. Available at `http://www.cs.princeton.edu/courses/archive/spring10/cos433`.

[BB04a] Dan Boneh and Xavier Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238, 2004.

[BB04b] Dan Boneh and Xavier Boyen. Short Signatures Without Random Oracles. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73, 2004.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical Identity Based Encryption with Constant Size Ciphertext. In *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456, 2005.

[BBP04] Mihir Bellare, Alexandora Boldyreva, and Adriana Palacio. An Uninstantiable Random-Oracle-Model Scheme for a Hybrid-Encryption Problem. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 171–188, 2004.

[BCTW16] Zvika Brakerski, David Cash, Rotem Tsabary, and Hoeteck Wee. Targeted Homomorphic Attribute Based Encryption. In *Theory of Cryptography*, volume 9986 of *Lecture Notes in Computer Science*, pages 330–360, 2016.

[BFKL94] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J. Lipton. Cryptographic Primitives Based on Hard Learning Problems. In *Advances in Cryptology - CRYPTO 1993*, volume 773 of *Lecture Notes in Computer Science*, pages 278–291, 1994.

[BFPV11] Olivier Blazy, George Fuchsbauer, David Pointcheval, and Damien Vergnaud. Signatures on Randomizable Ciphertexts. In *Public Key Cryptography - PKC 2011*, volume 6571 of *Lecture Notes in Computer Science*, pages 95–112, 2011.

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully Key-Homomorphic Encryption, Arithmetic

Circuit ABE, and Compact Garbled Circuits. In *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 535–556, 2014.

[BGH13] Zvika Brakerski, Craig Gentry, and Shai Halevi. Packed Ciphertexts in LWE-based Homomorphic Encryption. In *Public Key Cryptography - PKC 2013*, volume 7778 of *Lecture Notes in Computer Science*, pages 1–13, 2013.

[BGI+01] Boaz Barak, Oded Goldreich, Russel Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (Im)possibility of Obfuscating Programs. In *Advances in Cryptology - CRYPTO 2001*, pages 1–18, 2001.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) Fully Homomorphic Encryption without Bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 309–325, 2012.

[BNPS03] Mihir Bellare, Chanathip Namprepre, David Pointcheval, and Michael Semanko. The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme. *Journal of Cryptology*, 16(3):185–215, 2003.

[BPV12] Olivier Blazy, David Pointcheval, and Damien Vergnaud. Compact Round-Optimal Partially-Blind Signatures. In *Security and Cryptography for Networks*, pages 95–112, 2012.

[BR93] Mihir Bellare and Philip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Comunications Security*, pages 62–73, 1993.

[BR96] Mihir Bellare and Philip Rogaway. The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416, 1996.

[Bra12] Zvika Brakerski. Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In *Advances in Cryptology - CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886, 2012.

[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. In *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 97–106, 2011.

[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. Fully Homomorphic Encryption from Ring-LWE and Security for Key Depedent Messages. In *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524, 2011.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Lattice-Based FHE as Secure as PKE. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, pages 1–12, 2014.

[BV16] Zvika Brakerski and Vinod Vaikuntanathan. Circuit-ABE from LWE: Unbounded Attributes and Semi-Adaptive Security. In *Advances in Cryptology - CRYPTO 2016*, volume 9816 of *Lecture Notes in Computer Science*, pages 363–384, 2016.

[CCK+13] Jung Hee Cheon, Jean-Sébastian Coron, Jinsu Kim, Moon Sung Lee, Tancrède Lepoint, Mehdi Tibouchi, and Aaram Yun. Batch Fully Homomorphic Encryption over the Integers. In *Advances in Cryptology - EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 315–335, 2013.

[CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster Fully Homomorphic Encryption: Bootstrapping in less that 0.1 Seconds. In *Advances in Cryptology - ASIACRYPT 2016*, volume 10032 of *Lecture Notes in Computer Science*, pages 3–33, 2016.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The Random Oracle Methodology, revisited. In *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing*, pages 209–218, 1998.

[Cha82] David Chaum. Blind Signatures for Untraceable Payments. In *Advances in Cryptology: Proceedings of CRYPTO '82*, pages 199–203, 1982.

[CLT13] Jean-Sébastian Coron, Tancrède Lepoint, and Mehdi Tibouchi. Practical Multilinear Maps over the Integers. In *Advances in Cryptology - CRYPTO 2013*, volume 8042 of *Lecture Notes in Computer Science*, pages 476–493, 2013.

[CLT14] Jean-Sébastian Coron, Tancrède Lepoint, and Mehdi Tibouchi. Scale-Invariant Fully Homomorphic Encryption over the Integers. In *Public Key Cryptography - PKC 2014*, volume 8383 of *Lecture Notes in Computer Science*, pages 311–328, 2014.

[CMS99] Christian Cachin, Silvio Micali, and Markus Stadler. Computationally Private Information Retrieval. In *Advances in Cryptology - EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 402–414, 1999.

[Cor02] Jean-Sébastian Coron. Optimal Security Proofs for PSS and Ohter Signature Schemes. In *Advances in Cryptology - EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 272–287, 2002.

[CS00] Ronald Cramer and Victor Shoup. Signature Schemes based on the Strong RSA Assumption. *ACM Transactions on Information and System Security*, 3(3):161–185, 2000.

[Dam91] Ivan Damgård. Towards Practical Public-Key Cryptosystems Provably -Secure Against Chosen-Ciphertext Attacks. In *Advances in Cryptology - CRYPTO 1991*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456, 1991.

[Dam00] Ivan Damgård. Efficient Concurrent Zero-Knowledge in the Auxiliary String Model. In *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 418–430, 2000.

[DGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully Homomorphic Encryption over the Integers. In *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43, 2010.

[DH76] Whitfield Diffie and Martin Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DM15] Léo Ducas and Daniele Micciancio. FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second. In *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 617–640, 2015.

[FHPS13] Eduarda S. V. Freire, Dennis Hofheinz, Kenneth G. Paterson, and Christoph Striecks. Programmable Hash Functions in the Multilinear Setting. In *Advances in Cryptology - CRYPTO 2013*, volume 8043 of *Lecture Notes in Computer Science*, pages 513–530, 2013.

[Fis06] Mark Fischlin. Round-Optimal Composable Blind Signatures in the Common Reference String Model. In *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 60–77, 2006.

[FMNP16] Dario Fiore, Aikaterini Mitrokotsa, Luca Nizzardo, and Elena Pagnin. Multi-Key Homomorphic Authenticators. In *Advances in Cryptology - ASIACRYPT 2016*, volume 10032 of *Lecture Notes in Computer Science*, pages 499–530, 2016.

[Gen09a] Craig Gentry. *A FULLY HOMOMORPHIC ENCRYPTION SCHEME*. PhD thesis, Stanford University, Available at `http://crypto.stanford.edu/craig`, 2009.

[Gen09b] Craig Gentry. Fully Homomorphic Encryption using Ideal Lattices. In *Proceedings of the 41th Annual ACM Symposium on the Theory of Computing*, pages 169–178, 2009.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate Multilinear Maps from Ideal Lattices. In *Advances in Cryptology - EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Computer Science*, pages 1–17, 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate Indistinguishability Obfuscation and Functional Encryption for all circuits. In *Proceeedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 40–49, 2013.

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-Induced Multilinear Maps from Lattices. In *Theory of Cryptography*, volume 9015 of *Lecture Notes in Computer Science*, pages 498–527, 2015.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to Construct Random Functions. In *FOCS*, pages 464–479, 1984.

[GHS12] Craig Gentry, Shai Halevi, and Nigel P. Smart. Better Bootstrapping in Fully Homomorphic Encryption. In *Public Key Cryptography*

- *PKC 2012*, volume 7293 of *Lecture Notes in Computer Science*, pages 1–16, 2012.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (In)security of the Fiat-Shamir Paradigm. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 102–113, 2003.

[GSW13] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. In *Advances in Cryptology - CRYPTO 2013*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92, 2013.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Daniel Wichs. Leveled Fully Homomorphic Signatures from Standard Lattices. In *Proceedings of the 47th Annual ACM Symposium on the Theory of Computing*, pages 469–477, 2015.

[HAO15] Ryo Hiromasa, Masayuki Abe, and Tatsuaki Okamoto. Packing Messages and Optimizing Bootstrapping in GSW-FHE. In *Public Key Cryptography - PKC*, volume 9020 of *Lecture Notes in Computer Science*, pages 699–715, 2015.

[HAO16a] Ryo Hiromasa, Masayuki Abe, and Tatsuaki Okamoto. Packing Messages and Optimizing Bootstrapping in GSW-FHE. *IEICE Transactions*, 99-A(1):73–82, 2016.

[HAO16b] Ryo Hiromasa, Masayuki Abe, and Tatsuaki Okamoto. Tightly Secure Signatures based on the RSA Assumption. *Transactions of the Japan Society for Industrial and Applied Mathematics*, 26(4):416–439, 2016. To appear.

[HS14] Shai Halevi and Victor Shoup. Algorithms in HElib. In *Advances in Cryptology - CRYPTO 2014*, volume 8618 of *Lecture Notes in Computer Science*, pages 554–571, 2014.

[HS15] Shai Halevi and Victor Shoup. Bootstrapping for HElib. In *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 641–670, 2015.

[HSW13] Susan Hohenberger, Amit Sahai, and Brent Waters. Full Domain Hash from (Leveled) Multilinear Maps and Identity-Based Aggregate Signatures. In *Advances in Cryptology - CRYPTO 2013*, volume 8043 of *Lecture Notes in Computer Science*, pages 494–512, 2013.

[HSW14] Susan Hohenberger, Amit Sahai, and Brent Waters. Replacing a Random Oracle: Full Domain Hash From Indistinguishability Obfuscation. In *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Computer Science*, pages 201–220, 2014.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On Ideal Lattices and Learning with Errors Over Rings. In *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23, 2010.

[LRSW99] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym Systems. In *Selected Areas in Cryptography*, volume 1758 of *Lecture Notes in Computer Science*, pages 184–199, 1999.

[LTV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption. In *Proceedings of the 44th Annual ACM Symposium on the Theory of Computing*, pages 1219–1234, 2012.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions. In *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 465–484, 2011.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718, 2012.

[MSK02] Shigeo Mitsunari, Ryuichi Sakai, and Masao Kasahara. A New Traitor Tracing . *IEICE Transaction on Fundamentals*, E85-A(2):481–484, 2002.

[Oka92] Tatsuaki Okamoto. Provably Secure and Practical Identification Schemes and Corresponding Signatures. In *Advances in Cryptology - CRYPTO 1992*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53, 1992.

[Oka06] Tatsuaki Okamoto. Efficient Blind and Partially Blind Signatures Without Random Oracles. In *Theory of Cryptography*, volume 3876 of *Lecture Notes in Computer Science*, pages 80–99, 2006.

[Pai99] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Class. In *Advances in Cryptology - EUROCRYPT*

'99, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238, 1999.

[Pei09] Chris Peikert. Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem. In *Proceedings of the 41th Annual ACM Symposium on the Theory of Computing*, pages 333–342, 2009.

[Poi98] David Pointcheval. Strengthened Security for Blind Signatures. In *Advances in Cryptology - EUROCRYPT 1998*, volume 1403 of *Lecture Notes in Computer Science*, pages 391–405, 1998.

[PS96] David Pointcheval and Jacques Stern. Provably Secure Blind Signature Schemes. In *Advances in Cryptology - ASIACRYPT '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 252–265, 1996.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A Framework for Efficient and Composable Oblivious Transfer. In *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571, 2008.

[RAD78] Ronald L. Rivest, Len Adleman, and Michael Leonidas Dertouzos. On Data Banks and Privacy Homomorphisms. *Foundations of Secure Computation*, pages 169–180, 1978.

[Reg05] Oded Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *Proceedings of the 37th Annual ACM Symposium on the Theory of Computing*, pages 84–93, 2005.

[Rot11] Ron Rothblum. Homomorphic Encryption: from Private-Key to Public-Key. In *Theory of Cryptography*, pages 219–234, 2011.

[RSA78] Ronald L. Rivest, Adi Shamir, and Len Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystmes. *Communication of the ACM*, 21(2):120–126, 1978.

[RSA93] RSA LAboratories, Redwood city, California. *PKCS#1: RSA Encryption Standards*, Nov. 1993.

[SC12] Jae Hong Seo and Jung Hee Cheon. Beyond the Limitation of Prime-Order Bilinear Groups, and Round Optimal Blind Signatures. In *Theory of Cryptography*, volume 7194 of *Lecture Notes in Computer Science*, pages 133–150, 2012.

[SV10] Nigel P. Smart and Frederik Vercauteren. Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. In *Public Key Cryptography - PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443, 2010.

[SW14] Amit Sahai and Brent Waters. How to Use Indistinguishability Obfuscation: Deniable Encryption, and More. In *Proceedings of the 46th Annual ACM Symposium on the Theory of Computing*, pages 475–484, 2014.

[Ver12] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Yonina C. Eldar and Gitta Kutyniok, editors, *Compressed Sensing, Theory and Applications*, chapter 5, pages 210–268. Cambridge University Press, `http://www-personal.umich.edu/~romanv/papers/non-asymptotic-rmt-plain.pdf`, 2012.