A Study on Cryptography Resistant to Quantum Computing

(量子計算に耐性のある暗号技術に関する研究)

A dissertation

by

Shingo Sato

Supervisor: Prof. Dr. Junji Shikata

Graduate School of Environment and Information Sciences
Yokohama National University

March 2020

Abstract

Many researchers have worked on developing quantum computers which use quantum mechanics to perform computations. On the other hand, the standardized cryptosystems such as RSA encryption and elliptic curve cryptosystems are theoretically broken by utilizing quantum computing if sufficiently large-scale quantum computers are realized. Hence, it is important to design constructions of post-quantum cryptography or quantum-resistant cryptography, which is cryptography resistant to quantum computing. In fact, NIST (National Institute of Standards and Technology) has promoted the post-quantum cryptography (PQC) standardization project, and the development of post-quantum public key encryption and digital signatures has been advanced actively.

In this thesis, we aim at constructing quantum-secure cryptographic schemes of encryption, authentication, and cryptography with both properties of encryption and authentication. Quantum security means security against quantum computing in a quantum security model in which an adversary can commit a quantum computation in a black-box way. The properties of encryption and authentication are confidentiality and integrity of data, respectively. These are fundamental security notions in cryptology. In addition, quantum security models capture situations in which quantum computers are widespread and available to many users. Hence, it is important to consider confidentiality and integrity in a quantum security model.

First, we focus on public key encryption (PKE) satisfying selective opening (SO) security in quantum security models. Concretely, we prove that two PKE schemes from key encapsulation mechanism (KEM) constructions satisfy SO security against chosen ciphertext attacks (SO-CCA security) in the quantum random oracle model (QROM) or the quantum ideal cipher model (QICM). One is constructed from any KEM schemes meeting indistinguishability against chosen ciphertext attacks (IND-CCA security) and any data encapsulation mechanism (DEM) meeting both simulatability and integrity, and it satisfies SO-CCA security in the QICM. The other is constructed from a KEM scheme based on Fujisaki-Okamoto transformation and any message authentication code (MAC) meeting strong unforgeability, and it satisfies SO-CCA security in the QROM. We can obtain concrete SO-CCA secure PKE schemes from any KEM constructions meeting the security which standard-

ized KEM/PKE schemes are required to achieve (i.e., IND-CCA security) by combining with standardized DEM or MAC schemes.

Second, we deal with the quantum security of message authentication codes (MACs) with aggregation, which are called aggregate MAC (AMAC) and sequential AMAC (SAMAC). We formalize the quantum security of AMACs for the first time, and show that an existing AMAC scheme satisfies our security. In addition, we also formalize the quantum security of SAMACs and show that existing SAMAC schemes are broken in our security model. Then, we present two SAMAC schemes satisfying our security. One is constructed from any quantum-secure pseudorandom function, and the other is constructed from any randomized pseudorandom generator. To realize concrete AMAC/SAMAC schemes with the quantum security, we can apply existing cryptographic primitives including standardized ones. Hence, our schemes are useful in terms of practicality and security.

Third, we propose constructions of public key cryptography with both confidentiality and integrity, which is called signcryption. We present two schemes satisfying the security in the QROM or the standard model where there does not exist (quantum) random oracles and (quantum) ideal ciphers. One is based on lattice problems which are computationally hard problems even for quantum computers and satisfies the security of signcryption in the standard model. The other is a generic construction starting from any PKE scheme meeting a weaker security than IND-CCA security and any lossy identification scheme with several properties. By applying schemes submitted to the PQC standardization project, it is possible to construct concrete signcryption schemes in the QROM. In addition, we show that the key-size and ciphertext-size of our schemes are shorter than those of existing ones.

Acknowledgments

I would like to thank my advisor Professor Junji Shikata. He taught me how to research, make presentations, and write papers. Especially, he had discussed my research with me despite his tight schedule, and given me advice and suggestions based on deep insights. Thanks to him, I could build my academic career at graduate school. I am so grateful for his help in various aspects.

I would like to thank Professor Tsutomu Matsumoto. At meetings of the research center for information and physical society and other opportunities, he gave me fruitful comments from wide perspectives for my research. I thank Associate Professor Yoshioka for helpful comments to my research.

I am thankful to Professor Tatsunori Mori and Lecturer Shinichi Shirakawa for reviewing this thesis and giving helpful comments in spite of different research areas.

I am thankful to my collaborators on both works in this thesis and outside: Professor Shoichi Hirose (at University of Fukui), Tadahiro Uchikoshi, Masahiro Ebina, Tomoo Mikasa, Lu Cao, and Tomoki Miyazawa. I had discussed research with them, and they gave me helpful comments. Especially, Professor Shoichi Hirose gave me productive comments to my papers.

I am thankful to members of Shikata laboratory. Thanks to discussions and meetings with them, I could learn many things and deepen my perspective of modern cryptography, and I had enjoyable time at graduate school. Especially, I would like to thank Dr. Yohei Watanabe for caring about my future path.

I am grateful to secretaries of the research center for information and physical society: Yukiko Sugiyama, Mio Narimatsu, and Tomoko Ishidate. Thanks to their assistance, I could do my research smoothly.

Finally, I would like to thank my family. Thanks to their understanding and support, I could choose an academic career and continue my research.

Contents

Chapter 1

Introduction

1.1 Quantum Algorithms against Cryptosystems

In recent years, many researchers have worked on developing quantum computers, which leverage quantum mechanics to perform computations, because we can obtain various applications by utilizing quantum computations. Depending on applied principles, quantum computers are classified as follows: Universal quantum computers with logical operations (which are called quantum gates) aiming at solving any computational problems, and quantum computers aiming at solving specific optimization problems, such as quantum annealers. In order to carry out large quantum computations by using a universal quantum computer, it is necessary to correct errors throughout quantum computations. It seems that it will take a long time to realize large-scale quantum computers with fault-tolerance functionality though there are some experiments which have demonstrated universal sets of quantum gates with high fault-tolerance [33]. NIST (National Institute of Standards and Technology) has shown an opinion that it seems that a quantum computer which can break 2000-bit RSA encryption could be built by 2030 [33]. Regarding quantum annealers, it is reported that factoring problem can be solved by using quantum computers with quantum annealing though quantum annealing requires super-polynomial time [79].

On the other hand, since Diffie and Hellman introduced the notion of public key cryptography in 1976 [37], many (public key) cryptosystems based on number theoretic problems have been proposed. (e.g., RSA cryptosystems [114], ElGamal cryptosystems [50], Goldwasser-Micali encryption [56], Paillier encryption [109], elliptic-curve cryptosystems [103, 86].) However, if sufficiently large-scale (universal) quantum computers are realized in the future, these cryptosystems are broken theoretically. This is because there exist quantum polynomial-time algorithms solving number theoretic problems used in the cryptographic schemes while polynomial-time classical algorithms solving these problems have not been known. As for symmetric key cryptography,

it is also known that several standardized schemes are theoretically broken in a quantum security model where many users use quantum computers.

Concretely, Shor presented quantum polynomial-time algorithms which solve number theoretic problems such as factoring and discrete logarithm [121, 122]. Thus, by using Shor's algorithms, it is possible to break public key cryptosystems based on number theoretic problems like RSA, factoring, discrete logarithm Diffie-Hellman, and more cryptosystems. As for symmetric key cryptography, there exists a general attack using Grover's algorithm [57], which can find an n-bit secret key with time-complexity $O(2^{n/2})$. The measure against this attack is to double the bit-length of secret key-sizes. However, in quantum security models, several well-known schemes are broken in quantum polynomial-time even though we prevent the attack with Grover's algorithm. In 2013, Boneh and Zhandry introduced a quantum security model of message authentication codes (MACs) and proved that in this security model, there exist quantum polynomial-time algorithms which break the existing MACs [24]. In 2016, by utilizing Simon's quantum algorithm [123], Kaplan et al. presented quantum attacks breaking several well-known and classical symmetric key cryptosystems including standardized schemes, such as three-round Feistel scheme, CBC-MAC, PMAC, and GMAC [80] in the security models which were introduced in [24, 25]. In addition, other quantum attacks against several cryptosystems have been researched [10, 130, 70, 71, 75, 69].

In 2016, from the factors above, NIST called for proposals of standards of post-quantum cryptography: public key encryption, digital signatures, and key-establishment algorithms [106]. Since then, it has advanced post-quantum cryptography (PQC) standardization project actively [107]. In addition, European Telecommunications Standards Institute (ETSI) also have considered the road-map related to transforming to post-quantum cryptosystems. Notice that in this thesis, if we describe a post-quantum cryptography (or cryptosystem), the cryptography is resistant to quantum computing in a security model where adversaries can utilize quantum computations.

Therefore, it is essential to develop cryptographic primitives secure against attacks using quantum computers, and consider post-quantum ones with small time-complexity and small communication-complexity so that post-quantum primitives can be used in the real world. In the future, the following situations can be considered: (i) There is no threat of attacks utilizing quantum computations, (ii) An adversary can utilize quantum computing and the other users just use classical computing, (iii) all users including adversaries can leverage quantum computing. Notice that we do not consider that cryptosystems with quantum algorithms are used in these situations. In this thesis, we focus on situation (iii) because it is natural to assume situation (iii) in a quantum world where quantum computers are finally widespread and available to many users. In addition, from an academic viewpoint, establishing a post-quantum cryptography in situation (iii) is the most challenging, and the solutions in situation (iii) will be applicable even in other situations (i) and (ii). Regard-

ing the other situations, it is unlikely to consider situation (i) since many researches have paid much attention to the development of quantum computers, and it is predicted that quantum computers are realized in the near or distant future. Although situation (ii) may be natural as a situation in a few decades, we are interested in cryptography secure in a world where many users can use quantum computers.

1.2 Quantum Security of Cryptography

There are the following approaches to design cryptographic systems secure against quantum attacks:

- One is to construct a post-quantum cryptosystem in a classical security model where the adversary against the post-quantum cryptosystem can use quantum computing and only classical oracles which, given a classical query, returns the response. For example, we consider a post-quantum public key encryption scheme secure against chosen ciphertext attacks in a classical security model. We assume that this one is based on (computationally) hard problems even for quantum computers, such as lattice problems and error-correcting codes. The scheme is secure against attacks using quantum computing and accessing a decryption (classical) oracle which, given a ciphertext, returns the decrypted value.

- The other is to construct a post-quantum cryptosystem in a quantum security model where the adversary against the post-quantum cryptosystem can use not only quantum computing but also quantum oracles which, given a quantum superposition of queries (quantum query), returns the quantum superposition of the responses. For example, we consider a post-quantum public key encryption scheme secure against *quantum* chosen ciphertext attacks. This one is secure against attacks using quantum computing and accessing a decryption oracle which, given a quantum superposition of ciphertexts, returns the superposition of the decrypted values.

In this thesis, quantum security denotes security against attacks using quantum computation in a quantum security model. Notice that we do not consider the security model in which cryptographic primitives use quantum computing (e.g., [108]). This is because we are interested in how to construct post-quantum protocols starting from existing (standardized) cryptographic primitives with classical algorithms.

Regarding main post-quantum cryptography in a classical/quantum security model, there are cryptosystems based on the hardness of lattice problems, error-correcting codes, and solving systems of multivariate polynomials. These ones are based on the hardness of NP-hard problems or approximating these

problems, and resistant to attacks using quantum computing. This is because it is expected that the classes P and NP are not identical, and NP-hard problems cannot be solved in polynomial-time by using classical computers. Moreover, it is also believed that NP-hard problems cannot be solved in quantum polynomial-time even if quantum computers are utilized. In particular, of all post-quantum cryptosystems, ones based on lattice problems, which are called lattice-based cryptosystems, have been researched actively, because lattice problems or computational ones related to lattice problems are more suitable for applying to cryptography than other NP-hard problems. In addition, lattice problems can provide many cryptosystems with advanced functionality. As promising problems related to lattices, we can consider the problems of *learning with errors* (LWE) and *small integer solution* (SIS), which are focused on recently in constructions of lattice-based cryptography. So far, various and important constructions of cryptographic schemes have been proposed based on the problems: public key encryption [113, 111, 112, 90], and digital signatures [26, 53, 31, 97, 115, 96, 41, 42, 21, 139, 27]. In addition, there are constructions of key encapsulation mechanism [110, 31], identity-based encryption [53, 31, 3, 133, 139, 27, 134], identification schemes [94, 83, 95], and collision-resistant hash functions [54, 102].

Concerning quantum security models, the following models were formalized: Boneh et al. introduced the concept of the quantum random oracle model in 2011 [23], and Boneh and Zhandry formalized the quantum security models of message authentication codes (MACs), digital signatures (DSs), symmetric key encryption (SKE), and public key encryption (PKE) in 2013 [24, 25]. The quantum ideal cipher model was formalized by Hosoyamada and Yasuda in 2018 [72].

Regarding the quantum random oracle model (QROM), many works have aimed at presenting cryptosystems and security proofs in this security model. The (classical) random oracle model (ROM), which was formalized by Bellare and Rogaway in 1993 [18], is a security model in which a cryptosystem uses ideal random functions, and any adversary against the cryptosystem has access to the ideal random functions as oracles which are called random oracles. The QROM is a model in which any adversary is allowed to issue quantum queries to random oracles which are called quantum random oracles. In a real world, cryptosystems secure in the ROM/QROM use cryptographic hash functions as random oracles. It is natural to focus on designing post-quantum cryptosystems in the QROM. In fact, the security of standardized cryptosystems such as RSA-OAEP and Diffie-Hellman key exchange are proven in the ROM, and those of most schemes submitted to the PQC standardization project are guaranteed in the QROM. Since the QROM was introduced, many works have been studied about cryptosystems secure in the QROM as follows: PKE schemes [23, 127, 64, 116, 76, 73, 77, 78], DSs [131, 84, 91, 39], identity-based public-key encryption schemes [137, 81], the general proof techniques in the QROM [129, 9], and more works.

In addition, the classical/quantum ideal cipher model is a model where any adversary has access to ideal ciphers E as oracles (i.e., $E_{\mathsf{k}} : \mathcal{X} \to \mathcal{X}$ is an ideal random permutation for each key k). Concerning a work related to quantum ideal cipher model (QICM), it is proven that there exists a quantum one-way function if the underlying block cipher is a quantum ideal cipher [72].

The security of MACs in a quantum security model was introduced by Boneh and Zhandry [24]. In this security model, any adversary against a MAC scheme is allowed to issue quantum queries to a tagging oracle which, given a message, returns the tag on the message while in the classical security model, it is allowed to issue only classical queries. Furthermore, they showed that some existing MAC schemes such as Carter-Wegman MAC [132] and a one-time secure MAC from a pair-wise independent hash function, which are secure in the classical security model, can be broken in the security model. Moreover, they proposed constructions satisfying the quantum security. In addition, they also formalized the security of DSs, SKE, and PKE in the quantum security model and proposed generic constructions of these cryptographic protocols with the quantum security [25]. As described before, in 2016, Kaplan et.al presented quantum algorithms breaking several SKE schemes in the security models of [24, 25]. In particular, standardized MAC and authenticated encryption schemes such as CBC-MAC and GCM can be broken in the quantum security model. Hence, it is important to consider quantum security of cryptographic systems.

1.3 Overview of Our Contributions

In this thesis, we propose quantum-secure cryptosystems. We consider quantum security of (1) encryption, (2) authentication, and (3) cryptography with both properties of encryption and authentication. Encryption schemes guarantee confidentiality and prevent data from being compromised. Authentication schemes guarantee integrity and prevent data from being substituted. Cryptography with both properties of encryption and authentication guarantees both confidentiality and integrity.

We describe the overview of our contributions in (1), (2), and (3) as follows.

(1) Encryption

We aim at proving that hybrid encryption schemes satisfy *selective-opening security* (SO security) in the QROM or the QICM. Hybrid encryption is PKE which consists of public key cryptography and symmetric key cryptography, which are called key encapsulation mechanism (KEM) and data encapsulation mechanism (DEM), respectively. SO security, of which concept was introduced by Bellare, Hofheinz, and Yilek in 2009 [15], guarantees the confidentiality of ciphertexts whose messages and randomness are not compromised even though

adversaries get secret information such as messages and randomness of several ciphertexts.

Our main motivation is to construct SO-CCA secure PKE schemes from any IND-CCA secure KEM/PKE schemes which were submitted to the PQC standardization project. *Indistinguishability against chosen ciphertext attacks* (IND-CCA security) has been studied actively as one of the most important security of PKE, and NIST requires that standardized schemes satisfy this security. However, IND-CCA security may not be enough in the multi-user setting of *selective-opening security against chosen ciphertext attacks* (SO-CCA security). It is proven that IND-CCA secure PKE schemes do not necessarily fulfill SO-CCA security in [14, 67, 66]. Namely, there are counterexamples such as IND-CCA secure PKE constructions which are broken in the security model of SO-CCA security. All submitted schemes just guarantee IND-CCA security. Moreover, in fact, there are many situations where messages and randomness of ciphertexts are leaked because of weakness in system's design, even though cryptosystems are not broken theoretically. Therefore, it is important to consider how to construct SO-CCA secure PKE schemes from the submitted KEM/PKE constructions.

Our contribution is to prove that two hybrid encryption schemes from IND-CCA secure KEM/PKE schemes satisfy SO-CCA security in the QROM or the QICM, for the first time. One is a PKE scheme included in the standard KEM/DEM framework which was designed in [35]. We require that the underlying KEM meets IND-CCA security and the underlying DEM scheme meets both simulatability and integrity which were formalized in [61] and [17], respectively. This one satisfies SO-CCA security in the QICM. The other is constructed from any FO-based KEM scheme which is a generic construction of IND-CCA secure KEM, and any MAC with strong unforgeability. This construction satisfies SO-CCA security in the QROM. Notice that FO-based KEM schemes include most constructions submitted to the PQC standardization project. The differences between these hybrid encryption schemes are as follows:

- We can apply any IND-CCA secure KEM to the scheme in the standard KEM/DEM framework while the underlying KEM in the other one is a particular KEM scheme which was categorized in [64].

- The underlying DEM in the KEM/DEM framework must meet both integrity and a particular property (simulatability) which has been dealt only in [61] while the underlying MAC in the other scheme just needs to fulfill a well-known security (strong unforgeability) which standardized (deterministic) MACs meet.

Notice that PKE is a stronger cryptography than KEM. Namely, if a PKE scheme meets IND-CCA security, then it also satisfies the IND-CCA security

of KEM. Thus, we can apply any IND-CCA secure KEM/PKE scheme to either SO-CCA secure hybrid encryption.

(2) Authentication

Our purpose is to propose quantum-secure MACs with aggregation. Specifically, we deal with aggregate MAC (AMAC) and sequential aggregate MAC (SAMAC) in quantum security models. In the ordinary MAC, a sender generates a tag (MAC-tag) on a message by using its secret key, and a receiver verifies a message/MAC-tag pair by using the corresponding key. In the case in which multiple senders with distinct keys generate MAC-tags on their local messages, an AMAC compresses the multiple MAC-tags into a tag (aggregate tag), and a receiver with the corresponding keys verifies whether a pair of multiple messages and an aggregate tag is valid or not. SAMAC is AMAC which can verify not only messages but also the order of sequential messages.

Our main motivation is to construct quantum-secure AMACs and SAMACs. This is because when multiple senders send messages simultaneously, AMACs and SAMACs are widely used since it is possible to reduce the size of MAC-tags over channels. Thus, it is possible to utilize AMACs/SAMACs for applications using resource-constrained devices such as audit-logging systems, wireless network sensors, data-partitioning, and other applications. Furthermore, existing AMACs/SAMACs may be broken in quantum security models. In fact, it is known that several existing ordinary MACs including standardized schemes are broken in a quantum security model [24, 80]. However, there is no work which researches the quantum security of AMACs/SAMACs.

Our contribution is to formalize the quantum security of AMACs/SAMACs and present generic constructions with our security. More details are as follows:

- We formalize the quantum security of AMACs. Our definition is reasonable since this is the extension of the existing security [82] in the classical security model. Furthermore, we prove that an existing AMAC constructed from any MAC [82] satisfies our security if the underlying MAC scheme meets unforgeability in the quantum security model of [24]. Notice that it is possible to construct concrete AMAC schemes satisfying our security because several MACs [136, 24, 124] meet the quantum security of [24]. In particular, we can construct the AMACs from standardized MAC schemes [124].

- We formalize the quantum security of SAMACs, which is the extension of the existing security [44] in the classical security model. Besides, we show that existing SAMAC schemes [44, 128] are broken in our security model by using existing quantum attacks of [24, 80]. We present two generic constructions satisfying our security. One is constructed from any quantum-secure pseudorandom function (QPRF), and the other is constructed from any randomized pseudorandom generator (randomized

PRG). The difference between these schemes is that we can apply deterministic pseudorandom functions to the SAMAC from QPRF while we can apply randomized PRGs to the other one. This means that concrete constructions from QPRFs can be realized by applying well-known and practical MAC schemes such as NMAC/HMAC [124] while we can realize concrete ones from randomized PRGs which are based on computationally hard problems for quantum computers such as learning parity with noise (LPN) problem.

(3) Encryption and Authentication

Our goal is to propose quantum-secure signcryption schemes with short key-size and ciphertext-size. Signcryption is a public key cryptography satisfying both properties of PKE and DSs. Concretely, we construct two schemes with both confidentiality and integrity against inside adversary in a multi-user setting. Inside adversaries and the multi-user setting mean the following:

- Inside adversaries against a signcryption scheme can use either senders' or receivers' secret keys in order to break the scheme in a security game while outside adversaries use only public parameters and public keys. Thus, inside adversaries are stronger than outside adversaries.

- In the multi-user setting of signcryption, multiple senders and multiple receivers communicate one another while in the two-user setting, a sender and a receiver communicate each other. In particular, in the security model of the multi-user setting, the (inside) adversary against a signcryption scheme can generate receivers' or senders' key-pairs at any time and use them to break the scheme in a security game while in the two-user setting, the (inside) adversary generates only one receiver's or sender's key-pair at the beginning of a security game.

Hence, one of desirable security of signcryption is security against inside adversary in the multi-user setting. Notice that signcryption schemes obtained by combining PKE and DS constructions in the straightforward way do not necessarily satisfy both confidentiality and integrity in the security model [11, 100].

Our motivation is to construct quantum-secure signcryption schemes with short key-size and ciphertext-size. Concerning existing post-quantum signcryption schemes, there are constructions obtained by applying concrete lattice-based primitives to generic constructions of [34, 104]. However, the key-size and ciphertext-size of these signcryption schemes are much longer than those of constructions based on number theoretic problems. Moreover, by using signcryption schemes, we can realize secure channels guaranteed both confidentiality and integrity from insecure ones such as the Internet. Hence, it is significant to develop efficient cryptographic systems with both confidentiality and integrity in a quantum world.

Our contribution is to propose two quantum-secure signcryption schemes satisfying both confidentiality and integrity against inside adversaries in the multi-user setting:

- One is a lattice-based construction with the securities in the standard model which is a model without (quantum) random oracles and (quantum) ideal ciphers. Concretely, this one is based on the well-known computationally hard problems for quantum computers: Learning with errors (LWE) problem [113] and small integer solution (SIS) problem [102], which are related to lattice problems.

- The other is a generic construction in the QROM. This one is basically obtained by combining a PKE scheme based on Fujisaki-Okamoto transformation [48] and a DS scheme based on Fiat-Shamir transformation [46].

The reason for presenting two schemes is as follows: The standard model is a stronger security model than the QROM while constructions in QROM are more efficient than those in the standard model in terms of key-size, ciphertext-size, and time-complexity, generally. Therefore, the lattice-based construction is important in terms of security while the construction in the QROM is significant in terms of efficiency.

Furthermore, we show that the key-size and ciphertext-size of our schemes are shorter than those of existing ones, which are obtained by applying suitable lattice-based primitives to existing generic constructions of [34, 104]. These existing ones of [34, 104] satisfy both confidentiality and integrity against inside adversary in the multi-user setting.

Chapter 2

Preliminaries

2.1 Notations

We denote the set of integers by $\mathbb{Z}$, the set of real numbers by $\mathbb{R}$, and the set of complex numbers by $\mathbb{C}$. For a set S, let $|S|$ be the cardinality of S. For a set S, the sampling of a uniformly random element $x \in S$ is denoted by $x \xleftarrow{U} S$. For a positive integer n, let $[n] := \{1, 2, \ldots, n\}$. For a set S, let $\{x\}_{x \in S}$ be a set of all elements in S. In particular, for a positive integer n and values $x_1, \ldots, x_n$, let $\{x_i\}_{i \in [n]} = \{x_1, x_2, \ldots, x_n\}$ denote a set of $x_1, \ldots, x_n$, and let $(x_i)_{i \in [n]} = (x_1, x_2, \ldots, x_n)$ denote a sequence of $x_1, \ldots, x_n$.

Vectors are in column forms and written as bold italic letters $\boldsymbol{x}$. For a vector $\boldsymbol{x}$, let x_i be the i-th component of $\boldsymbol{x}$, and $\boldsymbol{x}^\top$ be the row vector of $\boldsymbol{x}$. Matrices are written as bold italic capital letters $\boldsymbol{X}$, and for a matrix $\boldsymbol{X}$, let $\boldsymbol{x}_i$ be the i-th column vector of $\boldsymbol{X}$. $\|\cdot\|$ denotes the Euclidean norm. For a matrix $\boldsymbol{X}$, let $\|\boldsymbol{X}\| := \max_i \|\boldsymbol{x}_i\|$. For a matrix $\boldsymbol{X}$, let $s_1(\boldsymbol{X}) := \max_{\boldsymbol{u}}(\boldsymbol{X}\boldsymbol{u})$, where $\boldsymbol{u}$ is a unit value.

We write that a function $f(\lambda)$ is negligible in λ (or f is a negligible function in λ) if $f(\lambda) < 1/g(\lambda)$ for a polynomial g and sufficiently large λ. A negligible function in λ is denoted by $\mathsf{negl}(\lambda)$. A polynomial of λ is denoted by $\mathsf{poly}(\lambda)$.

The statistical distance between two distributions $\mathcal{X}, \mathcal{Y}$ over a finite domain $\mathcal{D}$ is defined as $\Delta(\mathcal{X}, \mathcal{Y}) := \frac{1}{2}\sum_{w \in \mathcal{D}} |\mathcal{X}(w) - \mathcal{Y}(w)|$.

For a randomized algorithm A and the input x of A, $\mathsf{A}(x; r)$ denotes a deterministic algorithm, where r is a random coin used by A. "Probabilistic polynomial-time" and "quantum polynomial-time" are abbreviated as PPT and QPT, respectively.

2.2 Quantum Computation

Quantum Systems. A quantum system is defined as a complex Hilbert space with an inner product. The state of a quantum system is denoted by a vector $|\varphi\rangle \in \mathcal{H}$ such that $\langle\varphi|\varphi\rangle = 1$, where φ is a label for the vector, $\langle\varphi|$ is a dual

vector of $|\varphi\rangle$, and $\langle\varphi|\varphi'\rangle$ is an inner product between two vectors $|\varphi\rangle$ and $|\varphi'\rangle$. A single qubit is denoted by $|\varphi\rangle = \psi_0 |0\rangle + \psi_1 |1\rangle \in \mathbb{C}^2$ with an orthonormal computational basis $\{|0\rangle, |1\rangle\}$ (e.g., $|0\rangle = (1,0)^\top, |1\rangle = (0,1)^\top$) and amplitudes $\psi_0, \psi_1 \in \mathbb{C}$ such that $\psi_0^2 + \psi_1^2 = 1$. An n-qubit state is defined as a linear combination $|\varphi\rangle = \sum_{x\in\{0,1\}^n} \psi_x |x\rangle$, where $\{|x\rangle\}_{x\in\{0,1\}^n}$ is an orthonormal computational basis of an n-dimensional Hilbert space with an inner product, and $\psi_x \in \mathbb{C}$ ($x \in \{0,1\}^n$) are amplitudes such that $\sum_{x\in\{0,1\}^n} |\psi_x|^2 = 1$.

For different quantum systems $\mathcal{H}_A$ and $\mathcal{H}_B$, the composite quantum system is defined as the tensor product space $\mathcal{H}_A \otimes \mathcal{H}_B$. For $|\varphi_A\rangle \in \mathcal{H}_A$ and $|\varphi_B\rangle \in \mathcal{H}_B$, the tensor product state is denoted by $|\varphi_A\rangle |\varphi_B\rangle = |\varphi_A\rangle \otimes |\varphi_B\rangle$.

Measurement. If $|\varphi\rangle = \sum_{x\in\{0,1\}^n} \psi_x |x\rangle$ is measured in the computational basis $\{|x\rangle\}_{x\in\{0,1\}^n}$, $|\varphi\rangle$ collapses to a classical state $|x\rangle$ with probability $|\psi_x|^2$. More generally, information can be obtained from a quantum superposition state $|\varphi\rangle$ by using positive-operator valued measure (POVM) $M = \{M_i\}_{i\in[m]}$, where $M_1, \ldots, M_m$ are positive semi-definite matrices such that $\sum_i M_i = \boldsymbol{I}$. When we measure a state $|\varphi\rangle$, outcome i is obtained with probability $\langle\varphi|M_i|\varphi\rangle$.

Evolution of Quantum Systems. When a state $|\varphi\rangle \in \mathcal{H}$ is transformed into another state $|\varphi'\rangle \in \mathcal{H}$, a unitary transformation is applied. Namely, $|\varphi\rangle = \boldsymbol{U} |\varphi\rangle$, where $\boldsymbol{U}$ is a unitary matrix over $\mathcal{H}$.

Oracle Access For a quantum oracle $\mathsf{O} : \mathcal{X} \to \mathcal{Y}$, issuing a quantum query $\sum_{x\in\mathcal{X},y\in\mathcal{Y}} \psi_{x,y} |x,y\rangle$ to O (quantum access to O) is written as

$$\sum_{x\in\mathcal{X},y\in\mathcal{Y}} \psi_{x,y} |x,y\rangle \mapsto \sum_{x\in\mathcal{X},y\in\mathcal{Y}} \psi_{x,y} |x, y \oplus \mathsf{O}(x)\rangle .$$

2.2.1 Quantum Random Oracle Model

In the random oracle model, we assume that there exists an ideal random function H, and all parties can access this function. The quantum random oracle model (QROM) is defined as the model in which any quantum adversary can submit quantum queries to random oracles.

The quantum ideal cipher model (QICM), which was introduced in [72], is defined as follows: A block cipher with a key space $\mathcal{K}$ and a message space $\mathcal{X}$ is defined as a mapping $E : \mathcal{K}\times\mathcal{X} \to \mathcal{X}$ which is a permutation over $\mathcal{X}$ for any key in $\mathcal{K}$. In the QICM, quantum adversaries are allowed to issue quantum queries to oracles $E^+ : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ and $E^- : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$. For any $\mathsf{k} \in \mathcal{K}$ and any $y \in \mathcal{X}$, the response of $E^-(\mathsf{k}, y)$ is x such that $E^+(\mathsf{k}, x) = y$. For any $\mathsf{k} \in \mathcal{K}$, we write oracles (permutations over $\mathcal{X}$) $E_\mathsf{k}(\cdot) = E(\mathsf{k}, \cdot)$, $E^+_\mathsf{k}(\cdot) = E^+(\mathsf{k}, \cdot)$, and $E^-_\mathsf{k}(\cdot) = E^-(\mathsf{k}, \cdot)$.

In this thesis, QROM (resp. QICM) denotes the security model where quantum adversaries are allowed to issue quantum queries to random oracles (resp. ideal ciphers), but submit only classical queries to the other oracles.

2.2.2 Semi-Classical Oracle

We describe semi-classical oracle which was introduced in [9] and utilize this oracle for our security proofs. We consider quantum access to an oracle with a domain $\mathcal{X}$. A semi-classical oracle O_S^{SC} for a subset $S \subseteq \mathcal{X}$ uses an indicator function $f_S : \mathcal{X} \to \{0,1\}$ with the subset S which evaluates 1 if $x \in S$ is given, and evaluates 0 otherwise. When O_S^{SC} is given a quantum query $\sum_{x\in\mathcal{X}} \psi_x |x\rangle |0\rangle$ with the input register Q and the output register R, it maps

$$\sum_{x\in\mathcal{X}} \psi_{x,z} |x\rangle |0\rangle \mapsto \sum_{x\in\mathcal{X}} \psi_x |x\rangle |f_S(x)\rangle ,$$

and measures the register R. Then, the quantum query $\sum_{x\in\mathcal{X}} \psi_x |x\rangle |0\rangle$ collapses to either $\sum_{x\in\mathcal{X}\setminus S} \psi'_x |x\rangle |0\rangle$ or $\sum_{x\in S} \psi'_x |x\rangle |1\rangle$. Let Find be the event that O_S^{SC} returns $\sum_{x\in S} \psi'_x |x\rangle |1\rangle$ for a quantum query $\sum_{x\in S} \psi_x |x\rangle$. For a quantum oracle H with domain $\mathcal{X}$ and a subset $S \subseteq \mathcal{X}$, let $\mathsf{H}\backslash S$ be an oracle which first queries O_S^{SC} and then H.

By using semi-classical oracles, [9] proved the following propositions. We notice that for query depth d and the number of queries q, we use q such that $q \geq d$ in the same way as Theorem 2.8 in [73] or Lemma 3 in [78].

Proposition 2.1 (Theorem 1 in [9])**.** *Let $S \subseteq \mathcal{X}$ be random. Let $\mathsf{H} : \mathcal{X} \to \mathcal{Y}$, $\mathsf{G} : \mathcal{X} \to \mathcal{Y}$ be random functions such that $\mathsf{H}(x) = \mathsf{G}(x)$ for all $x \in \mathcal{X}\backslash S$, and let z be a random bit-string (S, H, G and z may have an arbitrary joint distribution). Let A be any quantum algorithm issuing at most q quantum queries to oracles. Then, it holds that*

$$\left|\Pr[1 \leftarrow \mathsf{A}^{\mathsf{H}}(z)] - \Pr[1 \leftarrow \mathsf{A}^{\mathsf{G}}(z)]\right| \leq 2\sqrt{q \cdot \Pr[\mathsf{Find} \mid 1 \leftarrow \mathsf{A}^{\mathsf{H}\backslash S}(z)]}.$$

Proposition 2.2 (Corollary 1 in [9])**.** *Let A be any quantum algorithm issuing at most q quantum queries to a semi-classical oracle with domain $\mathcal{X}$. Suppose that $S \subseteq \mathcal{X}$ and $z \in \{0,1\}^*$ are independent. Then, it holds that* $\Pr[\mathsf{Find} \mid \mathsf{A}^{\mathsf{O}_S^{SC}}(z)] \leq 4q \cdot P_{\max}$*, where* $P_{\max} = \max_{x\in\mathcal{X}} \Pr[x \in S]$.

2.2.3 Generic Search Problem

We describe lemmas related to generic (quantum) search problems [10, 74].

Let Ber_γ be the Bernoulli distribution with bias $\gamma \in (0,1)$. Namely, for a bit $b \leftarrow \mathsf{Ber}_\gamma$, $\Pr[b = 1] = \gamma$, and $\Pr[b = 0] = 1 - \gamma$. For some finite set $\mathcal{X}$ and a real number $\gamma \in [0,1]$, let $F : \mathcal{X} \to \{0,1\}$ be an oracle such that for $x \in \mathcal{X}$, $F(x)$ is distributed following Ber_γ. Let N be an oracle such that for $x \in \mathcal{X}$, $N(x) = 0$. The generic search problem is to determine whether the given oracle is F or N.

The following lemma related to generic search problem holds:

Lemma 2.1 (Generic Search Problem [10, 74])**.** *Let $\gamma \in [0,1]$ be a real number, $\mathcal{X}$ be a finite set, $q \geq 0$ be an integer, and $F : \mathcal{X} \to \{0,1\}$ be the following function: For each $x \in \mathcal{X}$, $F(x) = 1$ with probability γ, and $F(x) = 0$ else. Let N be the function with $\forall x \in \mathcal{X} : N(x) = 0$. For any QPT algorithm A issuing at most q queries, then*

$$\left|\Pr[\mathsf{A}^F \to 1] - \Pr[\mathsf{A}^N \to 1]\right| \leq 2q\sqrt{\gamma}.$$

Furthermore, concerning the variant of the generic search problem, which is called generic search problem with bounded probabilities [64, 84], the following lemma holds:

Lemma 2.2 (Generic Search Problem with Bounded Probabilities [64, 84])**.** *Let $\gamma \in [0,1]$ be a real number, $\mathcal{X}$ be a finite set, $q \geq 0$ be an integer, and $F : \mathcal{X} \to \{0,1\}$ be the following function: For each $x \in \mathcal{X}$, $F(x) = 1$ with probability γ, and $F(x) = 0$ else. For any QPT algorithm $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ issuing at most q queries, then we have*

$$\Pr[\mathsf{Exp}^{\mathrm{gspb}}_{\gamma,\mathsf{A}} \to 1] \leq 8\gamma(q+1)^2,$$

where $\mathsf{Exp}^{\mathrm{gspb}}_{\gamma,\mathsf{A}}$ is defined as follows:

1. *$(\gamma(x))_{x\in\mathcal{X}} \leftarrow \mathsf{A}_0$.*
2. *If $\exists x \in \mathcal{X}$: $\gamma(x) > \gamma$, return 0.*
3. *$\forall x \in \mathcal{X}$, $F(x) \leftarrow \mathsf{Ber}_{\gamma(x)}$.*
4. *$x \leftarrow \mathsf{A}_1^F$.*
5. *Return $F(x)$.*

2.3 Lattice Background

We define lattices, discrete Gaussian distributions, and computational problems related to lattices. As cryptographic tools, we describe lattice-based trapdoor functions and some algorithms using these trapdoor functions.

2.3.1 Lattices

For a positive integer n, let a basis $\boldsymbol{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n\}$ ($\boldsymbol{b}_i \in \mathbb{R}^n$ for $i \in [n]$) denote a set of n linearly independent vectors in $\mathbb{R}^n$. The n-dimensional lattice $\mathcal{L}$ generated by the basis $\boldsymbol{B}$ is defined as

$$\mathcal{L} = \mathcal{L}(\boldsymbol{B}) = \left\{ \sum_{i\in[n]} x_i \boldsymbol{b}_i \mid x_i \in \mathbb{Z} \text{ for } i \in [n] \right\}.$$

Notice that in this thesis, we consider the *full-rank* lattices above. For a lattice $\mathcal{L}$, the *successive minimum* $\lambda_i(\mathcal{L})$ is the smallest radius r such that $\mathcal{L}$ has i linearly independent lattice vectors of norm at most r. In particular, $\lambda_1(\mathcal{L})$ denotes the length of the shortest nonzero vector in $\mathcal{L}$. (i.e., $\lambda_1(\mathcal{L}) = \min_{\boldsymbol{x}\in\mathcal{L}} \|\boldsymbol{x}\|$.)

Furthermore, we define integer lattices. For positive integers n and q, a matrix $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, and a vector $\boldsymbol{u} \in \mathbb{Z}_q^n$, m-dimensional q-ary lattices are defined as

$$\begin{aligned}\Lambda(\boldsymbol{A}) &:= \{\boldsymbol{z} \mid \exists \boldsymbol{s} \in \mathbb{Z}_q^n, \boldsymbol{z} = \boldsymbol{A}^\top \boldsymbol{s} \bmod q\}\\ \Lambda^\perp(\boldsymbol{A}) &:= \{\boldsymbol{z} \mid \boldsymbol{A}\boldsymbol{z} = \boldsymbol{0} \bmod q\}\\ \Lambda_{\boldsymbol{u}}^\perp(\boldsymbol{A}) &:= \{\boldsymbol{z} \mid \boldsymbol{A}\boldsymbol{z} = \boldsymbol{u} \bmod q\}\end{aligned}$$

We can view $\Lambda_{\boldsymbol{u}}^\perp(\boldsymbol{A})$ as the shifted lattice of $\Lambda^\perp(\boldsymbol{A})$.

2.3.2 Gaussian

Let m be a positive integer, and Λ be a subset of $\mathbb{Z}^m$. For any real vector $\boldsymbol{c} \in \mathbb{R}^m$ and real number $\sigma \in \mathbb{R}$, let

$$\rho_{\sigma,\boldsymbol{c}} = \exp(-\pi\|\boldsymbol{x} - \boldsymbol{c}\|^2/\sigma^2) \in \mathbb{R}^m$$

be the Gaussian function with a center c and a parameter σ. The discrete Gaussian distribution over Λ with a center $\boldsymbol{c}$ and a parameter σ is defined by

$$\forall \boldsymbol{y} \in \mathcal{L},\ \mathcal{D}_{\Lambda,\boldsymbol{c},\sigma}(\boldsymbol{y}) = \frac{\rho_{\sigma,\boldsymbol{c}}(\boldsymbol{y})}{\rho_{\sigma,\boldsymbol{c}}(\mathcal{L})},$$

where $\rho_{\sigma,\boldsymbol{c}}(\Lambda) = \sum_{\boldsymbol{x}\in\Lambda} \rho_{\sigma,\boldsymbol{c}}(\boldsymbol{x})$. In this thesis, $\mathcal{D}_{\Lambda,\boldsymbol{0},\sigma}$ is abbreviated as $\mathcal{D}_{\Lambda,\sigma}$.

Regarding the Gaussian distribution, the following lemma holds.

Lemma 2.3 ([102])**.** *For any $\mathcal{L}$ with basis $\boldsymbol{T}$, $\boldsymbol{c} \in \mathbb{R}^m$ and Gaussian parameter $\sigma \geq \|\boldsymbol{T}\| \cdot \omega(\sqrt{\log m})$, we have* $\Pr[\|\boldsymbol{x} - \boldsymbol{c}\| > \sigma\sqrt{m} \mid \boldsymbol{x} \leftarrow \mathcal{D}_{\mathbb{Z},\boldsymbol{c},\sigma}^m] < \mathsf{negl}(n)$.

2.3.3 Computational Problems

We define some lattice problems and computational problems such as learning with errors (LWE) problem and small integer solution (SIS) problem.

Lattice Problems

We define *shortest vector problem* (SVP), which is one of well-studied lattice problems.

Definition 2.1 (Shortest Vector Problem (SVP))**.** *Given a basis $\boldsymbol{B}$ of a lattice $\mathcal{L} = \mathcal{L}(\boldsymbol{B})$, find a non-zero vector $\boldsymbol{v} \in \mathcal{L}$ such that $\|\boldsymbol{v}\| = \lambda_1(\mathcal{L})$.*

We define two lattice problems GapSVP_γ and SIVP_γ which are related to LWE and SIS problems. These are parameterized by an approximation factor $\gamma = \gamma(n) \geq 1$, where n is the dimension of a lattice. Informally, GapSVP_γ is the decisional approximate SVP, and *short independent vectors problem* (SIVP) is a problem which, given a basis of a lattice, finds a short basis of the lattice. Concretely, these problems are defined as follows.

Definition 2.2 (GapSVP_γ)**.** *Given a basis $\boldsymbol{B}$ of a lattce $\mathcal{L} = \mathcal{L}(\boldsymbol{B})$ and a real number $r > 0$, return 1 if $\lambda_1(\mathcal{L}) \leq r$, and return 0 if $\lambda_1(\mathcal{L}) > \gamma \cdot r$.*

Definition 2.3 (SIVP_γ)**.** *Given a basis $\boldsymbol{B}$ of a full-rank n-dimensional lattice $\mathcal{L} = \mathcal{L}(\boldsymbol{B})$, find a set $\boldsymbol{S} = \{\boldsymbol{s}_i\}_{i\in[n]} \subset \mathcal{L}$ of n linearly independent lattice vectors where $\|\boldsymbol{s}_i\| \leq \gamma \cdot \lambda_n(\mathcal{L})$ for all $i \in [n]$.*

Learning with Errors (LWE)

LWE problem was introduced by Regev in [113]. As cryptosystems based on LWE, it is possible to realize *cryptomania* such as trapdoor functions, public key encryption, oblivious transfer, and cryptography with advanced functionality.

Let n, q be positive integers. First, we define $\mathsf{O}_{\boldsymbol{s},\chi}$ as an oracle which, given a vector $\boldsymbol{s} \in \mathbb{Z}_q^n$ and a distribution χ over $\mathbb{Z}_q$, returns $(\boldsymbol{a}, \boldsymbol{s}^\top \boldsymbol{a} + e) \in \mathbb{Z}_q^{n+1}$, where $\boldsymbol{a} \xleftarrow{U} \mathbb{Z}_q^n$ and $e \leftarrow \chi$. Besides, O_U is an oracle which returns $(\boldsymbol{a}, u) \in \mathbb{Z}_q^{n+1}$, where $\boldsymbol{a} \xleftarrow{U} \mathbb{Z}_q^n$ and $u \xleftarrow{U} \mathbb{Z}_q$.

Then, the LWE assumption is defined as follows.

Definition 2.4 ($\mathsf{LWE}_{n,q,\chi}$ assumption)**.** *Let n be a positive integer, $q = q(n)$ be a prime, and χ be a probabilistic distribution over $\mathbb{Z}_q$. The $\mathsf{LWE}_{n,q,\chi}$ assumption holds if for any PPT algorithm D solving $\mathsf{LWE}_{n,q,\chi}$, the advantage*

$$\mathsf{Adv}_{\mathsf{D}}^{\mathsf{LWE}_{n,q,\chi}}(\lambda) := \left|\Pr[\mathsf{D}^{\mathsf{O}_{\boldsymbol{s},\chi}}(1^\lambda) \to 1] - \Pr[\mathsf{D}^{\mathsf{O}_U}(1^\lambda) \to 1]\right|$$

is negligible in λ, where $\boldsymbol{s} \xleftarrow{U} \mathbb{Z}_q^n$ and χ is a distribution over $\mathbb{Z}_q$.

Small Integer Solution (SIS)

SIS was introduced by Ajtai in [5] and is related to lattice problems such as GapSVP_γ and SIVP_γ. This problem provides one-way functions, pseudorandom functions, collision-resistant hash functions, identification schemes, digital signatures, and more cryptographic primitives. SIS was formalized by Micciancio and Regev in [102] and defined as follows.

Definition 2.5 ($\mathsf{SIS}_{n,q,\beta,m}$ assumption)**.** *Let n be a positive integer, $q = q(n)$ be a prime, $\beta = \beta(n)$ be a positive real number, and $m = m(n)$ be a positive*

integer. The $\mathsf{SIS}_{n,q,\beta,m}$ *assumption holds if for any PPT algorithm* F*, the advantage*

$$\mathsf{Adv}_{\mathsf{F}}^{\mathsf{SIS}_{n,q,\beta,m}}(\lambda) := \Pr\left[\begin{array}{l|l} \boldsymbol{A}\boldsymbol{e} = \boldsymbol{0} \bmod q \wedge & \boldsymbol{A} \xleftarrow{U} \mathbb{Z}_q^{n\times m}; \\ \|\boldsymbol{e}\| \leq \beta \wedge & \boldsymbol{e} \leftarrow \mathsf{F}(1^\lambda, \boldsymbol{A}, \beta) \\ \boldsymbol{e} \neq \boldsymbol{0} & \end{array}\right]$$

is negligible in λ*.*

The following theorems, which were proved in [113] and [102], show the hardness of LWE and SIS problems.

Theorem 2.1 ([113])**.** *Let* n *be a positive integer,* $q = \mathsf{poly}(n)$ *be a positive integer,* χ *be (discrete) Gaussian distribution* χ *with a parameter* $\alpha \in (0,1)$ *such that* $\alpha q > 2\sqrt{n}$*. If there exists a polynomial-time algorithm that solves* $\mathsf{LWE}_{n,q,\chi}$*, then there exists a quantum polynomial-time algorithm that approximates* $\mathsf{GapSVP}_{\tilde{O}(n/\alpha)}$ *and* $\mathsf{SIVP}_{\tilde{O}(n/\alpha)}$*.*

Theorem 2.2 ([102])**.** *Let* n *be a positive integer,* $\beta > 0$ *be a real number,* $q = \beta \cdot \mathsf{poly}(n)$ *be a prime, and* $m = \Theta(n \log n)$ *be an integer. If there exists a polynomial-time algorithm that solves* $\mathsf{SIS}_{n,q,\beta,m}$*, then there exists a polynomial-time algorithm that approximates* $\mathsf{GapSVP}_{\tilde{O}(\beta\sqrt{n})}$ *and* $\mathsf{SIVP}_{\tilde{O}(\beta\sqrt{n})}$*.*

2.3.4 Lattice-based Trapdoor

Briefly, a one-way function is a function which is easy to compute, but hard to invert. A trapdoor function is a one-way function which is easy to invert by using a secret called a trapdoor, but hard to invert without the trapdoor. A lattice-trapdoor notion with a gadget matrix $\boldsymbol{G}$, which was introduced in [101], is defined as follows:

Definition 2.6 (Definition 5.2 in [101])**.** *Let* $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ *and* $\boldsymbol{G} \in \mathbb{Z}_q^{n\times w}$ *be matrices with* $m \geq w \geq n$*. A* $\boldsymbol{G}$*-trapdoor for* $\boldsymbol{A}$ *is a matrix* $\boldsymbol{T} \in \mathbb{Z}^{(m-w)\times w}$ *such that* $\boldsymbol{A}\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{I}\end{bmatrix} = \boldsymbol{H}\boldsymbol{G}$ *for some invertible matrix* $\boldsymbol{H} \in \mathbb{Z}_q^{n\times n}$*.* $\boldsymbol{H}$ *can be viewed as a tag or a label. The quality of the trapdoor is measured by its largest singular value* $s_1(\boldsymbol{T})$*.*

By utilizing this lattice-trapdoor, it is possible to realize injective trapdoor functions and preimage sampleable trapdoor functions (regarding the definition of trapdoor functions, see Section 2.5), and the following proposition holds.

Theorem 2.3 (Theorem 5.1 in [101])**.** *There is an efficient randomized algorithm* $\mathsf{TrapGen}(1^n, 1^m, q)$ *that, given any integers* $n \geq 1, q \geq 2$ *and sufficiently large* $m = O(n \log n)$*, outputs a parity-check matrix* $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$ *and a trapdoor*

$\boldsymbol{T}$ such that the distribution of $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$ is $\mathsf{negl}(n)$-close to the uniform distribution in $\mathbb{Z}_q^{n \times m}$. Moreover, there are efficient algorithms, denoted by Invert and $\mathsf{SampleD}$, which do the following with overwhelming probability over all random choices:

- *For $\boldsymbol{b}^\top = \boldsymbol{s}^\top \boldsymbol{A} + \boldsymbol{e}^\top \in \mathbb{Z}_q^m$, where $\boldsymbol{s} \in \mathbb{Z}_q^n$ and either $\|\boldsymbol{e}\| < q/O(\sqrt{n \log q})$ or $\boldsymbol{e} \leftarrow \mathcal{D}_{\mathbb{Z}^m, \alpha q}$ for $1/\alpha \geq \sqrt{n \log q} \cdot \omega(\sqrt{\log n})$, the deterministic algorithm $\mathsf{Invert}(\boldsymbol{T}, \boldsymbol{A}, \boldsymbol{b})$ outputs $\boldsymbol{s}$ and $\boldsymbol{e}$.*
- *For any $\boldsymbol{u} \in \mathbb{Z}_q^n$, and large enough $s = O(\sqrt{n \log q})$, the randomized algorithm $\mathsf{SampleD}(\boldsymbol{T}, \boldsymbol{A}, \boldsymbol{u}, s)$ outputs samples from a distribution which is $\mathsf{negl}(n)$-close to $\mathcal{D}_{\Lambda_{\boldsymbol{u}}^\perp(\boldsymbol{A}), s \cdot \omega(\sqrt{\log n})}$.*

2.4 Pseudorandom Generator and Pseudorandom Function

First, we define k-wise independent hash functions. A function $f : \mathcal{X} \to \mathcal{Y}$ is a k-wise independent hash function if $\forall y_1, \ldots, y_k \in \mathcal{Y}$ and distinct $x_1, \ldots, x_k \in \mathcal{X}$,

$$\Pr[f(x_1) = y_1 \wedge f(x_2) = y_2 \wedge \ldots \wedge f(x_k) = y_k] = \frac{1}{|\mathcal{Y}|^k}.$$

Pairwise independent hash functions denote 2-wise independent hash functions.

Consider a function $G : \mathcal{X} \to \mathcal{Y}$, where for a security parameter λ, $\mathcal{X} = \mathcal{X}(\lambda)$ is a domain, and $\mathcal{Y} = \mathcal{Y}(\lambda)$ is a range. And, we assume that $|x| < |y|$ holds for all $x \in \mathcal{X}$ and all $y \in \mathcal{Y}$, where $|x|$ and $|y|$ are the bit-lengths of x and y, respectively. Then, G is said to be a *pseudorandom generator (PRG)*, if for any PPT algorithm A, the following $\mathsf{Adv}^{\mathrm{prg}}_{G,\mathsf{A}}(\lambda)$ is negligible in λ:

$$\mathsf{Adv}^{\mathrm{prg}}_{G,\mathsf{A}}(\lambda) := \left| \Pr[\mathsf{A}(G(x)) \to 1 \mid x \xleftarrow{U} \mathcal{X}] - \Pr[\mathsf{A}(y) \to 1 \mid y \xleftarrow{U} \mathcal{Y}] \right|.$$

In addition, a PRG $G : \mathcal{X} \to \mathcal{Y}$ with a randomness space $\mathcal{R}$ is called a *randomized PRG*, if for any PPT algorithm A, the following $\mathsf{Adv}^{\mathrm{prg}}_{G,\mathsf{A}}(\lambda)$ is negligible in λ:

$$\mathsf{Adv}^{\mathrm{prg}}_{G,\mathsf{A}}(\lambda) := \left| \Pr[\mathsf{A}(G(x;r)) \to 1 \mid x \xleftarrow{U} \mathcal{X}; r \leftarrow \mathcal{R}] - \Pr[\mathsf{A}(y) \to 1 \mid y \xleftarrow{U} \mathcal{Y}] \right|.$$

Note that for a randomized PRG G with a randomness space $\mathcal{R}$ and any seed $x \in \mathcal{X}$, we write $G(x;r)$ as a deterministic function, where randomness $r \in \mathcal{R}$ is not sampled uniformly.

A function $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$, where for a security parameter λ, $\mathcal{K} = \mathcal{K}(\lambda)$ is a key space, $\mathcal{X} = \mathcal{X}(\lambda)$ is a domain, and $\mathcal{Y} = \mathcal{Y}(\lambda)$ is a range, is a

pseudorandom function (PRF), if the following $\mathsf{Adv}^{\mathrm{pr}}_{\mathsf{D},\mathsf{PRF}}(\lambda)$ is negligible for any PPT algorithm,

$$\mathsf{Adv}^{\mathrm{pr}}_{\mathsf{PRF},\mathsf{D}}(\lambda) := \left|\Pr\left[\mathsf{D}^{\mathsf{PRF}_{\mathsf{k}}(\cdot)}(1^\lambda) \to 1\right] - \Pr\left[\mathsf{D}^{\mathsf{RF}(\cdot)}(1^\lambda) \to 1\right]\right|,$$

where $\mathsf{PRF}_{\mathsf{k}}(\cdot)$ is an oracle which, on input $x \in \mathcal{X}$, outputs $\mathsf{PRF}(\mathsf{k}, x)$, and $\mathsf{RF}(\cdot)$ is an oracle which, on input $x \in \mathcal{X}$, outputs a value $\mathsf{RF}(x)$ of a random function $\mathsf{RF} : \mathcal{X} \to \mathcal{Y}$.

In addition, a quantum-secure PRF (QPRF) is defined in a similar way as above by assuming that D is any QPT algorithm allowed to issue quantum superposition of queries to oracles.

2.5 Trapdoor Function

A family $\{g_a : \mathcal{D}(\lambda) \to \mathcal{R}(\lambda)\}$ of trapdoor functions is denoted by a tuple of polynomial-time algorithms $(\mathsf{TrapGen}, \mathsf{Eval}, \mathsf{Invert})$, where λ is a security parameter, a is a parameter of a function g, $\mathcal{D}(\lambda)$ is a domain, and $\mathcal{R}(\lambda)$ is a range.

Trapdoor Generation $\mathsf{TrapGen}$ is a randomized algorithm which, on input a security parameter 1^λ, outputs a parameter a and a trapdoor t.

Evaluation Eval is an algorithm which, on input a parameter a and $s \in \mathcal{D}(\lambda)$, evaluates the function $g_a(s) \in \mathcal{R}(\lambda)$.

Inversion Invert is a deterministic algorithm which, on input a trapdoor t and $b \in \mathcal{R}$, outputs $s \in \mathcal{D}(\lambda)$.

As the correctness, it is required that $\mathsf{TDF} = (\mathsf{TrapGen}, \mathsf{Eval}, \mathsf{Invert})$ meet the following: For all $(a, t) \leftarrow \mathsf{TrapGen}(1^\lambda)$ and all $s \in \mathcal{D}(\lambda)$, we have $s = \mathsf{Invert}(t, b)$ with overwhelming probability, where $b \leftarrow \mathsf{Eval}(a, s)$.

As the security of trapdoor functions, one-wayness is defined as follows:

Definition 2.7 (One-wayness)**.** *A family* $\{g_a : \mathcal{D}(\lambda) \to \mathcal{R}(\lambda)\}$ *of trapdoor functions which is given by* $\mathsf{TDF} = (\mathsf{TrapGen}, \mathsf{Eval}, \mathsf{Invert})$ *meets one-wayness if for any PPT algorithm* A*, the following advantage of* A *is negligible in* λ*:*

$$\mathsf{Adv}^{\text{one-wayness}}_{\mathsf{TDF},\mathsf{A}}(\lambda) := \Pr\left[s = s' \;\middle|\; \begin{array}{c}(a,t) \leftarrow \mathsf{TrapGen}(1^\lambda);\\ s \xleftarrow{U} \mathcal{D}(\lambda); b \leftarrow \mathsf{Eval}(a,s);\\ s' \leftarrow \mathsf{A}(1^\lambda, a, b)\end{array}\right].$$

In addition, we define preimage sampleable trapdoor functions. A family $\{f_a : \mathcal{D}(\lambda) \to \mathcal{R}(\lambda)\}$ of preimage sampleable trapdoor functions is denoted by a tuple of polynomial-time algorithms $(\mathsf{TrapGen}, \mathsf{Eval}, \mathsf{Sample}, \mathsf{SampleD})$, where λ is a security parameter, a is a parameter of a function f, $\mathcal{D}(\lambda)$ is a domain, and $\mathcal{R}(\lambda)$ is a range.

Trapdoor Generation $\mathsf{TrapGen}$ is a randomized algorithm which, on input a security parameter 1^λ, outputs a parameter a and a trapdoor t.

Evaluation Eval is an algorithm which, on input a function index a and $x \in \mathcal{D}(\lambda)$, evaluates the function $f_a(x) \in \mathcal{R}(\lambda)$.

Domain Sampling Sample is a randomized algorithm which, on input a security parameter λ, samples x from some distribution over $\mathcal{D}(\lambda)$ such that $f_a(x)$ is uniform over $\mathcal{R}(\lambda)$.

Preimage Sampling $\mathsf{SampleD}$ is a randomized algorithm which, on input a trapdoor t and $y \in \mathcal{R}(\lambda)$, samples from the distribution $x \in \mathsf{Sample}(\lambda)$ conditioned on $f_a(x) = y$.

As the security of preimage sampleable trapdoor functions, we define one-wayness, preimage min-entropy, and collision-resistance.

Definition 2.8 (One-wayness)**.** *A family* $\{f_a : \mathcal{D}(\lambda) \to \mathcal{R}(\lambda)\}$ *of trapdoor functions which is given by* $\mathsf{PSF} = (\mathsf{TrapGen}, \mathsf{Eval}, \mathsf{Sample}, \mathsf{SampleD})$ *satisfies one-wayness if for any PPT algorithm* A*, the following advantage of* A *is negligible in* λ*:*

$$\mathsf{Adv}^{\text{one-wayness}}_{\mathsf{PSF},\mathsf{A}}(\lambda) := \Pr\left[x' \in f_a^{-1}(y) \;\middle|\; \begin{array}{c} (a,t) \leftarrow \mathsf{TrapGen}(1^\lambda); \\ x \leftarrow \mathsf{Sample}(\lambda); y \leftarrow \mathsf{Eval}(a,x); \\ x' \leftarrow \mathsf{A}(1^\lambda, a, y) \end{array} \right].$$

Definition 2.9 (Preimage Min-entropy)**.** *For every* $y \in \mathcal{R}(\lambda)$*, the conditional min-entropy of* $x \leftarrow \mathsf{Sample}(1^\lambda)$ *given* $f_a(x) = y$ *is at least* $\omega(\log 1^\lambda)$*.*

Definition 2.10 (Collision-Resistance)**.** *A family* $\{f_a : \mathcal{D}(\lambda) \to \mathcal{R}(\lambda)\}$ *of trapdoor functions which is given by* $\mathsf{PSF} = (\mathsf{TrapGen}, \mathsf{Eval}, \mathsf{Sample}, \mathsf{SampleD})$ *satisfies collision-resistance if for any PPT algorithm* A*, the following advantage of* A *is negligible in* λ*:*

$$\mathsf{Adv}^{\text{cr}}_{\mathsf{PSF},\mathsf{A}}(\lambda) := \Pr\left[f_a(x) = f_a(x') \wedge x \neq x' \;\middle|\; \begin{array}{c} (a,t) \leftarrow \mathsf{TrapGen}(1^\lambda); \\ (x,x') \leftarrow \mathsf{A}(1^\lambda, a) \end{array} \right].$$

2.6 Encryption

In this section, we define the models and securities of public key encryption, key encapsulation mechanism, and data encapsulation mechanism. These schemes guarantee confidentiality.

2.6.1 Public Key Encryption

A public key encryption (PKE) scheme consists of three polynomial-time algorithms (KGen, Enc, Dec): For a security parameter λ, let $\mathcal{M} = \mathcal{M}(\lambda)$ be a message space.

Key Generation KGen is a randomized algorithm which, on input a security parameter 1^λ, outputs a public key pk and a secret key sk.

Encryption Enc is an algorithm which, on input a public key pk and a message $\mathsf{m} \in \mathcal{M}$, outputs a ciphertext ct.

Decryption Dec is a deterministic algorithm which, on input a secret key sk and a ciphertext ct, outputs a message $\mathsf{m} \in \mathcal{M}$ or an invalid symbol $\perp$.

A PKE scheme $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ meets δ-correctness if for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$, and all $\mathsf{m} \in \mathcal{M}$, we have $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \neq \mathsf{m}$ with at most probability δ, where $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$. Then, it is required that PKE schemes meet δ-correctness for a negligible δ in λ.

As security of PKE, we define the following.

Definition 2.11 ($\mathsf{OW\text{-}CPA}$ security)**.** *A PKE scheme* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *meets* $\mathsf{OW\text{-}CPA}$ *security if for any PPT adversary* A *against* PKE*, the advantage* $\mathsf{Adv}^{\text{ow-cpa}}_{\mathsf{PKE},\mathsf{A}}(\lambda) := \Pr[\mathsf{A}\ \textit{wins}]$ *is negligible in* λ*, where* $[\mathsf{A}\ \textit{wins}]$ *is an event that* A *wins in the following game:*

Key Generation: *A challenger generates* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$.

Challenge: *The challenger returns* $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}^*)$*, where* $\mathsf{m}^* \xleftarrow{U} \mathcal{M}$.

Output: A *outputs* $\mathsf{m}' \in \mathcal{M}$. A *wins if* $\mathsf{m}^* = \mathsf{m}'$.

Definition 2.12 ($\mathsf{IND\text{-}CPA}$ security)**.** *A PKE scheme* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *meets* $\mathsf{IND\text{-}CPA}$ *security if for any PPT adversary* A *against* PKE*, the advantage* $\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathsf{A}}(\lambda) := |2 \cdot \Pr[\mathsf{A}\ \textit{wins}] - 1|$ *is negligible in* λ*, where* $[\mathsf{A}\ \textit{wins}]$ *is an event that* A *wins in the following game:*

Key Generation: *A challenger generates* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$.

Challenge: *When* A *submits* $(\mathsf{m}_0, \mathsf{m}_1)$ *such that* $|\mathsf{m}_0| = |\mathsf{m}_1|$*, the challenger returns* $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$*, where* $b \xleftarrow{U} \{0, 1\}$.

Output: A *outputs the guessing bit* $b' \in \{0, 1\}$. A *wins if* $b = b'$.

Definition 2.13 ($\mathsf{IND\text{-}CCA}$ security)**.** *A PKE scheme* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *meets* $\mathsf{IND\text{-}CCA}$ *security if for any PPT algorithm* A *against* PKE*, the advantage* $\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{PKE},\mathsf{A}}(\lambda) := |2 \cdot \Pr[\mathsf{A}\ \textit{wins}] - 1|$ *is negligible in* λ*, where* $[\mathsf{A}\ \textit{wins}]$ *is the event that* A *wins in the following game:*

Key Generation: *A challenger generates* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$.

Queries 1: *Given a ciphertext* ct*, a decryption oracle* DEC *returns* $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$.

Challenge: *When* A *submits* $(\mathsf{m}_0, \mathsf{m}_1)$ *such that* $|\mathsf{m}_0| = |\mathsf{m}_1|$*, the challenger returns* $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b)$*, where* $b \xleftarrow{U} \{0, 1\}$.

$\underline{\mathsf{Exp}^{\text{real-so-cca}}_{\mathsf{PKE},\mathsf{A}}}$

$I \leftarrow \emptyset$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$

$(\mathcal{M}_\mathrm{D}, \mathsf{st}) \leftarrow \mathsf{A}_0(\mathsf{pk})$

$(\mathsf{m}_1, \ldots, \mathsf{m}_n) \leftarrow \mathcal{M}_\mathrm{D}$

$(r_1, \ldots, r_n) \leftarrow \mathcal{R}$

$\forall i \in [n], \mathsf{ct}_i = \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_i; r_i)$

$out \leftarrow \mathsf{A}_1^{\mathsf{OPEN},\mathsf{DEC}}(\mathsf{st}, \mathsf{ct}_1, \ldots, \mathsf{ct}_n)$

return $R(\mathcal{M}_\mathcal{D}, \mathsf{m}_1, \ldots, \mathsf{m}_n, I, out)$

$\underline{\mathsf{OPEN}(i)}$

$I \leftarrow I \cup \{i\}$

return (m_i, r_i)

$\underline{\mathsf{DEC}(\mathsf{ct})}$

if $\mathsf{ct} \in \{\mathsf{ct}_i\}_{i\in[n]}$, **return** $\perp$

$\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$

return $\mathsf{m} \in \mathcal{M} \cup \{\perp\}$

$\underline{\mathsf{Exp}^{\text{ideal-so-cca}}_{\mathsf{PKE},\mathsf{S}}}$

$I \leftarrow \emptyset$

$(\mathcal{M}_\mathrm{D}, \mathsf{st}) \leftarrow \mathsf{S}_0(1^\lambda)$

$(\mathsf{m}_1, \ldots, \mathsf{m}_n) \leftarrow \mathcal{M}_\mathrm{D}$

$out \leftarrow \mathsf{S}_1^{\mathsf{OPEN}}(\mathsf{st}, |\mathsf{m}_1|, \ldots, |\mathsf{m}_n|)$

return $R(\mathcal{M}_\mathcal{D}, \mathsf{m}_1, \ldots, \mathsf{m}_n, I, out)$

$\underline{\mathsf{OPEN}(i)}$

$I \leftarrow I \cup \{i\}$

return m_i

Figure 2.1: Experiments in Real-SIM-SO-CCA and Ideal-SIM-SO-CCA Games

Queries 2: *Given a ciphertext* ct*, a decryption oracle* DEC *returns* $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$*.* A *is not allowed to issue* ct^**.*

Output: A *outputs the guessing bit* $b' \in \{0, 1\}$*.* A *wins if* $b = b'$*.*

Definition 2.14 (SIM-SO-CCA security)**.** *A PKE scheme* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *meets* SIM-SO-CCA *security if for any PPT algorithms* $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$*,* $\mathsf{S} = (\mathsf{S}_0, \mathsf{S}_1)$ *and any relation* R*, the advantage* $\mathsf{Adv}^{\text{sim-so-cca}}_{\mathsf{PKE},\mathsf{A},\mathsf{S},R}(\lambda)$ *is negligible in* λ*.* $\mathsf{Adv}^{\text{sim-so-cca}}_{\mathsf{PKE},\mathsf{A},\mathsf{S},R}(\lambda)$ *is defined as follow:*

$$\mathsf{Adv}^{\text{sim-so-cca}}_{\mathsf{PKE},\mathsf{A},\mathsf{S},R}(\lambda) := \left|\Pr[\mathsf{Exp}^{\text{real-so-cca}}_{\mathsf{PKE},\mathsf{A}} \to 1] - \Pr[\mathsf{Exp}^{\text{ideal-so-cca}}_{\mathsf{PKE},\mathsf{S}} \to 1]\right|,$$

where the two experiments $\mathsf{Exp}^{\text{real-so-cca}}_{\mathsf{PKE},\mathsf{A}}$ *and* $\mathsf{Exp}^{\text{ideal-so-cca}}_{\mathsf{PKE},\mathsf{S}}$ *are defined in Figure 2.1.*

2.6.2 Key Encapsulation Mechanism

A key encapsulation mechanism (KEM) scheme consists of three polynomial-time algorithms (KGen, Encap, Decap): For a security parameter λ, let $\mathcal{K} = \mathcal{K}(\lambda)$ be a key space.

Key Generation KGen is a randomized algorithm which, on input a security parameter 1^λ, outputs a public key pk and a secret key sk.

Encapsulation Encap is an algorithm which, on input a public key pk, outputs a ciphertext ct and a key $\mathsf{k} \in \mathcal{K}$.

Decapsulation Decap is a deterministic algorithm which, on input a secret key sk and a ciphertext ct, outputs a key $\mathsf{k} \in \mathcal{K}$ or an invalid symbol $\perp$.

A KEM scheme $(\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ meets δ-correctness if for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$, we have $\mathsf{Decap}(\mathsf{sk}, \mathsf{ct}) \neq \mathsf{k}$ with at most probability δ, where $(\mathsf{ct}, \mathsf{k}) \leftarrow \mathsf{Encap}(\mathsf{pk})$. It is required that KEM schemes meet δ-correctness with a negligible function δ for λ.

As a security of KEM schemes, we define $\mathsf{IND\text{-}CCA}$ security.

Definition 2.15 ($\mathsf{IND\text{-}CCA}$ security)**.** *A KEM scheme* $\mathsf{KEM} = (\mathsf{KGen}, \mathsf{Encap}, \mathsf{Decap})$ *meets* $\mathsf{IND\text{-}CCA}$ *security if for any PPT adversary* A *against* KEM*, the advantage* $\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathsf{A}}(\lambda) := |2 \cdot \Pr[\mathsf{A} \text{ wins}] - 1|$ *is negligible in* λ*, where* $[\mathsf{A} \text{ wins}]$ *is the event that* A *wins in the following game:*

Setup: *A challenger generates* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ *and sends* pk *to* A*.*

Oracle Access: A *is allowed to access the following oracles:*

- $\mathsf{Challenge}()$*: Given a challenge request, the challenger computes* $(\mathsf{ct}^*, \mathsf{k}_0) \leftarrow \mathsf{Encap}(\mathsf{pk})$ *and chooses* $\mathsf{k}_1 \in \mathcal{K}$ *uniformly at random. It returns* $(\mathsf{ct}^*, \mathsf{k}_b)$*, where* $b \xleftarrow{U} \{0, 1\}$*.*
- $\mathsf{DEC}(\mathsf{ct})$*: Given a ciphertext query* ct*, a decapsulation oracle* $\mathsf{DEC}(\mathsf{ct})$ *returns* $\mathsf{k}' \leftarrow \mathsf{Decap}(\mathsf{sk}, \mathsf{ct}) \in \mathcal{K} \cup \{\perp\}$*.* A *is not allowed to submit* ct^* *to* $\mathsf{DEC}(\cdot)$*.*

Output: A *outputs the guessing bit* $b' \in \{0, 1\}$*.* A *wins if* $b = b'$*.*

2.6.3 Data Encapsulation Mechanism

A data encapsulation mechanism (DEM) scheme consists of two polynomial-time algorithms $(\mathsf{Enc}, \mathsf{Dec})$ with a key space $\mathcal{K} = \mathcal{K}(\lambda)$ and a message space $\mathcal{M} = \mathcal{M}(\lambda)$ for a security parameter λ.

Encryption Enc is an algorithm which, on input a secret key $\mathsf{k} \in \mathcal{K}$ and a message $\mathsf{m} \in \mathcal{M}$, outputs a ciphertext ct.

Decryption Dec is a deterministic algorithm which, on input a secret key $\mathsf{k} \in \mathcal{K}$, a ciphertext ct, outputs a message $\mathsf{m} \in \mathcal{M}$ or an invalid symbol $\perp$.

We require that DEM schemes meet correctness as follows: A DEM scheme $(\mathsf{Enc}, \mathsf{Dec})$ meets correctness if for any $\mathsf{k} \in \mathcal{K}$ and any $\mathsf{m} \in \mathcal{M}$, it holds that $\mathsf{m} = \mathsf{Dec}(\mathsf{k}, \mathsf{ct})$, where $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{k}, \mathsf{m})$.

As security of DEM, we define *indistinguishability against one-time attacks* ($\mathsf{IND\text{-}OT}$ security), *one-time integrity of chosen ciphertext attacks* ($\mathsf{OT\text{-}INT\text{-}CTXT}$ security [17]), and *simulatability* [61].

Definition 2.16 (IND-OT security)**.** *A DEM scheme* $\mathsf{DEM} = (\mathsf{Enc}, \mathsf{Dec})$ *with a key space* $\mathcal{K}$ *meets* IND-OT *security if for any PPT adversary* A *against* DEM*, the advantage* $\mathsf{Adv}^{\text{ind-ot}}_{\mathsf{DEM},\mathsf{A}}(\lambda) := |\Pr[\mathsf{A}\ \textit{wins}] - 1/2|$ *is negligible in* λ*, where* [A *wins*] *is the event that* A *wins in the following game.*

Setup: *A challenger chooses* $\mathsf{k} \in \mathcal{K}$ *uniformly at random.*

Challenge: *When* A *submits* $(\mathsf{m}_0, \mathsf{m}_1)$ *such that* $|\mathsf{m}_0| = |\mathsf{m}_1|$*, the challenger chooses* $b \in \{0,1\}$ *uniformly at random and returns* $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{k}, \mathsf{m}_b)$*.*

Output: A *outputs* $b' \in \{0,1\}$*.* A *wins if* $b = b'$*.*

Definition 2.17 (OT-INT-CTXT security)**.** *A DEM scheme* $\mathsf{DEM} = (\mathsf{Enc}, \mathsf{Dec})$ *with a key space* $\mathcal{K}$ *meets* OT-INT-CTXT *security if for any PPT adversary* A *against* DEM*, the advantage* $\mathsf{Adv}^{\text{int-ctxt}}_{\mathsf{DEM},\mathsf{A}}(\lambda) := \Pr[\mathsf{A}\ \textit{wins}]$ *is negligible in* λ*, where* [A *wins*] *is the event that* A *wins in the following game:*

Setup: *A challenger chooses a key* $\mathsf{k} \in \mathcal{K}$ *uniformly at random, and sets* $\mathsf{win} \leftarrow 0$ *and* $L_{\mathsf{ct}} \leftarrow \emptyset$*.*

Oracle Access: A *is allowed to access the following oracles:*

- $\mathsf{ENC}(\mathsf{m})$*: If* $L_{\mathsf{ct}} \neq \emptyset$*, an encryption oracle* $\mathsf{ENC}(\mathsf{m})$ *returns* $\perp$*. Otherwise, it returns* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{k}, \mathsf{m})$*, and sets* $L_{\mathsf{ct}} \leftarrow L_{\mathsf{ct}} \cup \{\mathsf{ct}\}$*.*
- $\mathsf{VRFY}(\mathsf{ct})$*: Given a ciphertext query* ct*, a verification oracle* $\mathsf{VRFY}(\mathsf{ct})$ *runs* $\mathsf{m}' \leftarrow \mathsf{Dec}(\mathsf{k}, \mathsf{m})$*. If* $\mathsf{m}' \neq \perp$ *and* $\mathsf{ct} \notin L_{\mathsf{ct}}$*, it sets* $\mathsf{win} \leftarrow 1$*. It returns* 1 *if* $\mathsf{m}' \neq \perp$*, and returns* 0 *otherwise.*

Final: A *wins if* $\mathsf{win} = 1$*.*

Furthermore, in order to define simulatability, we regard DEM as block cipher-based DEM which uses a block cipher as a black-box. In addition, we view the key space $\mathcal{K}$ of DEM schemes as a product set $\mathcal{K} = \mathcal{K}' \times \mathcal{K}''$, where $\mathcal{K}'$ is the key space of a block cipher, and $\mathcal{K}''$ is the key space of encryption using a block cipher as a black-box.

To define simulatable DEM, oracle DEM and permutation-driven DEM are defined following [61].

Definition 2.18 (Oracle DEM)**.** *A DEM scheme* $(\mathsf{O.Enc}^\pi, \mathsf{O.Dec}^\pi)$ *with a key space* $\mathcal{K}$ *and a message space* $\mathcal{M}$ *is an oracle DEM scheme for a domain* $\mathcal{X}$ *if* $(\mathsf{O.Enc}, \mathsf{O.DEM})$ *has access to a permutation* π *on* $\mathcal{D}$*, and if for all permutations* $\pi : \mathcal{X} \to \mathcal{X}$*, all* $\mathsf{k} \in \mathcal{K}$*, and all* $\mathsf{m} \in \mathcal{M}$*, it holds that* $\mathsf{m} = \mathsf{Dec}^\pi(\mathsf{k}, \mathsf{ct})$*, where* $\mathsf{ct} \leftarrow \mathsf{Enc}^\pi(\mathsf{k}, \mathsf{m})$*, as the correctness of the DEM* $(\mathsf{O.Enc}^\pi, \mathsf{O.Dec}^\pi)$*.*

Definition 2.19 (Permutation-Driven DEM)**.** *A DEM scheme* $\mathsf{DEM} = (\mathsf{Enc}, \mathsf{Dec})$ *with a key space* $\mathcal{K} = \mathcal{K}' \times \mathcal{K}''$ *and a message space* $\mathcal{M}$ *is a* $(\mathcal{K} \times \mathcal{X})$*-permutation-driven DEM if the following conditions hold:*

- DEM *is an oracle DEM* $(\mathsf{O.Enc}^{\pi}, \mathsf{O.Dec}^{\pi})$ *for a domain* $\mathcal{X}$ *with a block cipher* $\{E_{\mathsf{k}'} : \mathcal{X} \to \mathcal{X}\}_{\mathsf{k}' \in \mathcal{K}'}$ *as the permutation* π *over* $\mathcal{X}$*.*
- *For any key* $(\mathsf{k}', \mathsf{k}'') \in \mathcal{K}$*, any message* $\mathsf{m} \in \mathcal{M}$*, and any ciphertexts* ct*, it holds that* $\mathsf{Enc}((\mathsf{k}', \mathsf{k}''), \mathsf{m}) = \mathsf{O.Enc}^{E_{\mathsf{k}'}}(\mathsf{k}'', \mathsf{m})$ *and* $\mathsf{Dec}((\mathsf{k}', \mathsf{k}''), \mathsf{ct}) = \mathsf{O.Dec}^{E_{\mathsf{k}'}}(\mathsf{k}'', \mathsf{ct})$*.*

Then, the simulatability of oracle DEM [61] is defined as follows.

Definition 2.20 (Simulatability of Oracle DEM)**.** *Let* $\mathsf{DEM} = (\mathsf{Enc}, \mathsf{Dec})$ *with a key space* $\mathcal{K} = \mathcal{K}' \times \mathcal{K}''$ *and a message space* $\mathcal{M}$ *be an oracle DEM scheme for a domain* $\mathcal{X}$*. And, we assume that* DEM *has the following algorithms* Fake *and* Make*:*

- Fake*: A randomized algorithm which, given a key* $\mathsf{k}'' \in \mathcal{K}''$ *and the bit-length* $|\mathsf{m}|$ *of messages, outputs a ciphertext* ct *and a state* st*.*
- Make*: A randomized algorithm which, given a state* st *and a message* $\mathsf{m} \in \mathcal{M}$*, outputs a relation* $\tilde{\pi} \in \mathcal{X} \times \mathcal{X}$ *which has functions* $\tilde{\pi}^{+} : \mathcal{X} \to \mathcal{X}$ *and* $\tilde{\pi}^{-} : \mathcal{X} \to \mathcal{X}$ *such that if* $(\alpha, \beta) \in \tilde{\pi}$*,* $\alpha = \tilde{\pi}^{+}(\beta)$ *and* $\beta = \tilde{\pi}^{-}(\alpha)$ *hold.*

The oracle DEM scheme DEM *meets* ϵ*-simulatablility if for all* $\mathsf{k} = (\mathsf{k}', \mathsf{k}'') \in \mathcal{K}$*, all* $\mathsf{m} \in \mathcal{M}$*, and the set* $\Pi^{\mathsf{m}}_{\mathsf{k}''} := \{\tilde{\pi} \mid (\mathsf{ct}, \mathsf{st}) \leftarrow \mathsf{Fake}(\mathsf{k}'', |\mathsf{m}|); \tilde{\pi} \leftarrow \mathsf{Make}(\mathsf{st}, \mathsf{m})\}$*, the following conditions hold:*

- *The set* $\Pi^{\mathsf{m}}_{\mathsf{k}''}$ *can be extended to a set of uniformly distributed permutations on* $\mathcal{X}$*.*
- *For any permutation* π *extended* $\Pi^{\mathsf{m}}_{\mathsf{k}''}$*, it holds that* $\Pr[\mathsf{ct} \neq \mathsf{O.Enc}^{\pi}(\mathsf{k}'', \mathsf{m})] \leq \epsilon$*, where* $\mathsf{ct} \leftarrow \mathsf{Fake}(\mathsf{k}'', |\mathsf{m}|)$*.*
- *The time-complexity of algorithms* $\mathsf{Fake}(\mathsf{k}', |\mathsf{m}|)$ *and* $\mathsf{Make}(\mathsf{st}, \mathsf{m})$ *does not exceed the time-complexity of algorithm* $\mathsf{Enc}(\mathsf{k}, \mathsf{m})$ *without counting that of oracles which is accessed by* $\mathsf{Enc}(\cdot)$*.*

2.7 Authentication

Authentication schemes achieve integrity. We define the models and securities of authentication schemes such as digital signatures, identification schemes, and message authentication codes.

2.7.1 Digital Signature

A digital signature consists of the following three polynomial-time algorithms (KGen,Sign,Vrfy): For a security parameter λ, let $\mathcal{M} = \mathcal{M}(\lambda)$ be a message space.

Key Generation KGen is a randomized algorithm which, on input a security parameter 1^λ, outputs a public key pk and a secret key sk.

Signing Sign is a randomized or deterministic algorithm which, on input a secret key sk and a message $\mathsf{m} \in \mathcal{M}$, outputs a signature σ.

Verification Vrfy is a deterministic algorithm which, on input a public key pk, a message $\mathsf{m} \in \mathcal{M}$, and a signature σ, outputs 1 or 0.

It is required that digital signatures meet correctness as follows: For all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ and all $\mathsf{m} \in \mathcal{M}$, we have $\mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}, \sigma) = 1$, where $\mathsf{Sign} \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$.

As a security of digital signatures, *existential unforgeability against chosen message attacks* (EUF-CMA security) and *strong unforgeability against chosen message attacks* (sUF-CMA security) are defined as follows:

Definition 2.21 (EUF-CMA security)**.** *A digital signature scheme* $\mathsf{DS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ *meets* EUF-CMA *security if for any PPT adversary* A *against* DS*, the advantage* $\mathsf{Adv}^{\text{euf-cma}}_{\mathsf{DS},\mathsf{A}}(\lambda) := \Pr[\mathsf{A}\ \textit{wins}]$ *is negligible in* λ*, where* $[\mathsf{A}\ \textit{wins}]$ *is the event that* A *wins the following game:*

Setup: *A challenger generates a key pair* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ *and sets* $L_{\mathsf{Sign}} \leftarrow \emptyset$*.*

Oracle Access: A *is allowed to access the following oracle:*

- $\mathsf{SIGN}(\mathsf{m})$*: Given a message query* $\mathsf{m} \in \mathcal{M}$*, it returns* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$ *and sets* $L_{\mathsf{Sign}} \leftarrow L_{\mathsf{Sign}} \cup \{\mathsf{m}\}$*.*

Output: A *outputs* (m^*, σ^*)*.* A *wins if* $\mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}^*, \sigma^*) = 1$ *and* $\mathsf{m}^* \notin L_{\mathsf{Sign}}$*.*

Definition 2.22 (sUF-CMA security)**.** *A digital signature scheme* $\mathsf{DS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ *meets* sUF-CMA *security if for any PPT adversary* A *against* DS*, the advantage* $\mathsf{Adv}^{\text{suf-cma}}_{\mathsf{DS},\mathsf{A}}(\lambda) := \Pr[\mathsf{A}\ \textit{wins}]$ *is negligible in* λ*, where* $[\mathsf{A}\ \textit{wins}]$ *is the event that* A *wins the following game:*

Setup: *A challenger generates* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ *and sets* $L_{\mathsf{Sign}} \leftarrow \emptyset$*.*

Oracle Access: A *is allowed to access the following oracle:*

- $\mathsf{SIGN}(\mathsf{m})$*: Given a message query* $\mathsf{m} \in \mathcal{M}$*, it returns* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$ *and sets* $L_{\mathsf{Sign}} \leftarrow L_{\mathsf{Sign}} \cup \{(\mathsf{m}, \sigma)\}$*.*

Output: A *outputs* (m^*, σ^*)*.* A *wins if* $\mathsf{Vrfy}(\mathsf{pk}, \mathsf{m}^*, \sigma^*) = 1$ *and* $(\mathsf{m}^*, \sigma^*) \notin L_{\mathsf{Sign}}$*.*

2.7.2 Identification Scheme

An identification scheme consists of the following four polynomial-time algorithms (KGen,P,$\mathcal{C}$,V):

Key Generation KGen is a randomized algorithm which, on input a security parameter 1^λ, outputs a public key pk and a secret key sk.

Prover $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$ is split into the following two polynomial-time algorithms P_1 and P_2:

- P_1 is a randomized algorithm which, on input a secret key sk, outputs a commitment $\mathsf{W} \in \mathcal{W}$ and a state st.
- P_2 is a randomized algorithm which, on input a secret key sk, a commitment $\mathsf{W} \in \mathcal{W}$, a challenge $\mathsf{c} \in \mathcal{C}$, and a state st, outputs a response $\mathsf{Z} \in \mathcal{Z}$.

Verifier V is a deterministic algorithm which, on input a public key pk, a commitment $\mathsf{W} \in \mathcal{W}$, a challenge $\mathsf{c} \in \mathcal{C}$, and a response $\mathsf{Z} \in \mathcal{Z}$, outputs 1 or 0.

We define the transcript oracle $\mathsf{Trans}^{ids}(\mathsf{sk})$ of identification schemes by following [1]. Namely, given a secret key sk, it returns a transcript $(\mathsf{W}, \mathsf{c}, \mathsf{Z}) \in \mathcal{W} \times \mathcal{C} \times \mathcal{Z} \cup \{(\perp, \perp, \perp)\}$. It computes a transcript by using a real interaction between a prover P and a verifier $\mathcal{C}$, and returns $(\perp, \perp, \perp)$ if $\mathsf{Z} = \perp$. Concretely, $\mathsf{Trans}^{ids}(\mathsf{sk})$ does the following:

1. $(\mathsf{W}, \mathsf{st}) \leftarrow \mathsf{P}_1(\mathsf{sk})$.
2. $\mathsf{c} \xleftarrow{U} \mathcal{C}$.
3. $\mathsf{Z} \leftarrow \mathsf{P}_2(\mathsf{sk}, \mathsf{W}, \mathsf{c}, \mathsf{st})$.
4. If $\mathsf{Z} = \perp$, return $(\perp, \perp, \perp)$.
5. Return $(\mathsf{W}, \mathsf{c}, \mathsf{Z})$.

An identification scheme meets δ-correctness if for all $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$, the following holds:

- For all $(\mathsf{c}, \mathsf{st}) \leftarrow \mathsf{P}_1(\mathsf{sk})$, all $\mathsf{c} \in \mathcal{C}$, and all $\mathsf{Z} \leftarrow \mathsf{P}_2(\mathsf{sk}, \mathsf{W}, \mathsf{c}, \mathsf{st})$ such that $\mathsf{Z} \neq \perp$, we have $1 \leftarrow \mathsf{V}(\mathsf{pk}, \mathsf{W}, \mathsf{c}, \mathsf{Z})$.
- We have $\Pr[\mathsf{Z} = \perp \mid (\mathsf{W}, \mathsf{c}, \mathsf{Z}) \leftarrow \mathsf{Trans}^{ids}(\mathsf{sk})] \leq \delta$.

As properties of identification schemes, we define the following.

Definition 2.23 (Commitment-Recoverable)**.** *An identification scheme* $\mathsf{IDS} = (\mathsf{KGen}, \mathsf{P}, \mathcal{C}, \mathsf{V})$ *is commitment-recoverable if for any* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$*, any* $\mathsf{c} \in \mathcal{C}$*, and any* $\mathsf{Z} \in \mathcal{Z}$*, there exists a unique* $\mathsf{W} \in \mathcal{W}$ *such that* $\mathsf{V}(\mathsf{pk}, \mathsf{W}, \mathsf{c}, \mathsf{Z}) = 1$*. This unique* W *can be publicly computed using a polynomial-time algorithm* Rec *which, on input* pk*,* $\mathsf{c} \in \mathcal{C}$*, and* $\mathsf{Z} \in \mathcal{Z}$*, outputs the unique* $\mathsf{W} \in \mathcal{W}$*.*

Definition 2.24 (Non-Abort Honest-Verifier Zero-Knowledge)**.** *An identification scheme* $\mathsf{IDS} = (\mathsf{KGen}, \mathsf{P}, \mathcal{C}, \mathsf{V})$ *meets* $\varepsilon_{\mathsf{zk}}$*-perfect* naHVZK *if there exists a PPT algorithm* S *which, on input a public key* pk*, outputs a transcript* $(\mathsf{W}, \mathsf{c}, \mathsf{Z})$ *such that*

- *The difference between distributions of* $(\mathsf{W}, \mathsf{c}, \mathsf{Z}) \leftarrow \mathsf{S}(\mathsf{pk})$ *and* $(\mathsf{W}', \mathsf{c}', \mathsf{Z}') \leftarrow \mathsf{Trans}(\mathsf{sk})$ *is at most* $\varepsilon_{\mathsf{zk}}$*.*
- *The distribution of* c *from* $(\mathsf{W}, \mathsf{c}, \mathsf{Z}) \leftarrow \mathsf{S}(\mathsf{pk})$ *with* $\mathsf{c} \neq \perp$ *is uniformly random in* $\mathcal{C}$

Definition 2.25 (Min-Entropy)**.** *If the most likely value of random variables* W *chosen from a distribution* D *occurs with probability* $2^{-\alpha}$ *for a random value* W*, we write* $\mathsf{min\text{-}entropy}(\mathsf{W} \mid \mathsf{W} \leftarrow D) = \alpha$*. An identification scheme* $\mathsf{IDS} = (\mathsf{KGen}, \mathsf{P}, \mathcal{C}, \mathsf{V})$ *meets* α *bits* $\mathsf{min\text{-}entropy}$ *if for all* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$*, it holds that* $\Pr[\mathsf{min\text{-}entropy}(\mathsf{W} \mid (\mathsf{W}, \mathsf{st}) \leftarrow \mathsf{P}_1(\mathsf{sk})) \geq \alpha] \geq 1 - 2^{-\alpha}$*.*

Definition 2.26 (Computational Unique Response (CUR))**.** *An identification scheme* $\mathsf{IDS} = (\mathsf{KGen}, \mathsf{P}, \mathcal{C}, \mathsf{V})$ *meets* CUR *property if for any PPT adversary against* A*, the advantage* $\mathsf{Adv}^{\mathrm{cur}}_{\mathsf{IDS},\mathsf{A}}(\lambda)$ *is negligible in* λ*, where* $\mathsf{Adv}^{\mathrm{cur}}_{\mathsf{IDS},\mathsf{A}}(\lambda)$ *is defined as follows:*

$$\mathsf{Adv}^{\mathrm{cur}}_{\mathsf{IDS},\mathsf{A}}(\lambda) := \Pr\left[\begin{array}{l|l} \mathsf{V}(\mathsf{pk}, \mathsf{W}, \mathsf{c}, \mathsf{Z}) = 1 \wedge & (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda); \\ \mathsf{V}(\mathsf{pk}, \mathsf{W}, \mathsf{c}, \mathsf{Z}') = 1 \wedge & (\mathsf{W}, \mathsf{c}, \mathsf{Z}, \mathsf{Z}') \leftarrow \mathsf{A}(\mathsf{pk}) \\ \mathsf{Z} \neq \mathsf{Z}' & \end{array}\right].$$

In addition, we define lossy identification scheme which was introduced by Abdalla et al. in [1]. An identification scheme $\mathsf{IDS} = (\mathsf{KGen}, \mathsf{P}, \mathcal{C}, \mathsf{V})$ is a lossy identification scheme if there exists a PPT algorithm $\mathsf{LossyKGen}$ which, on input a security parameter 1^λ, outputs a public key $\mathsf{pk}_{\mathsf{ls}}$. In this thesis, we refer to $\mathsf{LIDS} = (\mathsf{KGen}, \mathsf{LossyKGen}, \mathsf{P}, \mathcal{C}, \mathsf{V})$ as a lossy identification scheme.

Then, two security notions of lossy identification schemes are defined as follows:

Definition 2.27 (Key-Indistinguishability)**.** *A lossy identification scheme* $\mathsf{LIDS} = (\mathsf{KGen}, \mathsf{LossyKGen}, \mathsf{P}, \mathcal{C}, \mathsf{V})$ *meets* $\mathsf{key\text{-}indistinguishability}$ *if for any PPT adversary* A *against* LIDS*, the advantage* $\mathsf{Adv}^{\text{key-ind}}_{\mathsf{LIDS},\mathsf{A}}(\lambda)$ *is negligible in* λ*, where* $\mathsf{Adv}^{\text{key-ind}}_{\mathsf{LIDS},\mathsf{A}}(\lambda)$ *is defined as follows:*

$$\begin{aligned}\mathsf{Adv}^{\text{key-ind}}_{\mathsf{LIDS},\mathsf{A}}(\lambda) := \Big| &\Pr[\mathsf{A}(\mathsf{pk}) \to 1 \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)] \\ &- \Pr[\mathsf{A}(\mathsf{pk}_{\mathsf{ls}}) \to 1 \mid \mathsf{pk}_{\mathsf{ls}} \leftarrow \mathsf{LossyKGen}(1^\lambda)]\Big|.\end{aligned}$$

Definition 2.28 (Lossy-Soundness)**.** *A lossy identification scheme* $\mathsf{LIDS} = (\mathsf{KGen}, \mathsf{LossyKGen}, \mathsf{P}, \mathcal{C}, \mathsf{V})$ *meets* $\varepsilon_{\mathsf{ls}}$-lossy-soundness *if for any PPT adversary* A *against* LIDS, $\Pr[\mathsf{Exp}^{\text{lossy-imp}}_{\mathsf{LIDS},\mathsf{A}} \to 1] \leq \varepsilon_{\mathsf{ls}}$ *holds, where the experiment* $\mathsf{Exp}^{\text{lossy-imp}}_{\mathsf{LIDS},\mathsf{A}}$ *is defined as follows:*

Step 1. $\mathsf{pk}_{\mathsf{ls}} \leftarrow \mathsf{LossyKGen}(1^\lambda)$

Step 2. $(\mathsf{W}^*, \mathsf{st}) \leftarrow \mathsf{A}(\mathsf{pk}_{\mathsf{ls}})$, $\mathsf{c} \xleftarrow{U} \mathcal{C}$, $\mathsf{Z}^* \leftarrow \mathsf{A}(\mathsf{st}, \mathsf{c})$.

Step 3. Return $\mathsf{V}(\mathsf{pk}_{\mathsf{ls}}, \mathsf{W}^*, \mathsf{c}, \mathsf{Z}^*)$.

2.7.3 Message Authentication Code

A message authentication code (MAC) consists of two polynomial time algorithms $(\mathsf{Tag}, \mathsf{Vrfy})$ with a key space $\mathcal{K} = \mathcal{K}(\lambda)$ and a message space $\mathcal{M} = \mathcal{M}(\lambda)$ for a security parameter λ.

Tagging Tag is an algorithm which, on input a secret key $\mathsf{k} \in \mathcal{K}$ and a message $\mathsf{m} \in \mathcal{M}$, outputs a tag t.

Verification Vrfy is a deterministic algorithm which, on input a secret key $\mathsf{k} \in \mathcal{K}$, a message m, and a tag t, outputs 1 or 0.

It is required that MAC schemes meet correctness as follows: A MAC scheme $\mathsf{MAC} = (\mathsf{Tag}, \mathsf{Vrfy})$ with a key space $\mathcal{K}$ and a message space $\mathcal{M}$ meets correctness if for all $\mathsf{k} \in \mathcal{K}$ and all $\mathsf{m} \in \mathcal{M}$, we have $\mathsf{Vrfy}(\mathsf{k}, \mathsf{m}, \mathsf{t}) = 1$, where $\mathsf{t} \leftarrow \mathsf{Tag}(\mathsf{k}, \mathsf{m})$.

Strong unforgeability against one-time chosen message attacks (sUF-OT-CMA security) of MACs is defined as follows.

Definition 2.29 (sUF-OT-CMA security)**.** *A MAC scheme* $\mathsf{MAC} = (\mathsf{Tag}, \mathsf{Vrfy})$ *with a key space* $\mathcal{K}$ *meets* sUF-OT-CMA *security if for any PPT adversary* A *against* MAC*, the advantage* $\mathsf{Adv}^{\text{suf-cma}}_{\mathsf{MAC},\mathsf{A}} := \Pr[\mathsf{A}\ \textit{wins}]$ *is negligible in* λ*, where* $[\mathsf{A}\ \textit{wins}]$ *is the event that* A *wins in the following game:*

Setup: *A challenger chooses a key* $\mathsf{k} \in \mathcal{K}$ *uniformly at random and sets* $L_{\mathsf{t}} \leftarrow \emptyset$ *and* $\mathsf{win} \leftarrow 0$.

Oracle Access: A *is allowed to access the following oracles:*

- $\mathsf{TAG}(\mathsf{m})$*: If* $L_{\mathsf{t}} \neq \emptyset$*, a tagging oracle* $\mathsf{TAG}(\mathsf{m})$ *returns* $\perp$*. Otherwise, it returns* $\mathsf{t} \leftarrow \mathsf{Tag}(\mathsf{k}, \mathsf{m})$ *and sets* $L_{\mathsf{t}} \leftarrow L_{\mathsf{t}} \cup \{(\mathsf{m}, \mathsf{t})\}$.
- $\mathsf{VRFY}(\mathsf{m}, \mathsf{t})$*: Given a message and a tag* (m, t)*, a verification oracle* $\mathsf{VRFY}(\mathsf{m}, \mathsf{t})$ *returns* $b \leftarrow \mathsf{Vrfy}(\mathsf{k}, \mathsf{m}, \mathsf{t})$*. If* $b = 1$ *and* $(\mathsf{m}, \mathsf{t}) \notin L_{\mathsf{t}}$*, it sets* $\mathsf{win} \leftarrow 1$.

Final: A *wins if* $\mathsf{win} = 1$.

2.8 Encryption and Authentication

We defined signcryption which is cryptography with both functionalities of PKE and DSs.

A signcryption scheme consists of five polynomial-time algorithms (Setup, KGen_R, KGen_S,SC,USC): For a security parameter λ, let $\mathcal{M} = \mathcal{M}(\lambda)$ be a message space.

Setup Setup is an algorithm which, on input a security parameter 1^λ, outputs a public parameter pp.

Receiver's Key Generation KGen_R is a randomized algorithm which, on input a public parameter pp, outputs a receiver's public key pk_R and a receiver's secret key sk_R.

Sender's Key Generation KGen_S is a randomized algorithm which, on input a public parameter pp, outputs a sender's public key pk_S and a sender's secret key sk_S.

Signcryption SC is an algorithm which, on input a public parameter pp, a receiver's public key pk_R, a sender's secret key sk_S, and a message $\mathsf{m} \in \mathcal{M}$, outputs a ciphertext ct.

Unsigncryption USC is a deterministic algorithm which, on input a public parameter pp, a sender's public key pk_S, a receiver's secret key sk_R, and a ciphertext ct, outputs a message $\mathsf{m} \in \mathcal{M}$ or an invalid symbol $\perp$.

We require that signcryption schemes meet correctness: A signcryption scheme (Setup,KGen_R,KGen_S,SC, USC) meets correctness if for all $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, all $(\mathsf{pk}_R, \mathsf{sk}_R) \leftarrow \mathsf{KGen}_R(\mathsf{pp})$, all $(\mathsf{pk}_S, \mathsf{sk}_S) \leftarrow \mathsf{KGen}_S(\mathsf{pp})$, and all $\mathsf{m} \in \mathcal{M}$, we have $\mathsf{m} = \mathsf{USC}(\mathsf{pp}, \mathsf{pk}_S, \mathsf{sk}_R, \mathsf{ct})$ with overwhelming probability, where $\mathsf{ct} \leftarrow \mathsf{SC}(\mathsf{pp}, \mathsf{pk}_R, \mathsf{sk}_S, \mathsf{m})$.

Definition 2.30 (MU-IND-iCCA security)**.** *A signcryption scheme* $\mathsf{SCS} = (\mathsf{Setup}, \mathsf{KGen}_R, \mathsf{KGen}_S, \mathsf{SC}, \mathsf{USC})$ *meets* MU-IND-iCCA *security if for any PPT adversary* A*, the advantage* $\mathsf{Adv}^{\text{mu-ind-icca}}_{\mathsf{SCS},\mathsf{A}}(\lambda) := |\Pr[\mathsf{A}\ \textit{wins}] - 1/2|$ *is negligible in* λ*, where* $[\mathsf{A}\ \textit{wins}]$ *is the event that* A *wins in the following game.*

Setup: *A challenger generates* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ *and* $(\mathsf{pk}_R, \mathsf{sk}_R) \leftarrow \mathsf{Setup}_R(\mathsf{pp})$*, and sends* $(\mathsf{pp}, \mathsf{pk}_R)$ *to* A*.*

Queries 1: *Given a sender's public key and a ciphertext* $(\mathsf{pk}_S, \mathsf{ct})$*, the unsigncrypt oracle* $\mathsf{USC}^{\mathsf{O}}$ *returns* $\mathsf{m}/\perp \leftarrow \mathsf{USC}(\mathsf{pp}, \mathsf{pk}_S, \mathsf{sk}_R, \mathsf{ct})$*.*

Challenge: *When* A *submits* $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{pk}^*_S, \mathsf{sk}^*_S)$ *such that* $|\mathsf{m}_0| = |\mathsf{m}_1|$*, the challenger chooses* $b \in \{0, 1\}$ *uniformly at random and returns a challenge ciphertext* $\mathsf{ct}^* \leftarrow \mathsf{SC}(\mathsf{pp}, \mathsf{pk}_R, \mathsf{sk}^*_S, \mathsf{m}_b)$*,*

Queries 2: *Given a sender's public key and a ciphertext* $(\mathsf{pk}_S, \mathsf{ct})$*,* $\mathsf{USC}^{\mathsf{O}}$ *returns* $\mathsf{m}/\perp \leftarrow \mathsf{USC}(\mathsf{pp}, \mathsf{pk}_S, \mathsf{sk}_R, \mathsf{ct})$*.* A *is not allowed to issue a query* $(\mathsf{pk}_S, \mathsf{ct})$ *such that* $(\mathsf{pk}_S, \mathsf{ct}) = (\mathsf{pk}_S^*, \mathsf{ct}^*)$

Output: A *outputs the guessing bit* $b' \in \{0, 1\}$*.* A *wins if* $b = b'$*.*

Definition 2.31 (MU-sUF-iCMA security)**.** *A signcryption scheme* $\mathsf{SCS} = (\mathsf{Setup}, \mathsf{KGen}_R, \mathsf{KGen}_S, \mathsf{SC}, \mathsf{USC})$ *meets* MU-sUF-iCMA *security if for any PPT adversary* A*, the advantage* $\mathsf{Adv}^{\text{mu-suf-icma}}_{\mathsf{SCS},\mathsf{A}}(\lambda) := \Pr[\mathsf{A}\ \textit{wins}]$ *is negligible in* λ*, where* $[\mathsf{A}\ \textit{wins}]$ *is the event that* A *wins in the following game.*

Setup: *A challenger generates* $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ *and* $(\mathsf{pk}_S, \mathsf{sk}_S) \leftarrow \mathsf{KGen}_S(\mathsf{pp})$*, and sets* $L_{\mathsf{SC}} \leftarrow \emptyset$*. It sends* $(\mathsf{pp}, \mathsf{pk}_S)$ *to* A*.*

Queries: *Given a receiver's public key and a message* $(\mathsf{pk}_R, \mathsf{m})$*, the signcrypt oracle* SC^{O} *returns* $\mathsf{ct} \leftarrow \mathsf{SC}(\mathsf{pp}, \mathsf{pk}_R, \mathsf{sk}_S, \mathsf{m})$ *and sets* $L_{\mathsf{SC}} \leftarrow L_{\mathsf{SC}} \cup \{(\mathsf{pk}_R, \mathsf{m}, \mathsf{ct})\}$*.*

Output: A *outputs* $(\mathsf{pk}_R^*, \mathsf{sk}_R^*, \mathsf{ct}^*)$*.* A *wins if* $\mathsf{USC}(\mathsf{pp}, \mathsf{pk}_S, \mathsf{sk}_R^*, \mathsf{ct}^*) \rightarrow \mathsf{m}^* \neq \perp$ *and* $(\mathsf{pk}_R^*, \mathsf{m}^*, \mathsf{ct}^*) \notin L_{\mathsf{SC}}$ *hold.*

Chapter 3

Quantum-Secure Public Key Encryption

3.1 Background of Selective Opening Security

The security model of selective-opening (SO) security captures a situation in which an adversary gets secret information of many ciphertexts. In fact, there are cases in which adversaries obtain messages and randomness of ciphertexts because of side-channel attacks and weakness in system's design or implementation. This security is one of the most important securities of PKE in the multi-user setting. There are the following works related to SO security: Generic constructions of PKE [15, 59, 60], number theory-based PKE [45, 63, 65], hybrid encryption [92, 61, 93], identity-based encryption [19, 87], and lattice-based PKE [28, 89].

Furthermore, SO security is roughly classified as simulation-based SO (SIM-SO) security and indistinguishability-based SO (IND-SO) security. In this paper, we consider SIM-SO security against chosen ciphertext attacks called SIM-SO-CCA security, since it seems that it is harder to achieve SIM-SO security [20, 65] and several works have aimed at proposing SIM-SO-CCA secure PKE schemes [45, 60, 92, 61, 89, 93]. Hence, it is natural to consider SIM-SO-CCA security as our goal in the multi-user setting.

3.2 Contribution

Our goal is to present SIM-SO-CCA secure PKE schemes obtained from KEM schemes in the quantum random oracle model (QROM) or the quantum ideal cipher model (QICM). Our main motivation is that we would like to transform any PKE/KEM schemes submitted to the post-quantum cryptography standardization to SIM-SO-CCA secure PKE without loss of efficiency in terms of key-size, ciphertext-size, and time-complexity. To the best of our knowledge, there is no work which researched quantum-secure PKE with (SIM-)SO-CCA

security.

In the classical random oracle model, classical ideal cipher model, or the standard model (i.e., the model without random oracles or ideal ciphers), several SIM-SO-CCA secure PKE schemes constructed from KEM schemes have been studied. Liu and Paterson proposed a SIM-SO-CCA secure PKE scheme constructed from any KEM scheme secure against tailored constrained chosen ciphertext attacks and any strengthened cross authentication code (XAC) [92]. Heuer et al. proposed a SIM-SO-CCA secure construction by combining any KEM secure against plaintext checking attacks and any one-time secure message authentication code (MAC) [60]. Heuer and Poettering proved that a PKE scheme in the KEM/DEM framework meets SIM-SO-CCA security in the classical ideal cipher model if the underlying KEM satisfies IND-CCA security and the underlying DEM satisfies both of simulatability and one-time integrity of chosen ciphertext attacks, which is called OT-INT-CTXT security [61]. Lyu et al. proposed a tightly secure PKE starting from any KEM scheme meeting both of multi-encapsulation pseudorandom security and random encapsulation rejection security, and any strengthened XAC [93]. Table 3.1 shows the above primitives and security models of the existing constructions.

In the QROM or QICM, how to construct PKE schemes meeting SIM-SO-CCA security is not obvious because of the following reason: In the classical random oracle model or classical ideal cipher model, the security proof of existing schemes [92, 61] utilize the lists of query/response pairs submitted to random oracles or ideal ciphers. In the QROM and QICM, we cannot use such lists, since it is impossible to record query/response pairs in principle due to the quantum no-cloning theorem. Hence, it is worth to consider secure PKE schemes in the models where quantum queries are issued.

As for the PKE schemes obtained from KEM schemes in the standard model [92, 93], ciphertext-size and time-complexity of encryption and decryption algorithms linearly depend on the bit-length of messages. Since we are aiming at constructing practical PKE schemes, we do not focus on these schemes in this paper due to the lack of efficiency in terms of ciphertext-size and time-complexity.

In this paper, we propose two constructions of SIM-SO-CCA secure PKE schemes from KEM schemes and symmetric key encryption (SKE) schemes. The details are explained as follows:

1. The first scheme PKE_1^{hy} is the KEM/DEM scheme [35]. We prove that this scheme meets SIM-SO-CCA security in the QICM if the underlying KEM scheme satisfies IND-CCA security, and the underlying DEM scheme satisfies both of simulatability [61] and one-time integrity of chosen ciphertext attacks (OT-INT-CTXT security) [17]. The advantage of this scheme is that we can apply any IND-CCA secure KEM scheme such as any PKE/KEM schemes submitted to the post-quantum cryptography standardization, and we can obtain a SIM-SO-CCA secure PKE

Scheme	Primitives	Security Model
[92]	IND-tCCCA secure KEM	Standard Model
	XAC	
[60]	OW-PCA secure KEM	Random Oracle Model
	sUF-OT-CMA secure MAC	
[61]	IND-CCA secure KEM	Ideal Cipher Model
	Simulatable DEM	
[93]	mPR-CCCA and RER secure KEM	Standard Model
	XAC	
PKE_1^{hy}	IND-CCA secure KEM	Quantum
	Simulatable DEM	Ideal Cipher Model
PKE_2^{hy}	FO-based KEM (from IND-CPA secure PKE)	Quantum Random Oracle Model
	sUF-OT-CMA secure MAC	

Table 3.1: SIM-SO-CCA secure PKE constructed from KEM schemes: IND-tCCCA means indistinguishability against tailored constrained chosen ciphertext attacks. IND-PCA means indistinguishability against plaintext checking attacks. mPR-CCCA means multi-encapsulation pseudorandom security against constrained chosen ciphertext attacks. RER means random encapsulation rejection security. XAC means cross authentication code. IND-CPA means indistinguishability against chosen message attacks. FO-based KEM means $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}_m^{\not\perp}$, $\mathsf{QFO}^{\not\perp}$, and $\mathsf{QFO}_m^{\not\perp}$. Standard model denotes the security model without random oracles and ideal ciphers.

schemes in the QICM. In addition, almost all standardized DEM schemes satisfy simulatability and OT-INT-CTXT security. Hence, we can realize concrete PKE schemes in the QICM.

2. The second one PKE_2^{hy} is a concrete scheme constructed from any FO-based KEM scheme such as $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}_m^{\not\perp}$, $\mathsf{QFO}^{\not\perp}$, and $\mathsf{QFO}_m^{\not\perp}$, which are categorized in [64], and any MAC meeting strong unforgeability against one-time chosen message attacks called sUF-OT-CMA security. The underlying KEM scheme is FO-based KEM with implicit rejection. That is, these schemes output a random key which is not encapsulated if a given ciphertext is invalid. We require that the underlying PKE in $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}_m^{\not\perp}$, $\mathsf{QFO}^{\not\perp}$, or $\mathsf{QFO}_m^{\not\perp}$ is injective and satisfies indistinguishability against chosen plaintext attacks called IND-CPA security. In addition, many KEM schemes submitted to the NIST post-quantum cryptography standardization are classified as $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}_m^{\not\perp}$, $\mathsf{QFO}^{\not\perp}$, or $\mathsf{QFO}_m^{\not\perp}$. Hence, the advantage of PKE_2^{hy} is that a lot of PKE/KEM schemes submitted to the post-quantum standardization can satisfy SIM-SO-CCA security without demanding any special property such as simulatability, for the

underlying SKE.

The difference between PKE_1^{hy} and PKE_2^{hy} is given as follows:

- Any $\mathsf{IND\text{-}CCA}$ secure KEM resistant to quantum computing can be applied to PKE_1^{hy} while a particular KEM scheme (i.e., $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}_m^{\not\perp}$, $\mathsf{QFO}^{\not\perp}$, or $\mathsf{QFO}_m^{\not\perp}$) are applied to PKE_2^{hy}.
- PKE_1^{hy} requires that the underlying DEM scheme satisfies a special property such as simulatability while PKE_2^{hy} does not require that the underlying MAC satisfies any special property.

In Sections 3.3 and 3.4, we describe concrete primitives which can be applied to PKE_1^{hy} and PKE_2^{hy}, respectively.

3.3 KEM/DEM framework

In this section, we focus on hybrind encryption scheme PKE_1^{hy} with the KEM/DEM approach [35], which is constructed from any $\mathsf{IND\text{-}CCA}$ secure KEM and any DEM with both simulatability and $\mathsf{OT\text{-}INT\text{-}CTXT}$ security, and prove that PKE_1^{hy} meets $\mathsf{SIM\text{-}SO\text{-}CCA}$ security in the QICM. This security proof is based on the proof of Theorem 2 in [61]. However, it is not obvious that it satisfies $\mathsf{SIM\text{-}SO\text{-}CCA}$ security in the QICM because the proof in [61] uses the list of query/response pairs issued to ideal cipher oracles and we cannot apply this technique due to the quantum no-cloning theorem. To resolve this problem, we utilize a semi-classical oracle to check whether quantum queries meeting a condition are submitted to ideal cipher oracles or not, instead of using the list of ideal cipher oracles.

It is possible to construct concrete $\mathsf{SIM\text{-}SO\text{-}CCA}$ secure PKE schemes in the QICM because several DEM schemes such as CTR-DEM, CBC-DEM, CCM-DEM, and hidden-shift CBC-DEM meet simulatability [61, 8]. As a quantum (ideal) block cipher, hidden-shift Even-Mansour ciphers in [8] may be used.

To construct PKE_1^{hy} with a message space $\mathcal{M}$, we use the following primitives: Let $\mathsf{KEM} = (\mathsf{KGen}^{asy}, \mathsf{Encap}, \mathsf{Decap})$ be a KEM scheme with a key space $\mathcal{K} = \mathcal{K}' \times \mathcal{K}''$ and a randomness space $\mathcal{R}^{asy}$. Let $\mathsf{DEM} = (\mathsf{Enc}^{sym}, \mathsf{Dec}^{sym})$ be a DEM scheme with a key space $\mathcal{K} = \mathcal{K}' \times \mathcal{K}''$ and a message space $\mathcal{M}$.

The PKE scheme $\mathsf{PKE}_1^{hy} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is described as follows:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: Generate $(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}) \leftarrow \mathsf{KGen}^{asy}(1^\lambda)$ and output $\mathsf{pk} := \mathsf{pk}^{asy}$ and $\mathsf{sk} := \mathsf{sk}^{asy}$.
- $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$: Encrypt a message $\mathsf{m} \in \mathcal{M}$ as follows:
  1. $(e, \mathsf{k}) \leftarrow \mathsf{Encap}(\mathsf{pk}^{asy})$, and $d \leftarrow \mathsf{Enc}^{sym}(\mathsf{k}, \mathsf{m})$.
  2. Output $\mathsf{ct} := (e, d)$.

- $\mathsf{m}/\perp \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: Decrypt a ciphertext $\mathsf{ct} = (e, d)$ as follows:
  1. $\mathsf{k} \leftarrow \mathsf{Decap}(\mathsf{sk}^{asy}, e)$.
  2. Output $\mathsf{m}' \leftarrow \mathsf{Dec}^{sym}(\mathsf{k}, d)$ if $\mathsf{k} \neq \perp$, and output $\perp$ otherwise.

Theorem 3.1. *If a KEM scheme* KEM *meets* IND-CCA *security, and a* $(\mathcal{K}, \mathcal{X})$*-permutation-driven DEM scheme* DEM *with an oracle DEM scheme* (O.Enc, O.Dec) *for a domain* $\mathcal{X}$ *and a block cipher* E *meets both of* ϵ_{sim}*-simulatability and* OT-INT-CTXT *security, then* PKE_1^{hy} *satisfies* SIM-SO-CCA *security in the quantum ideal cipher model.*

Proof. Let A be a QPT adversary against PKE_1^{hy}. Let q_d be the number of accessing $\mathsf{DEC}(\cdot)$, and q_e be the total number of accessing $E^+(\cdot)$ and $E^-(\cdot)$. For $J \subseteq [n]$, let $K'_J := \{\mathsf{k}'_j \mid j \in J\}$. We write $E_{\mathsf{k}'}(\cdot) = E(\mathsf{k}', \cdot)$ for an ideal cipher E with a key k'.

For each $i \in \{0, 1, 2, 3, 4\}$, we consider a security game Game_i, and let W_i be the event that A outputs *out* such that $R(\mathcal{M}_{\mathrm{D}}, \mathsf{m}_1, \ldots, \mathsf{m}_n, I, out) = 1$ in Game_i.

Game_0: This game is the same as Real-SIM-SO-CCA security game. We have $\Pr[\mathsf{Exp}^{\text{real-so-cca}}_{\mathsf{PKE}^{hy},\mathsf{A}} \to 1] = \Pr[W_0]$. ■

Game_1: This game is the same as Game_1 except that DEC oracle returns $\perp$ if a query (e, d) such that $e \in \{e_i\}_{i \in [n] \setminus I}$ is submitted.

We show $|\Pr[W_0] - \Pr[W_1]| \leq n \cdot (\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathsf{D}_1}(\lambda) + \mathsf{Adv}^{\text{int-ctxt}}_{\mathsf{DEM},\mathsf{F}}(\lambda))$. Let Bad be the event that A submits a ciphertext query (e, d) such that $e \in \{e_i\}_{i \in [n] \setminus I}$ and $\mathsf{Dec}(\mathsf{sk}, (e, d)) \neq \perp$. Unless Bad occurs, Game_1 is identical to Game_0. Besides, we consider the following events: Let Bad_1 be the event that Bad happens in Game_0, and let Bad_2 be the same event as Bad_1 except that for $i \in [n] \setminus I$, keys k_i are chosen uniformly at random.

To show $|\Pr[\mathsf{Bad}_1] - \Pr[\mathsf{Bad}_2]| \leq n \cdot \mathsf{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathsf{D}_1}(\lambda)$, we construct a PPT algorithm D_1 breaking the IND-CCA security of KEM in the following way: At the beginning of the security game, D_1 takes pk^{asy} as input. It sets $i^* \xleftarrow{U} [n]$ and chooses a random polynomial f_E of degree $2q_e - 1$ over $GF(2^\kappa)$ uniformly at random as a $2q_e$-wise independent hash function, where κ is the bit-length of elements in $\mathcal{K}' \times \mathcal{X}$. Then, it sets $I \leftarrow \emptyset$ and sends $\mathsf{pk} := \mathsf{pk}^{asy}$ to A. When A submits $\mathcal{M}_{\mathrm{D}}$, it does the following for each $i \in [n]$:

1. If $i = i^*$, request a challenge $(e_{i^*}, \mathsf{k}_{i^*})$ in IND-CCA game. Otherwise, compute $(e_i, \mathsf{k}_i) \leftarrow \mathsf{Encap}(\mathsf{pk}; r_i)$, where $r_i \in \mathcal{R}^{asy}$ is sampled at random.
2. $d_i \leftarrow \mathsf{Enc}^{sym}(\mathsf{k}_i, \mathsf{m}_i)$, where $\mathsf{m}_i \leftarrow \mathcal{M}_D$.

Then, it returns $((e_i, d_i))_{i \in [n]}$ to A. D_1 simulates oracles as follows:

- $E^+(\mathsf{k}', \alpha)$: Return $f_E(\mathsf{k}', \alpha)$.

- $E^-(\mathsf{k}', \beta)$: Compute the set R of all roots of the polynomial $f_E(\mathsf{k}', \cdot) - \beta$ and return $\alpha \in R$.

- $\mathsf{DEC}(\mathsf{ct})$: Take $\mathsf{ct} = (e, d)$ as input. In the case of $e = e_{i^*}$, halt and output 1 if $\perp \neq \mathsf{Dec}^{sym}(\mathsf{k}_{i^*}, d)$, and return $\perp$ otherwise. In the case of $e \neq e_{i^*}$, submit e to the given decapsulation oracle and receive k. Return $\perp$ if $\mathsf{k} = \perp$, and return $\mathsf{Dec}^{sym}(\mathsf{k}, d)$ if $\mathsf{k} \neq \perp$.

- $\mathsf{OPEN}(i)$: Set $I \leftarrow I \cup \{i\}$. Abort if $i = i^*$. Return (m_i, r_i) if $i \neq i^*$.

Note that quantum ideal ciphers E^+ and E^- can be simulated by using $2q_e$-wise independent hash functions from Theorem 6.1 in [137].

When A outputs *out*, D_1 outputs 0 if Bad does not happen. D_1 simulates the view of A completely. If A submits a decryption query meeting the condition of Bad, it can distinguish the two games, and D_1 breaks IND-CCA security with at least probability $|\Pr[\mathsf{Bad}_1] - \Pr[\mathsf{Bad}_2]|/n$. Thus, we get the bound.

To show $\Pr[\mathsf{Bad}_2] \leq n \cdot \mathsf{Adv}^{\text{int-ctxt}}_{\mathsf{DEM},\mathsf{F}}(\lambda)$, we construct a PPT algorithm F breaking OT-INT-CTXT security as follows: It is given the two oracles $\mathsf{ENC}(\cdot)$ and $\mathsf{VRFY}(\cdot)$ in OT-INT-CTXT game. At the beginning of the security game, F generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ and chooses $i^* \in [n]$ uniformly at random. When A submits $\mathcal{M}_{\mathrm{D}}$, it does the following for each $i \in [n]$:

1. $(e_i, \mathsf{k}_i) \leftarrow \mathsf{Encap}(\mathsf{pk}; r_i)$, where $r_i \in \mathcal{R}^{asy}$ is sampled at random.

2. $\mathsf{m}_i \leftarrow \mathcal{M}_{\mathrm{D}}$.

3. If $i = i^*$, $d_{i^*} \leftarrow \mathsf{ENC}(\mathsf{m}_{i^*})$. Otherwise, $d_i \leftarrow \mathsf{Enc}^{sym}(\mathsf{k}_i, \mathsf{m}_i)$.

Then, it returns $((e_i, d_i))_{i \in [n]}$. F simulates oracles $E^+(\cdot,\cdot)$, $E^-(\cdot,\cdot)$, and $\mathsf{OPEN}(\cdot)$ in the same way as the above algorithm D_1. $\mathsf{DEC}(\cdot)$ is simulated as follows: If $e = e_{i^*}$ for a given $\mathsf{ct} = (e, d)$, it submits (e, d) to $\mathsf{VRFY}(\cdot)$. F halts if it returns 1, and returns $\perp$ otherwise. If $e \neq e_{i^*}$, F computes $\mathsf{k} \leftarrow \mathsf{Decap}(\mathsf{sk}^{asy}, e)$ and returns $\mathsf{Dec}^{sym}(\mathsf{k}, d) \in \mathcal{M} \cup \{\perp\}$. When A outputs *out*, F aborts this game if Bad does not happen.

The success condition of F is identical to the condition that Bad occurs. Hence, F wins in OT-INT-CTXT game if A outputs a ciphertext query (e, d) such that $e = e_{i^*}$ and oracle $\mathsf{VRFY}(d)$ returns 1. The success probability of F is at least $\Pr[\mathsf{Bad}_2]/n$.

Therefore, we have $|\Pr[W_0] - \Pr[W_1]| \leq n \cdot (\mathsf{Adv}^{\text{ind-cca}}_{\mathsf{PKE}^{hy},\mathsf{D}_1}(\lambda) + \mathsf{Adv}^{\text{int-ctxt}}_{\mathsf{DEM},\mathsf{F}}(\lambda))$ in the straightforward way. ∎

$\ddot{E}^+$ (resp. $\ddot{E}^-$) is an ideal cipher oracle such that $\ddot{E}^+(\mathsf{k}', \alpha)$ (resp. $\ddot{E}^-(\mathsf{k}', \beta)$) is sampled from $\mathcal{X}$ uniformly at random if $\mathsf{k}' \in \{\mathsf{k}'_i\}_{i \in [n] \setminus I}$, and $\ddot{E}^+(\mathsf{k}', \alpha) = E^+(\mathsf{k}', \alpha)$ (resp. $\ddot{E}^-(\mathsf{k}', \beta) = E^-(\mathsf{k}', \beta)$) holds otherwise.

Game_2: This game is the same as Game_1 except that at the beginning of the security game, the challenger computes $(e_i, \mathsf{k}_i) \leftarrow \mathsf{Encap}(\mathsf{pk})$ for $i \in [n]$

$(\mathsf{k}_i = (\mathsf{k}'_i, \mathsf{k}''_i))$, and oracles E^+ and E^- are replaced by $\ddot{E}^+ \backslash S$ and $\ddot{E}^- \backslash S$ for $S = \{\mathsf{k}'_i\}_{i \in [n] \backslash I}$, respectively.

We show $|\Pr[W_1] - \Pr[W_2]| \leq 2\sqrt{nq \cdot \mathsf{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathsf{D}_2}(\lambda)} + 4q\sqrt{n/|\mathcal{K}'|}$. Let Bad' be the event that a semi-classical oracle O^{SC}_S returns $|1\rangle$ when A submits a query to an oracle $E^+(\cdot,\cdot)$ or $E^-(\cdot,\cdot)$. Besides, we consider the following events: Let Bad'_1 be the event that Bad' happens in Game_1, and let Bad'_2 be the same event as Bad'_1 except that for $i \in [n] \backslash I$, keys k_i are chosen uniformly at random. From Proposition 2.1 and the hybrid argument, we have $|\Pr[W_1] - \Pr[W_2]| \leq 2\sqrt{q \cdot \Pr[\mathsf{Bad}'_1]} \leq 2\sqrt{q\left|\Pr[\mathsf{Bad}'_1] - \Pr[\mathsf{Bad}'_2]\right| + q \cdot \Pr[\mathsf{Bad}'_2]}$.

We show $\left|\Pr[\mathsf{Bad}'_1] - \Pr[\mathsf{Bad}'_2]\right| \leq n \cdot \mathsf{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathsf{D}_2}(\lambda)$ by constructing a PPT algorithm D_2 breaking IND-CCA security. Notice that running $(e_i, \mathsf{k}_i) \leftarrow \mathsf{Encap}(\mathsf{pk}; r_i)$ at the beginning of the game is a conceptual modification. D_2 is constructed as follows: Given $(\mathsf{pk}^{asy}, e^*, \mathsf{k}^*)$, it chooses $i^* \in [n]$ uniformly at random, sets $(e_{i^*}, \mathsf{k}_{i^*}) := (e^*, \mathsf{k}^*)$, and generates $(e_i, \mathsf{k}_i) \leftarrow \mathsf{Encap}(\mathsf{pk}^{asy}; r_i)$ for all $i \in [n] \backslash \{i^*\}$, where r_i is sampled from $\mathcal{R}^{asy}$ at random. And then, it sets $I \leftarrow \emptyset$ and sends $\mathsf{pk} := \mathsf{pk}^{asy}$ to A. When A submits $\mathcal{M}_{\mathrm{D}}$, it samples $\mathsf{m}_i \leftarrow \mathcal{M}_{\mathrm{D}}$ and computes $d_i \leftarrow \mathsf{Enc}^{sym}(\mathsf{k}_i, \mathsf{m}_i)$ for $i \in [n]$. And then, it returns $((e_i, d_i))_{i \in [n]}$ to A. When A issues a quantum query $\sum_{\mathsf{k}' \in \mathcal{K}', x \in \mathcal{X}} \psi_{\mathsf{k}',x} |\mathsf{k}', x\rangle$ to E^+ or E^-, D_2 submits $\sum_{\mathsf{k}' \in \mathcal{K}', x \in \mathcal{X}} \psi_{\mathsf{k}',x} |\mathsf{k}', x\rangle |0\rangle$ to a semi-classical oracle O^{SC}_S. It halts and outputs 1 if O^{SC}_S returns a quantum superposition state $\sum_{\mathsf{k}' \in \mathcal{K}', x \in \mathcal{X}} \psi'_{\mathsf{k}',x} |\mathsf{k}', x\rangle |1\rangle$. It returns a quantum state by accessing E^+ or E^- otherwise. In addition, D_2 simulates the following oracles:

- $\mathsf{DEC}(\mathsf{ct})$: Take $\mathsf{ct} = (e, d)$ as input. If $e \in \{e_i\}_{i \in [n] \backslash I}$, return $\perp$. If $e \notin \{e_i\}_{i \in [n] \backslash I}$, submit e to the given decapsulation oracle and receive k. Return $\perp$ if $\mathsf{k} = \perp$, and return $\mathsf{Dec}^{sym}(\mathsf{k}', d)$ if $\mathsf{k} \neq \perp$.
- $\mathsf{OPEN}(i)$: Set $I \leftarrow I \cup \{i\}$. If $i = i^*$, abort this game. If $i \neq i^*$, set $E_{\mathsf{k}'_i} \leftarrow \emptyset$ and return (m_i, r_i) if $i \neq i^*$.

When A outputs a value *out*, D_2 aborts this game if Bad' does not occur. Then, D_2 simulates the environment of A completely. If A submits a quantum query including the valid key k_i of e_i to E^+ or E^-, A can distinguish the two games, and D_2 breaks the IND-CCA security of KEM. The success probability of D_2 is at least $\left|\Pr[\mathsf{Bad}'_1] - \Pr[\mathsf{Bad}'_2]\right| / n$.

In addition, we have $\Pr[\mathsf{Bad}'_2] \leq 4nq_e/|\mathcal{K}'|$ from Proposition 2.2. Therefore, we obtain the following inequality

$$\begin{aligned} |\Pr[W_1] - \Pr[W_2]| &\leq 2\sqrt{q_e\left(n \cdot \mathsf{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathsf{D}_2}(\lambda) + \frac{4q_e n}{|\mathcal{K}'|}\right)} \\ &\leq 2\sqrt{nq_e \cdot \mathsf{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathsf{D}_2}(\lambda)} + 4q_e\sqrt{\frac{n}{|\mathcal{K}'|}}, \end{aligned}$$

and the proof is completed. ■

Game_3: This game is the same as Game_2 except that the game is aborted if the challenger generates $(e_i, (\mathsf{k}'_i, \mathsf{k}''_i)) \leftarrow \mathsf{Encap}(\mathsf{pk})$ such that $\mathsf{k}'_i \in K'_{[i-1]}$ for $i \in [n]$.

The probability of choosing $\mathsf{k}'_i \in K'_{[i-1]}$ by running $\mathsf{Encap}(\mathsf{pk})$ for $i \in [n]$ is at most $n^2/|\mathcal{K}'|$. Thus, we have $|\Pr[W_2] - \Pr[W_3]| \leq n^2/|\mathcal{K}'|$. ■

Game_4: This game is the same as Game_3 except that for all $i \in [n]$, we replace Enc^{sym} algorithm by $(\mathsf{Fake}, \mathsf{Make})$. Namely, the process of the challenger and OPEN oracle is modified as follows: Given $\mathcal{M}_{\mathrm{D}}$, the challenger runs $(d_i, \mathsf{st}_i) \leftarrow \mathsf{Fake}(\mathsf{k}''_i, |\mathsf{m}_i|)$ and returns (e_i, d_i) for each $i \in [n]$. In addition, OPEN oracle is modified as follows:

1. $I \leftarrow I \cup \{i\}$.
2. $m_i \leftarrow \mathcal{M}_{\mathrm{D}}$.
3. $\tilde{\pi} \leftarrow \mathsf{Make}(\mathsf{st}_i, \mathsf{m}_i)$ and oracles $\ddot{E}^+(\mathsf{k}'_i, \cdot), \ddot{E}^-(\mathsf{k}'_i, \cdot)$ follow this relation $\tilde{\pi}$.
4. Abort this game if $d_i \neq \mathsf{O.Enc}^{E_{\mathsf{k}'_i}}(\mathsf{k}''_i, \mathsf{m}_i)$.
5. Return (m_i, r_i).

We show $|\Pr[W_3] - \Pr[W_4]| \leq n \cdot \epsilon_{sim}$. From the simulatability of DEM, A cannot distinguish d_i in the two games. In the process of OPEN oracle, we can define a relation $\tilde{\pi}$ in this phase since A cannot find $\mathsf{k}'_i \in \{\mathsf{k}'_i\}_{i \in [n] \setminus I}$ from the game-hop of Game_2. In addition, for each $i \in [n]$, the probability that the aborting event happens in OPEN oracle is negligible in λ from the simulatability of DEM. Hence, we have the inequality. ■

Finally, we prove $\Pr[\mathsf{Exp}^{\text{ideal-so-cca}}_{\mathsf{PKE}^{hy},\mathsf{S}} \to 1] = \Pr[W_4]$. We construct a simulator S in the following way: It is given $\overline{\mathsf{OPEN}}$ oracle in $\mathsf{Ideal\text{-}SIM\text{-}SO\text{-}CCA}$ game. At the beginning of the security game, S generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ and $(e_i, \mathsf{k}_i) \leftarrow \mathsf{Encap}(\mathsf{pk}; r_i)$ for $i \in [n]$. When A submits $\mathcal{M}_{\mathrm{D}}$, it receives $|\mathsf{m}_i|$ from the challenger of $\mathsf{Ideal\text{-}SIM\text{-}SO\text{-}CCA}$ game, generates $d_i \leftarrow \mathsf{Fake}(\mathsf{k}''_i, |\mathsf{m}_i|)$ for $i \in [n]$, and returns $((e_i, d_i))_{i \in [n]}$. In the same way as the game-hop of Game_4, S simulates $\ddot{E}^+$ and $\ddot{E}^-$ by using a $2q_e$-wise independent hash function and algorithms $(\mathsf{Fake}, \mathsf{Make})$. It simulates oracles $\mathsf{DEC}(\cdot)$ and $\mathsf{OPEN}(\cdot)$ as follows:

- $\mathsf{DEC}(\mathsf{ct})$: Take $\mathsf{ct} = (e, d)$ as input and do the following.
  1. Return $\perp$ if $e \in \{e_i\}_{i \in [n] \setminus I}$.
  2. $\mathsf{k} \leftarrow \mathsf{Decap}(\mathsf{sk}, e)$.
  3. Return $\perp$ if $\mathsf{k} = \perp$. Return $\mathsf{Dec}^{sym}(\mathsf{k}, d) \in \mathcal{M} \cup \{\perp\}$ otherwise.
- $\mathsf{OPEN}(i)$: Take $i \in [n]$ as input and do the following:
  1. $I \leftarrow I \cup \{i\}$.

2. Receive $\mathsf{m}_i \leftarrow \overline{\mathsf{OPEN}}(i)$.
3. $\tilde{\pi} \leftarrow \mathsf{Make}(\mathsf{st}_i, \mathsf{m}_i)$ and oracles $\ddot{E}^{+}(\mathsf{k}'_i, \cdot), \ddot{E}^{-}(\mathsf{k}'_i, \cdot)$ follow this relation $\tilde{\pi}$.
4. Abort this game if $d_i \neq \mathsf{O.Enc}^{E_{\mathsf{k}'_i}}(\mathsf{k}''_i, \mathsf{m}_i)$.
5. Return (m_i, r_i).

When A outputs out, S halts and outputs $R(\mathcal{M}_{\mathrm{D}}, m_1, \ldots, m_n, I, out)$.

S completely simulates Game_4 by using only the given oracle $\overline{\mathsf{OPEN}}$. Thus, we have $\Pr[\mathsf{Exp}^{\text{ideal-so-cca}}_{\mathsf{PKE}_1^{hy},\mathsf{S}} \to 1] = \Pr[W_4]$.

Therefore, we obtain the following advantage

$$\mathsf{Adv}^{\text{sim-so-cca}}_{\mathsf{PKE}_1^{hy},\mathsf{A},\mathsf{S},R}(\lambda) \leq n \cdot \mathsf{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathsf{D}_1}(\lambda) + 2\sqrt{nq_e \cdot \mathsf{Adv}^{\text{ind-cca}}_{\mathsf{KEM},\mathsf{D}_2}(\lambda)} + n \cdot \mathsf{Adv}^{\text{int-ctxt}}_{\mathsf{DEM},\mathsf{F}}(\lambda) + n \cdot \epsilon_{sim} + 4q_e\sqrt{\frac{n}{|\mathcal{K}'|}} + \frac{n^2}{|\mathcal{K}'|}.$$

From the discussion above, the proof is completed. □

3.4 PKE from FO-based KEM schemes

We describe a PKE scheme PKE_2^{hy} constructed from an FO-based KEM $\mathsf{FO}^{\not\perp}$ and any $\mathsf{sUF\text{-}OT\text{-}CMA}$ secure MAC, and prove that this scheme meets $\mathsf{SIM\text{-}SO\text{-}CCA}$ security in the QROM. As FO-based KEM schemes, we can apply not only $\mathsf{FO}^{\not\perp}$ but also other FO-based schemes $\mathsf{FO}^{\not\perp}_m$, $\mathsf{QFO}^{\not\perp}$, and $\mathsf{QFO}^{\not\perp}_m$, which are classified in [64]. In this paper, we select $\mathsf{FO}^{\not\perp}$ to construct PKE_2^{hy} because $\mathsf{FO}^{\not\perp}$ is used to construct many KEMs submitted to the PQC standardization project. Besides, it does not have to append additional hash [127, 64] to ciphertexts while $\mathsf{QFO}^{\not\perp}$ and $\mathsf{QFO}^{\not\perp}_m$ need additional hash. Notice that in the same way as the security proof of PKE_2^{hy} (Theorem 3.2), it is possible to prove the security of PKE_2^{hy} using $\mathsf{FO}^{\not\perp}_m$, $\mathsf{QFO}^{\not\perp}$, or $\mathsf{QFO}^{\not\perp}_m$, instead of $\mathsf{FO}^{\not\perp}$.

Concretely, we can apply CRYSTALS-Kyber, SABER, SIKE, and LEDAkem to the KEM scheme $\mathsf{FO}^{\not\perp}$, and apply FrodoKEM, NewHope, ThreeBears, and more other schemes [107] to $\mathsf{FO}^{\not\perp}_m$, $\mathsf{QFO}^{\not\perp}$, or $\mathsf{QFO}^{\not\perp}_m$. As concrete MAC schemes, we can use deterministic MACs standardized by NIST.

To construct PKE_2^{hy} with a message space $\mathcal{M}$, we use the following primitives: Let $\mathsf{PKE}^{asy} = (\mathsf{KGen}^{asy}, \mathsf{Enc}^{asy}, \mathsf{Dec}^{asy})$ with δ-correctness be a PKE scheme with a message space $\mathcal{M}^{asy}$, a randomness space $\mathcal{R}^{asy}$, and a ciphertext space $\mathcal{C}^{asy}$. Let $\mathsf{MAC} = (\mathsf{Tag}, \mathsf{Vrfy})$ be a MAC scheme with a key space $\mathcal{K}^{mac}$. Let $\mathsf{H} : \mathcal{M}^{asy} \times \mathcal{C}^{asy} \to \mathcal{K}^{sym} \times \mathcal{K}^{mac}$, $\mathsf{G} : \mathcal{M}^{asy} \to \mathcal{R}^{asy}$ be random oracles, where $\mathcal{K}^{sym} = \mathcal{M}$ is a key space. $\mathsf{PKE}_2^{hy} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is constructed as follows:

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$: Generate $(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}) \leftarrow \mathsf{KGen}^{asy}(1^\lambda)$ and $s \xleftarrow{U} \mathcal{M}^{asy}$. Then, output $\mathsf{pk} := \mathsf{pk}^{asy}$ and $\mathsf{sk} := (\mathsf{sk}^{asy}, s)$.

- $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$: Encrypt $\mathsf{m} \in \mathcal{M}$ as follows:
  1. $r \xleftarrow{U} \mathcal{M}^{asy}$.
  2. $e \leftarrow \mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r; \mathsf{G}(r))$.
  3. $(\mathsf{k}^{sym}, \mathsf{k}^{mac}) \leftarrow \mathsf{H}(r, e)$.
  4. $d \leftarrow \mathsf{k}^{sym} \oplus \mathsf{m}$, $\mathsf{t} \leftarrow \mathsf{Tag}(\mathsf{k}^{mac}, d)$.
  5. Output $\mathsf{ct} := (e, d, \mathsf{t})$.
- $\mathsf{m}/\perp \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: Decrypt $\mathsf{ct} = (e, d, \mathsf{t})$ as follows:
  1. $r' \leftarrow \mathsf{Dec}^{asy}(\mathsf{sk}^{asy}, e)$.
  2. $(\mathsf{k}^{sym}, \mathsf{k}^{mac}) \leftarrow \mathsf{H}(s, e)$ if $e \neq \mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r'; \mathsf{G}(r'))$.
  3. $(\mathsf{k}^{sym}, \mathsf{k}^{mac}) \leftarrow \mathsf{H}(r', e)$ otherwise.
  4. Output $\mathsf{m} := d \oplus \mathsf{k}^{sym}$ if $\mathsf{Vrfy}(\mathsf{k}^{mac}, d, \mathsf{t}) = 1$, and output $\perp$ otherwise.

As the security of PKE_2^{hy}, Theorem 3.2 holds.

Theorem 3.2. *If a PKE scheme* PKE^{asy} *with* δ*-correctness meets* IND-CPA *security, and a MAC scheme* MAC *meets* sUF-OT-CMA *security, then* PKE_2^{hy} *satisfies* SIM-SO-CCA *security in the quantum random oracle model.*

$\underline{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)}$

1 : $(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}) \leftarrow \mathsf{KGen}^{asy}(1^\lambda)$.
2 : $s \xleftarrow{U} \mathcal{M}^{asy}$.
3 : **return** $\mathsf{pk} := \mathsf{pk}^{asy}$ *and* $\mathsf{sk} := (\mathsf{sk}^{asy}, s)$

$\underline{(e, \mathsf{k}) \leftarrow \mathsf{Encap}(\mathsf{pk})}$

1 : $r \xleftarrow{U} \mathcal{M}^{asy}$
2 : $e \leftarrow \mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r; \mathsf{G}(r))$
3 : $\mathsf{k} \leftarrow \mathsf{H}(r, e)$
4 : **return** (e, k)

$\underline{\mathsf{k} \leftarrow \mathsf{Decap}(\mathsf{sk}, e)}$

1 : $r' \leftarrow \mathsf{Dec}^{asy}(\mathsf{sk}^{asy}, e)$
2 : **if** $e \neq \mathsf{Enc}^{asy}(\mathsf{pk}, r'; \mathsf{G}(r'))$:
  return $\mathsf{k} := \mathsf{H}(s, e)$
3 : **return** $\mathsf{k} := \mathsf{H}(r', e)$

Figure 3.1: KEM scheme $\mathsf{FO}^{\not\perp}$ in PKE_2^{hy}

Proof. Let A be a QPT adversary against PKE_2^{hy}. Let q_d be the number of accessing $\mathsf{DEC}(\cdot)$, q_h be the number of accessing $\mathsf{H}(\cdot)$, q_g be the number of accessing $\mathsf{G}(\cdot)$. For a subset $J \subseteq [n]$, let $K_J^{sym} := \{\mathsf{k}_j^{sym} \mid j \in J\}$. Notice that we can view $\mathsf{FO}^{\not\perp}$ in Figure 3.1 as the underlying KEM scheme in PKE_2^{hy}.

For $i \in \{0, 1, \ldots, 9\}$, we consider a security game Game_i, and let W_i be the event that A outputs *out* such that $R(\mathcal{M}_{\mathrm{D}}, \mathsf{m}_1, \ldots, \mathsf{m}_n, I, out) = 1$ in Game_i.

Game_0: This game is the same as Real-SIM-SO-CCA security game. Thus, we have $\Pr[\mathsf{Exp}^{\text{real-so-cca}}_{\mathsf{PKE}_2^{hy},\mathsf{A}} \to 1] = \Pr[W_0]$. ■

Game_1: This game is the same as Game_0 except that DEC oracle computes $(\mathsf{k}^{sym}, \mathsf{k}^{mac}) \leftarrow \mathsf{H}'_q(e)$ instead of $(\mathsf{k}^{sym}, \mathsf{k}^{mac}) \leftarrow \mathsf{H}(s, e)$ if $e \neq \mathsf{Enc}^{asy}(\mathsf{pk}, r'; \mathsf{G}(r'))$, where $\mathsf{H}'_q : \mathcal{C}^{asy} \to \mathcal{K}^{sym} \times \mathcal{K}^{mac}$ is a random oracle. By using Lemma 4 in [76], we have $|\Pr[W_0] - \Pr[W_1]| \leq 2q_h/\sqrt{|\mathcal{M}^{asy}|}$. ■

We define $\mathsf{G}' : \mathcal{M}^{asy} \to \mathcal{R}^{asy}$ as a random oracle which, on input $r \in \mathcal{M}^{asy}$, returns a value sampled from the uniform distribution over a set of "good" random coins $\mathcal{R}^{asy}_{good}(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}, r) = \{\hat{r} \in \mathcal{R}^{asy} \mid \mathsf{Dec}^{asy}(\mathsf{sk}^{asy}, \mathsf{Enc}^{asy}(\mathsf{pk}, r; \hat{r})) = r\}$. Let $\delta(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}, r) = |\mathcal{R}^{asy} \backslash \mathcal{R}^{asy}_{good}(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}, r)|/|\mathcal{R}^{asy}|$ denote the fraction of bad random coins, and let $\delta(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}) = \max_{r \in \mathcal{M}^{asy}} \delta(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}, r)$. And then, we have $\delta = \mathbf{E}[\delta(\mathsf{pk}^{asy}, \mathsf{sk}^{asy})]$ as the expectation of $\delta(\mathsf{pk}^{asy}, \mathsf{sk}^{asy})$, which is taken over $(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}) \leftarrow \mathsf{KGen}^{asy}(1^\lambda)$.

Game_2: This game is the same as Game_1 except that we replace the random oracle $\mathsf{G}(\cdot)$ by $\mathsf{G}' : \mathcal{M}^{asy} \to \mathcal{R}^{asy}$.

In the same way as the proof of Theorem 1 in [78], we can apply Lemma 2.1. Namely, G and G' can be viewed as F and N oracles in the generic search problem, respectively. Thus, we get $|\Pr[W_1] - \Pr[W_2]| \leq 2q_g\sqrt{\delta}$. ■

Game_3: This game is the same as Game_2 except that the random oracle $\mathsf{H}(r, e)$ returns $\mathsf{H}_q(\mathsf{Enc}^{asy}(\mathsf{pk}, r; \mathsf{G}'(r)))$ if $e = \mathsf{Enc}^{asy}(\mathsf{pk}, r; \mathsf{G}'(r))$, and returns $\mathsf{H}'(r, e)$ otherwise. $\mathsf{H}_q : \mathcal{C}^{asy} \to \mathcal{K}^{sym} \times \mathcal{K}^{mac}$ and $\mathsf{H}' : \mathcal{M}^{asy} \times \mathcal{C}^{asy} \to \mathcal{K}^{sym} \times \mathcal{K}^{mac}$ are random oracles.

Since $\mathsf{G}'(\cdot)$ oracle returns "good" random coins, $\mathsf{Enc}^{asy}(\mathsf{pk}, \cdot; \mathsf{G}'(\cdot))$ is injective. Hence, we can view $\mathsf{H}_q(\mathsf{Enc}^{asy}(\mathsf{pk}, \cdot; \mathsf{G}(\cdot)))$ as a perfect random oracle, and $\Pr[W_3] = \Pr[W_2]$ holds. ■

Game_4: This game is the same as Game_3 except that DEC oracle is modified as follows: Take $\mathsf{ct} = (e, d, \mathsf{t})$ as input and compute $(\mathsf{k}^{sym}, \mathsf{k}^{mac}) \leftarrow \mathsf{H}_q(e)$. Then, return $\mathsf{m} \leftarrow \mathsf{k}^{sym} \oplus d$ if $\mathsf{Vrfy}(\mathsf{k}^{mac}, d, \mathsf{t}) = 1$, and return $\perp$ otherwise.

In the case where $e = \mathsf{Enc}^{asy}(\mathsf{pk}, r; \mathsf{G}'(r))$ holds, both Decap algorithms of Figure 3.1 in Hybrid_3 and Hybrid_4 return the same value. In the case where $e \neq \mathsf{Enc}^{asy}(\mathsf{pk}, r; \mathsf{G}(r))$ holds, A cannot distinguish Game_3 and Game_4 since both H oracles in the two games return uniformly random values. Thus, we have $\Pr[W_4] = \Pr[W_3]$. ■

Game_5: This game is the same as Game_4 except that we replace the random oracle $\mathsf{G}'(\cdot)$ by $\mathsf{G}(\cdot)$. In the same way as the game-hop of Game_2, we have $|\Pr[W_4] - \Pr[W_5]| \leq 2q_g\sqrt{\delta}$. ■

We define $\ddot{\mathsf{G}}$ (resp. $\ddot{\mathsf{H}}$) as a random oracle such that for $i \in [n]$, the value $\ddot{\mathsf{G}}(r_i)$ (resp. $\ddot{\mathsf{H}}(r_i, e)$) is sampled from $\mathcal{R}^{asy}$ (resp. $\mathcal{K}^{sym} \times \mathcal{K}^{mac}$) uniformly at random and patches the uniformly random value, and for $r \notin \{r_i\}_{i \in [n] \backslash I}$, $\ddot{\mathsf{G}}(r) = \mathsf{G}(r)$ (resp. $\ddot{\mathsf{H}}(r, e) = \mathsf{H}(r, e)$) holds.

Game_6: This game is the same as Game_5 except that at the beginning of the security game, the challenger computes (e_i, k_i) for $i \in [n]$, and oracles H and G are replaced by $\ddot{\mathsf{H}}\backslash S$ and $\ddot{\mathsf{G}}\backslash S$ for $S = \{r_i\}_{i\in[n]\backslash I}$, respectively, before A queries to OPEN oracle.

In the similar way as the proof of Theorem 1 in [78], the following lemma holds.

Lemma 3.1. *For any QPT algorithm* A *against* PKE_2^{hy} *that makes at most* q_g *queries to* G *and at most* q_h *queries to* H*, there exists a PPT algorithm* D *against* PKE^{asy} *such that*

$$|\Pr[W_5] - \Pr[W_6]| \leq 2\sqrt{n(q_g+q_h)\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE}^{asy},\mathsf{D}}(\lambda)} + 4(q_g+q_h)\sqrt{\frac{n}{|\mathcal{M}^{asy}|}}.$$

Proof. We use the same notations defined in the proof of Theorem 3.2. For $i \in \{0, 1, \ldots, 4\}$, we consider games Hybrid_i, and let H_i be the event that A outputs *out* such that $R(\mathcal{M}_{\mathrm{D}}, \mathsf{m}_1, \ldots, \mathsf{m}_n, I, out) = 1$ in Hybrid_i, Find_i be the event that a semi-classical oracle O_S^{SC} returns $\sum_{x\in S, y\in\mathcal{Y}} \psi'_{x,y} |x, y\rangle |1\rangle$ for a quantum query $\sum_{x\in\mathcal{M}^{asy}, y\in\mathcal{Y}} \psi_{x,y} |x, y\rangle$ to the random oracle G (resp. H), where $S = \{r_i\}_{i\in[n]\backslash I}$ and $\mathcal{Y} = \mathcal{R}^{asy}$ (resp. $\mathcal{Y} = \mathcal{C}^{asy} \times \mathcal{K}^{sym} \times \mathcal{K}^{mac}$). In addition, in the same way as the proof in Theorem 3.2, random oracles $\ddot{\mathsf{G}}$ and $\ddot{\mathsf{H}}$ are defined.

Hybrid_0: This game is the same as Game_5 in the proof of Theorem 3.2. Then, we have $\Pr[H_0] = \Pr[W_5]$. ■

Hybrid_1: This game is the same as Hybrid_0 except that we replace G and H by $\ddot{\mathsf{G}}\backslash S$ and $\ddot{\mathsf{H}}\backslash S$, respectively, where $S = \{r_i\}_{i\in[n]\backslash I}$.

From Proposition 2.1, we have $|\Pr[H_0] - \Pr[H_1]| \leq 2\sqrt{(q_g+q_h)\Pr[\mathsf{Find}_1]}$. Notice that we also have $\Pr[H_1] = \Pr[W_6]$. ■

Hybrid_2: This game is the same as Hybrid_1 except that for all $i \in [n]$, we replace $\hat{r}_i \xleftarrow{U} \mathcal{R}^{asy}$ and $(\mathsf{k}_i^{sym}, \mathsf{k}_i^{mac}) \xleftarrow{U} \mathcal{K}^{sym} \times \mathcal{K}^{mac}$ instead of $\hat{r}_i \leftarrow \mathsf{G}(r_i)$ and $(\mathsf{k}_i^{sym}, \mathsf{k}_i^{mac}) \leftarrow \mathsf{H}(r_i, e_i)$, respectively. We have $\Pr[\mathsf{Find}_2] = \Pr[\mathsf{Find}_1]$ because we do not focus on the output of A. ■

Hybrid_3: This game is the same as Hybrid_2 except that we replace $\ddot{\mathsf{G}}$ and $\ddot{\mathsf{H}}$ by G and H, respectively. Because there is no difference between the view of A in the two games by this change, $\Pr[\mathsf{Find}_3] = \Pr[\mathsf{Find}_2]$ holds. ■

Hybrid_4: This game is the same as Hybrid_3 except that we replace r_i by r'_i for all $i \in [n]$. Notice that we do not replace the set $S = \{r_i\}_{i\in[n]\backslash I}$ by $\{r'_i\}_{i\in[n]\backslash I}$.

We show $|\Pr[\mathsf{Find}_3] - \Pr[\mathsf{Find}_4]| \leq n \cdot \mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathsf{D}}(\lambda)$ by constructing the following PPT algorithm D breaking $\mathsf{IND\text{-}CPA}$ security of PKE^{asy}: Given a public key pk^{asy}, D chooses $i^* \in [n]$, $r_{i^*}, r'_{i^*} \in \mathcal{M}^{asy}$ uniformly at random. It submits (r_i, r'_i) to the challenger in $\mathsf{IND\text{-}CPA}$ game and receives e_{i^*}. And then, it computes $e_i \leftarrow \mathsf{Enc}^{asy}(\mathsf{pk}, r_i; \mathsf{G}(r_i))$ and $\mathsf{k}_i \leftarrow \mathsf{H}_q(e_i)$ for $i \in [n]\backslash\{i^*\}$. In order to

simulate a random oracle G (resp. H_q), D chooses a $2q_g$-wise independent hash function (resp. a $2q_h$-wise independent hash function) uniformly at random. It sets $I \leftarrow \emptyset$ and sends $\mathsf{pk} := \mathsf{pk}^{asy}$ to A.

When A submits $\mathcal{M}_{\mathrm{D}}$, D chooses $\mathsf{m}_i \xleftarrow{U} \mathcal{M}_{\mathrm{D}}$ and computes $d_i \leftarrow \mathsf{k}_i^{sym} \oplus \mathsf{m}_i$ and $\mathsf{t}_i \leftarrow \mathsf{Tag}(\mathsf{k}_i^{mac}, d_i)$ for $i \in [n]$. Then, it returns $((e_i, d_i, \mathsf{t}_i))_{i\in[n]}$.

D simulates oracles in the following way: When A issues a quantum query $\sum_{r\in\mathcal{M}^{asy},y\in\mathcal{Y}} \psi_{r,y} |r, y\rangle$ to the random oracle G (resp. H) for $\mathcal{Y} = \mathcal{R}^{asy}$ (resp. $\mathcal{Y} = \mathcal{C}^{asy} \times \mathcal{K}^{sym} \times \mathcal{K}^{mac}$), D submits $\sum_{r\in\mathcal{M}^{asy},y\in\mathcal{Y}} \psi_{r,y} |r, y\rangle |0\rangle$ to a semi-classical oracle O_S^{SC}. It halts and outputs 1 if O_S^{SC} returns the quantum superposition state $\sum_{r\in\mathcal{M}^{asy},y\in\mathcal{Y}} \psi'_{r,y} |r, y\rangle |1\rangle$. It returns a quantum state by accessing G (resp. H) otherwise.

- $\mathsf{DEC}(\mathsf{ct})$: Take $\mathsf{ct} = (e, d, \mathsf{t})$ as input and do the following.
  1. $(\mathsf{k}^{sym}, \mathsf{k}^{mac}) \leftarrow \mathsf{H}_q(e)$.
  2. Return $\mathsf{m} \leftarrow \mathsf{k}^{sym} \oplus d$ if $\mathsf{Vrfy}(\mathsf{k}^{mac}, d, \mathsf{t}) = 1$. Return $\perp$ otherwise.
- $\mathsf{OPEN}(i)$: Set $I \leftarrow I \cup \{i\}$. Abort if $i = i^*$. Return (m_i, r_i) otherwise.

When A outputs a value *out* and halts, D outputs 0. D simulates the view of A in Game_3 (resp. Game_4) if the challenger chooses r_i (resp. r'_i). Then, the success probability of D is at least $|\Pr[\mathsf{Find}_3] - \Pr[\mathsf{Find}_4]| / n$, and we have the inequality.

In addition, we get $\Pr[\mathsf{Find}_4] \leq 4(q_g + q_h)/|\mathcal{M}^{asy}|$ for each $i \in [n]$, from Proposition 2.2.

Therefore, from the union bound, we obtain

$$|\Pr[\mathsf{Find}_3] - \Pr[\mathsf{Find}_4]| + \Pr[\mathsf{Find}_4] \leq n \cdot \mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathsf{D}}(\lambda) + \frac{4n(q_g + q_h)}{|\mathcal{M}^{asy}|}.$$

■

From the discussion above, we obtain the following inequality

$$\begin{aligned}
|\Pr[W_5] - \Pr[W_6]| \leq & 2\sqrt{(q_g + q_h)\Pr[\mathsf{Find}_1]} \\
\leq & 2\sqrt{n(q_g + q_h)\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathsf{D}}(\lambda) + 4n\frac{(q_g + q_h)^2}{|\mathcal{M}^{asy}|}} \\
\leq & 2\sqrt{n(q_g + q_h)\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathsf{D}}(\lambda)} + 4(q_g + q_h)\sqrt{\frac{n}{|\mathcal{M}^{asy}|}}.
\end{aligned}$$

Therefore, we complete the proof. □

$|\Pr[W_5] - \Pr[W_6]|$ is negligible in λ if PKE^{asy} meets $\mathsf{IND\text{-}CPA}$ security. ■

Game_7: This game is the same as Game_6 except that DEC oracle returns $\perp$ if a query (e, d, t) such that $e \in \{e_i\}_{i\in[n]\setminus I}$ is submitted.

Let Bad be the event that A submits a ciphertext query (e, d, t) such that $e \in \{e_i\}_{i\in[n]\setminus I}$ and $\mathsf{Vrfy}(\mathsf{k}^{mac}, d, \mathsf{t}) = 1$. Besides, we consider the following events: Let Bad_1 be the event that Bad happens in Game_6, and let Bad_2 be the same event as Bad_1 except that keys k_i are chosen uniformly at random for all $i \in [n]$.

Then, Game_6 and Game_7 are identical until Bad_1 occurs. The modification of Bad_2 is conceptual from the game-hop of Game_6. Thus, we have $\Pr[\mathsf{Bad}_1] = \Pr[\mathsf{Bad}_2]$.

Next, we show $\Pr[\mathsf{Bad}_2] \leq n \cdot \mathsf{Adv}^{\text{suf-cma}}_{\mathsf{MAC},\mathsf{F}}(\lambda)$. We construct a PPT algorithm F breaking $\mathsf{sUF\text{-}OT\text{-}CMA}$ security as follows: It is given oracles TAG and VRFY in $\mathsf{sUF\text{-}OT\text{-}CMA}$ game. At the beginning of the security game, F generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ and chooses $i^* \in [n]$ uniformly at random. Then, it sets $I \leftarrow \emptyset$ and sends pk to A. When A submits $\mathcal{M}_{\mathrm{D}}$, it does the following for every $i \in [n]$:

1. $e_i \leftarrow \mathsf{Enc}^{asy}(\mathsf{pk}; r_i; \mathsf{G}(r_i))$, where $r_i \in \mathcal{M}^{asy}$ is sampled at random.
2. $\mathsf{m}_i \leftarrow \mathcal{M}_{\mathrm{D}}$ and $(\mathsf{k}_i^{sym}, \mathsf{k}_i^{mac}) \leftarrow \mathsf{H}(r_i, e_i)$.
3. If $i = i^*$, choose $d_{i^*} \xleftarrow{U} \mathcal{K}^{sym}$ and let $\mathsf{t}_{i^*} := \mathsf{TAG}(d_{i^*})$. Otherwise, $d_i \leftarrow \mathsf{k}_i^{sym} \oplus \mathsf{m}_i$ and $\tau_i \leftarrow \mathsf{Tag}(\mathsf{k}_i^{mac}, d_i)$.

Then, it returns $\{(e_i, d_i, \mathsf{t}_i)\}_{i\in[n]}$. F simulates oracles in the following way: From Theorem 6.1 in [137], random oracles H and G can be simulated by using a $2q_h$-wise pairwise independent hash function and a $2q_g$-wise independent hash function, respectively. The other oracles are simulated as follows:

- $\mathsf{DEC}(\mathsf{ct})$: Take $\mathsf{ct} = (e, d, \mathsf{t})$ as input. If $e = e_{i^*}$, submit (d, t) to VRFY oracle. Halt if VRFY returns 1, and return $\perp$ otherwise. If $e \neq e_{i^*}$, return $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \in \mathcal{M} \cup \{\perp\}$.
- $\mathsf{OPEN}(i)$: Set $I \leftarrow I \cup \{i\}$. Abort this game if $i = i^*$. Return (m_i, r_i) if $i \neq i^*$.

Finally, when A outputs *out*, F aborts if Bad does not occur. Then, the success condition of F is identical to the condition that Bad occurs. Hence, F wins in $\mathsf{sUF\text{-}OT\text{-}CMA}$ game if A submits a ciphertext query (e, d, t) such that $e = e_{i^*}$ and $\mathsf{VRFY}(d, \mathsf{t})$ returns 1, and the success probability of F is at least $\Pr[\mathsf{Bad}_2]/n$. From the union bound, we have $\Pr[\mathsf{Bad}_2] \leq n \cdot \mathsf{Adv}^{\text{suf-cma}}_{\mathsf{MAC},\mathsf{F}}(\lambda)$.

Therefore, we obtain

$$|\Pr[W_6] - \Pr[W_7]| \leq n \cdot \mathsf{Adv}^{\text{suf-cma}}_{\mathsf{MAC},\mathsf{F}}(\lambda).$$

$|\Pr[W_6] - \Pr[W_7]|$ is negligible in λ if MAC meets $\mathsf{sUF\text{-}OT\text{-}CMA}$ security. ■

Game_8: This game is the same as Game_7 except that the game is aborted if for $i \in [n]$, the challenger chooses $r_i \in \mathcal{M}^{asy}$ such that $\mathsf{k}_i^{sym} \in K^{sym}_{[i-1]}$, where $(\mathsf{k}_i^{sym}, \mathsf{k}_i^{mac}) = \mathsf{H}(r_i, e_i)$.

The probability of choosing $\mathsf{k}_i^{sym} \in K_{[i-1]}^{sym}$ is at most $n^2/|\mathcal{K}^{sym}|$ from the collision resistance of random oracles. ■

Game_9: This game is the same as Game_8 except that the challenge phase and OPEN oracle are modified as follows: When A submits $\mathcal{M}_{\mathrm{D}}$, the challenger chooses $d_i \in \mathcal{K}^{sym}$ and $\mathsf{k}_i^{mac} \in \mathcal{K}^{mac}$ uniformly at random, computes $\mathsf{t}_i \leftarrow \mathsf{Tag}(\mathsf{k}_i^{mac}, d_i)$, and returns (e_i, d_i, t_i) for $i \in [n]$. In addition, OPEN oracle does the following:

1. $I \leftarrow I \cup \{i\}$.
2. $\mathsf{m}_i \leftarrow \mathcal{M}_{\mathrm{D}}$.
3. Let $\mathsf{H}(r_i, e_i) := (d_i \oplus \mathsf{m}_i, \mathsf{k}_i^{mac})$.
4. Return (m_i, r_i).

Game_9 is identical to Game_8. Any QPT adversary A cannot distinguish d_i in the two games since both ciphertexts in these games are uniformly random and A cannot find $r \in \{r_i\}_{i\in[n]\setminus I}$ before querying OPEN oracle. For this reason, it is possible to define $\mathsf{H}(r_i, e_i)$ when A submits i to OPEN oracle. Hence, we have $\Pr[W_9] = \Pr[W_8]$. ■

Finally, we prove $\Pr[\mathsf{Exp}^{\text{ideal-so-cca}}_{\mathsf{PKE}_2^{hy},\mathsf{S}} \to 1] = \Pr[W_9]$ by constructing a simulator S in the following way: It is given $\overline{\mathsf{OPEN}}$ oracle. At the beginning of $\mathsf{Ideal\text{-}SIM\text{-}SO\text{-}CCA}$ security game, S generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ and e_i for $i \in [n]$. And then, it sets $I \leftarrow \emptyset$ and sends pk to A. When A submits $\mathcal{M}_{\mathrm{D}}$, it chooses $d_i \in \mathcal{M}_{\mathrm{D}}$ and $\mathsf{k}_i^{mac} \in \mathcal{K}^{mac}$ uniformly at random, and computes $\mathsf{t}_i \leftarrow \mathsf{Tag}(\mathsf{k}_i^{mac}, d_i)$ for $i \in [n]$. And then it returns $((e_i, d_i, \mathsf{t}_i))_{i\in[n]}$. It simulates random oracles by using a $2q_h$-wise independent hash function and a $2q_g$-wise independent hash function. Oracles $\mathsf{DEC}(\cdot)$ and $\mathsf{OPEN}(\cdot)$ are simulated as follows:

- $\mathsf{DEC}(\mathsf{ct})$: Take $\mathsf{ct} = (e, d, \mathsf{t})$ as input and do the following.
  1. Return $\perp$ if $e \in \{e_i\}_{i\in[n]\setminus I}$.
  2. $(\mathsf{k}^{sym}, \mathsf{k}^{mac}) \leftarrow \mathsf{H}_q(e)$.
  3. Return $\mathsf{m} \leftarrow \mathsf{k}^{sym} \oplus d$ if $\mathsf{Vrfy}(\mathsf{k}^{mac}, d, \mathsf{t}) = 1$. Return $\perp$ otherwise.
- $\mathsf{OPEN}(i)$: Take $i \in [n]$ as input and do the following:
  1. $I \leftarrow I \cup \{i\}$.
  2. Receive m_i by accessing the given $\overline{\mathsf{OPEN}}(i)$.
  3. Let $\mathsf{H}(r_i, e_i) := (d_i \oplus m_i, \mathsf{k}_i^{mac})$.
  4. Return (m_i, r_i).

When A outputs out, S halts and outputs $R(\mathcal{M}_{\mathrm{D}}, m_1, \ldots, m_n, I, out)$. Because S can simulate the view of A only with $\overline{\mathsf{OPEN}}$ oracle, we have $\Pr[\mathsf{Exp}^{\text{ideal-so-cca}}_{\mathsf{PKE}_2^{hy},\mathsf{S}} \to 1] = \Pr[W_9]$.

From the discussion above, we obtain

$$\begin{aligned}
\mathsf{Adv}^{\text{sim-so-cca}}_{\mathsf{PKE}_2^{hy},\mathsf{A},\mathsf{S},R}(\lambda) \leq & 2\sqrt{n(q_g+q_h)\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE}^{asy},\mathsf{D}}(\lambda)} + n \cdot \mathsf{Adv}^{\text{suf-cma}}_{\mathsf{MAC},\mathsf{F}}(\lambda) \\
& + 4(q_g+q_h)\sqrt{\frac{n}{|\mathcal{M}^{asy}|}} + \frac{2q_h}{\sqrt{|\mathcal{M}^{asy}|}} + 4q_g\sqrt{\delta} + \frac{n^2}{|\mathcal{K}^{sym}|} \\
\leq & 2\sqrt{n(q_g+q_h)\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE}^{asy},\mathsf{D}}(\lambda)} + n \cdot \mathsf{Adv}^{\text{suf-cma}}_{\mathsf{MAC},\mathsf{F}}(\lambda) \\
& + (4q_g+6q_h)\sqrt{\frac{n}{|\mathcal{M}^{asy}|}} + 4q_g\sqrt{\delta} + \frac{n^2}{|\mathcal{K}^{sym}|},
\end{aligned}$$

and complete the proof. □

Chapter 4

Quantum-Secure Message Authentication with Aggregation

4.1 Background of (Sequential) Aggregate MAC

Message authentication code (MAC) is a fundamental and important primitive in symmetric cryptography for message authentication by generating MAC tags on messages. In addition, there are works which researched message authentication schemes with advanced functionalities, such as aggregate MACs (AMACs) [82, 44], homomorphic message authenticators [32, 52, 47], and blind MACs [2, 105]. In this chapter, we focus on AMACs which can compress multiple MAC tags on multiple messages into a short tag (aggregate-tag). The reasons are as follows: When many MAC tags are sent to a receiver via a network, AMACs are effective since it is possible to reduce the total size of MAC tags. In addition, concrete AMACs can be constructed from standardized MACs such as CMAC and HMAC, and it is easy to implement AMAC schemes without replacing ordinary MACs implemented in devices [82, 44] while practical homomorphic message authenticators [32] and blind MACs [2, 105] are not standardized, and we have to replace implemented ordinary MACs (e.g., CMAC and HMAC) by the homomorphic or blind message authentication schemes in implementing these ones.

We describe the existing works related to AMACs. In [82], Katz and Lindell formalized the model and security of AMACs for the first time, and proposed a generic construction starting from any MAC. Sequential aggregate MAC (SAMAC) is AMAC which can verify the validity of the order of sequential messages. We can consider several applications of SAMACs such as audit-logging systems and wireless network sensor systems, and others for resource-constrained devices. In [44], Eikemeier et al. defined the model and security of SAMACs, and proposed a generic construction from any MAC and

pseudorandom permutation. In [62], Hirose and Kuwakado formalized forward security of SAMACs and proposed a generic construction from any pseudorandom function (PRF) and any pseudorandom generator. Tomita et al. [128] defined sequential aggregate authentication codes with information-theoretic (one-time) security, and they proposed constructions of SAMACs meeting this security.

Recently, quantum algorithms breaking the existing cryptosystems have been proposed and the development of quantum computers has been promoted. In fact, post-quantum cryptosystems have been studied in both areas of public key cryptography and symmetric key cryptography. In symmetric key cryptography, we focus on the security model where adversaries are allowed to submit quantum superposition states of queries (quantum queries) to oracles since we would like to establish quantum secure systems in a stronger sense. It is known that there exist quantum attacks against MAC schemes such as CBC-MAC, PMAC, and Carter-Wegman MAC in this model [24, 80]. In prior work, various MAC schemes satisfying the security in the quantum query model have been proposed. In [24], Boneh and Zhandry defined the security of MACs in this model for the first time. They also proposed several MAC schemes meeting this security: a variant of Carter-Wegman MAC, pseudorandom functions meeting the quantum security defined in [136], and a q-time MAC scheme, where q is the number of classical/quantum queries to the tagging oracle. In [124], Song and Yun showed that NMAC and HMAC met the quantum security of pseudorandom functions defined in [136], if the underlying pseudorandom functions meet the quantum security. However, no paper reports about MACs with advanced functionality of compressing multiple tags, AMACs and SAMACs. Notice that there is no work which researched the quantum-security of homomorphic message authenticators and blind MACs.

4.2 Contribution

Our purpose is to propose AMAC/SAMAC schemes meeting quantum security, namely AMAC/SAMAC schemes secure in quantum security models. To the best of our knowledge, the security of AMACs/SAMACs in this model has not been dealt with in the literature. We formalize the model and security of AMACs/SAMACs in the quantum security model. Then, we show that generic constructions of AMAC/SAMAC schemes that satisfy the security in the quantum security model. Specifically, the contribution is described as follows.

1. In Section 4.4, we formalize the security of AMACs in the quantum security model. In addition, we show that the generic construction of AMAC from any MAC, which was proposed by Katz and Lindell [82], fulfills our security, if the underlying MAC meets the post-quantum security defined in [24].

2. In Section 4.5, we formalize the quantum security of SAMACs. Our security formalization includes the existing security definition [44] in the classical security model, and hence our formalization is considered to be reasonable. In terms of quantum security, we analyze security of known SAMACs, and the results are summarized in Table 4.1. In particular, we can break the security of SAMACs of [44, 128] by using quantum algorithms proposed in [24, 80].

3. In Section 4.5.3 and 4.5.4, we propose two generic constructions of SAMACs, SAMAC_1 and SAMAC_2. SAMAC_1 is constructed from any quantum-secure pseudorandom function (QPRF), which is formalized in [136], while SAMAC_2 is constructed from any randomized pseudorandom generator (PRG) and any classical PRF. The features of those constructions are explained as follows.

   SAMAC_1 uses a deterministic PRF satisfying the quantum security formalized in [136]. In particular, we can apply the quantum secure PRFs of [136, 124] to SAMAC_1, since those are deterministic. More specifically, we can apply NMAC/HMAC to SAMAC_1 as a quantum-secure PRF, since these MACs are shown to be quantum-secure PRFs in [124].

   SAMAC_2 uses a randomized function (i.e., randomized PRG). The advantage of using randomized primitives lies in constructing quantum secure SAMAC schemes based on computationally hard problems even for quantum computers such as the learning parity with noise (LPN) problem. Since LPN-based cryptography has been studied in constructing various cryptographic systems such as public key encryption [40, 85], oblivious transfer [36], symmetric key encryption [12], MACs [38], and randomized PRGs/PRFs [135, 12], it is even interesting to consider quantum-secure SAMACs from LPN-based primitives. LPN-based primitives consist of randomized algorithms, and hence, those can be applied to SAMAC_2. In particular, we can apply randomized PRGs [135, 12] based on the LPN problem to SAMAC_2.

4.3 Existing Quantum Security of MAC

We describe the quantum security of MACs (i.e., EUF-qCMA (existential unforgeability against quantum chosen message attacks) security) following [24] in order to show that our quantum security of AMACs is an expansion of the security definition of [24]. Concerning the model of MACs, see Section 2.7.3.

Definition 4.1 ($\mathsf{EUF\text{-}qCMA}$ security [24])**.** *A MAC scheme* $\mathsf{MAC} = (\mathsf{Tag}, \mathsf{Vrfy})$ *with a key space* $\mathcal{K}$ *meets* $\mathsf{EUF\text{-}qCMA}$ *security, if for any QPT adversary*

Scheme	Primitive	Quantum Security	Attacking algorithm
[44]	MAC	insecure	Quantum algorithm against CBC-MAC (see Section 5.1 in [80])
	PRP		
[128] Scheme 1	A-code	insecure	Quantum algorithm (see the proof of Lemma 6.3 in [24])
[128] Scheme 2	A-code		
SAMAC_1	QPRF	secure	n/a
SAMAC_2	RPRG	secure	n/a
	PRF		

Table 4.1: Security of SAMAC Schemes in the Quantum Query Model: The term "Primitive" means cryptographic primitives required in the generic constructions, "Quantum Security" means security in the quantum query model, and "Attacking algorithm" means a quantum algorithm which makes the target scheme insecure in the quantum query model. PRP means a pseudorandom permutation, A-code means an authentication code with information theoretic (one-time) security, (Q)PRF means a (quantum) pseudorandom function, RPRG means a randomized pseudorandom generator, and PIH means pairwise independent hashing.

A *against* MAC, $\mathsf{Adv}^{\text{euf-qcma}}_{\mathsf{MAC},\mathsf{A}}(\lambda) := \Pr[\mathsf{A}\ \textit{wins}]$ *is negligible in* λ*, where* [A *wins*] *is an event that* A *wins in the following game:*

Setup: *A challenger chooses a secret key* $\mathsf{k} \xleftarrow{U} \mathcal{K}$.

Queries: *When* A *submits a quantum query (i.e., a superposition of messages)* $|\psi\rangle = \sum_{\mathsf{m}\in\mathcal{M},t\in\mathcal{T},z} \psi_{\mathsf{m},t,z} |\mathsf{m},t,z\rangle$ *to the tagging oracle, it chooses randomness* r *used in* Tag *algorithm, where it does not need to choose randomness* $\boldsymbol{r}$ *if* Tag *is deterministic. Then, it returns*

$$\sum_{\mathsf{m}\in\mathcal{M},t\in\mathcal{T},z} \psi_{\mathsf{m},t,z} |\mathsf{m}, t \oplus \mathsf{Tag}(\mathsf{k},\mathsf{m};r), z\rangle .$$

Let q *be the number of queries which* A *submits to the tagging oracle.*

Output: A *outputs* $(q+1)$ *message/tag pairs* $(\mathsf{m}_1,\mathsf{t}_1),\ldots,(\mathsf{m}_{q+1},\mathsf{t}_{q+1})$. A *wins if the following holds:*

- $1 \leftarrow \mathsf{Vrfy}(\mathsf{k},\mathsf{m}_i,\mathsf{t}_i)$ *for all* $i \in [q+1]$.
- $\mathsf{m}_i \neq \mathsf{m}_j$ *for any distinct* $i,j \in [q+1]$.

4.4 Quantum-Secure AMAC

4.4.1 Quantum Security of AMAC

In this section, we formalize the quantum security of AMACs by taking into account the quantum security of MACs in [24] and (classical) security of AMACs in [82].

First, we describe the model of AMACs which is an existing one of [82]. An AMAC scheme consists of four polynomial-time algorithms (KGen, Tag, Agg, AVrfy): Let λ be a security parameter, and let $n = \mathsf{poly}(\lambda)$ be the number of tagging users. $\mathcal{ID} = \{\mathsf{id}_i\}_{i\in[n]} \in \left(\{0,1\}^{O(\lambda)}\right)^n$ is an ID space, $\mathcal{K} = \mathcal{K}(\lambda)$ is a key space, $\mathcal{M} = \mathcal{M}(\lambda)$ is a message space, and $\mathcal{T} = \mathcal{T}(\lambda)$ is a tag space.

Key Generation. KGen is a randomized algorithm which, on input a security parameter 1^λ and an ID $\mathsf{id} \in \mathcal{ID}$, outputs a secret key $\mathsf{k}_{\mathsf{id}} \in \mathcal{K}$. We write $\mathsf{k}_{\mathsf{id}} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id})$.

Tagging. Tag is an algorithm which, on input a secret key $\mathsf{k}_{\mathsf{id}} \in \mathcal{K}$ and a message $\mathsf{m} \in \mathcal{M}$, outputs a tag $\mathsf{t} \in \mathcal{T}$. We write $\mathsf{t} \leftarrow \mathsf{Tag}(\mathsf{k}_{\mathsf{id}}, \mathsf{m})$.

Aggregation. Agg is a deterministic algorithm which, on input a set of arbitrary ℓ pairs of IDs and tags $T = \{(\mathsf{id}_{\sigma(i)}, \mathsf{t}_i)\}_{i\in[\ell]}$ ($\ell \le n$), outputs an aggregate tag τ. We write $\tau \leftarrow \mathsf{Agg}(T)$.

Verification. AVrfy is a deterministic algorithm which, on input a set of secret keys $K = \{\mathsf{k}_{\mathsf{id}_i}\}_{i\in[n]}$, a set of arbitrary ℓ pairs of IDs and messages $M = \{(\mathsf{id}_{\sigma(i)}, \mathsf{m}_i)\}_{i\in[\ell]}$, and an aggregate tag τ, outputs 1 (accept) or 0 (reject). We write $1/0 \leftarrow \mathsf{AVrfy}(K, M, \tau)$.

We require that AMAC schemes $(\mathsf{KGen}, \mathsf{Tag}, \mathsf{Agg}, \mathsf{AVrfy})$ meet correctness as follows: For any set $K = \{\mathsf{k}_{\mathsf{id}_i}\}_{i\in[n]}$ of secret keys ($\forall \mathsf{id}_i \in \mathcal{ID}$, $\mathsf{k}_{\mathsf{id}_i} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id}_i)$), and any set M of ID/message pairs, we have $1 \leftarrow \mathsf{AVrfy}(K, M, \tau)$, where $\tau \leftarrow \mathsf{Agg}(\{\mathsf{id}_{\sigma(i)}, \mathsf{t}_i\}_{i\in[\ell]})$, where $\mathsf{t}_i \leftarrow \mathsf{Tag}(\mathsf{k}_{\mathsf{id}_{\sigma(i)}}, \mathsf{m}_i)$ for $i \in [\ell]$ ($1 \le \ell \le n$).

Next, we define the quantum security of AMACs: *aggregate unforgeability against quantum chosen message attacks*, which we call $\mathsf{aggUF\text{-}qCMA}$ security, as follows.

Definition 4.2 ($\mathsf{aggUF\text{-}qCMA}$ security)**.** *An AMAC scheme* $\mathsf{AMAC} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Agg}, \mathsf{AVrfy})$ *meets* $\mathsf{aggUF\text{-}qCMA}$ *security, if for any QPT adversary* A *against* AMAC, $\mathsf{Adv}^{\mathrm{agguf\text{-}qcma}}_{\mathsf{AMAC},\mathsf{A}}(\lambda) := \Pr[\mathsf{A}\ \textit{wins}]$ *is negligible in* λ*, where* $[\mathsf{A}\ \textit{wins}]$ *is the event that* A *wins in the following game:*

Setup: *Generate secret keys* $\mathsf{k}_{\mathsf{id}_i} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id}_i)$ *for all* $\mathsf{id}_i \in \mathcal{ID}$*. Set a list* $L_{\mathsf{Cor}} \leftarrow \emptyset$*.*

Queries: A *is allowed to submit queries to the following oracles* $\mathsf{O}_{\mathsf{Cor}}, \mathsf{O}_{\mathsf{Tag}}$*:*

- $\mathsf{O}_{\mathsf{Cor}}$*: Given a query* $\mathsf{id} \in \mathcal{ID}$*, a corrupt oracle* $\mathsf{O}_{\mathsf{Cor}}$ *returns the corresponding key* k_{id} *and sets* $L_{\mathsf{Cor}} \leftarrow L_{\mathsf{Cor}} \cup \{\mathsf{id}\}$.
- $\mathsf{O}_{\mathsf{Tag}}$*: Given an ID* $\mathsf{id} \in \mathcal{ID}$ *and a quantum superposition of messages* $|\psi\rangle = \sum_{\mathsf{m}\in\mathcal{M},t\in\mathcal{T},z} \psi_{\mathsf{m},t,z} |\mathsf{m}, t, z\rangle$*, a tagging oracle* $\mathsf{O}_{\mathsf{Tag}}$ *chooses randomness* r *used in* Tag *algorithm, where it does not need to choose randomness* r *if* Tag *is deterministic. Then, it returns* $\sum_{\mathsf{m}\in\mathcal{M},t\in\mathcal{T},z} \psi_{\mathsf{m},t,z} |\mathsf{m}, t \oplus \mathsf{Tag}(\mathsf{k}_{\mathsf{id}}, \mathsf{m}; r), z\rangle$*. Let* q *be the number of issued queries to* $\mathsf{O}_{\mathsf{Tag}}$*, such that* $\mathsf{id} \notin L_{\mathsf{Cor}}$.

Output: A *outputs* q *ID/message/tag triplets* $(\mathsf{id}^{(1)}, \mathsf{m}^{(1)}, \mathsf{t}^{(1)}), \ldots, (\mathsf{id}^{(q)}, \mathsf{m}^{(q)}, \mathsf{t}^{(q)})$ *and* (M, τ)*, where* $\mathsf{id}^{(i)} \in \mathcal{ID}$ *(*$i \in [q]$*), and* $M = \{(\mathsf{id}_{\sigma(i)}, \mathsf{m}_i)\}_{i\in[\ell]}$ $(1 \leq \ell \leq n)$ *is a set of arbitrary* ℓ *pairs of IDs and messages and* τ *is an aggregate tag. Then,* A *wins if the following holds:*

- $1 \leftarrow \mathsf{AVrfy}(\mathsf{k}_{\mathsf{id}^{(i)}}, (\mathsf{id}^{(i)}, \mathsf{m}^{(i)}), \mathsf{t}^{(i)})$ *for all* $i \in [q]$*, and* $1 \leftarrow \mathsf{AVrfy}(K, M, \tau)$.
- *There exists some* $(\mathsf{id}, \mathsf{m}) \in M$ *such that* $\mathsf{id} \notin L_{\mathsf{Cor}}$ *and* $(\mathsf{id}, \mathsf{m}) \notin \{(\mathsf{id}^{(1)}, \mathsf{m}^{(1)}), \ldots, (\mathsf{id}^{(q)}, \mathsf{m}^{(q)})\}$.

Definition 4.2 is regarded as an extension from both security notions of the quantum security of MACs in [24] and (classical) security of AMACs in [82] from the following reasons:

- Consider a special case $n = \ell = 1$ in Definition 4.2. Suppose that, in the $\mathsf{aggUF\text{-}qCMA}$ security game, a QPT adversary A finally outputs q ID/message/tag triplets $(\mathsf{id}^{(1)}, \mathsf{m}^{(1)}, \mathsf{t}^{(1)}), \ldots, (\mathsf{id}^{(1)}, \mathsf{m}^{(q)}, \mathsf{t}^{(q)})$ for the same ID, and (M, τ), where $M = \{m\}$ is a set consisting of a single element and τ is a single tag. Then, A wins, if $1 \leftarrow \mathsf{AVrfy}(\mathsf{k}_{\mathsf{id}^{(1)}}, (\mathsf{id}^{(1)}, \mathsf{m}^{(i)}), \mathsf{t}^{(i)})$ for all $i \in [q]$ and $1 \leftarrow \mathsf{AVrfy}(\mathsf{k}_{\mathsf{id}^{(1)}}, m, t)$, and $m \notin \{\mathsf{m}^{(1)}, \ldots, \mathsf{m}^{(q)}\}$. This is the same as Definition 4.1, and hence Definition 4.2 is regarded as an extension from quantum security of MACs in [24].
- Consider a special case where PPT algorithm A obtains valid q triplets $(\mathsf{id}^{(1)}, \mathsf{m}^{(1)}, \mathsf{t}^{(1)}), \ldots, (\mathsf{id}^{(q)}, \mathsf{m}^{(q)}, \mathsf{t}^{(q)})$ by having access to the oracle $\mathsf{O}_{\mathsf{Tag}}$ with classical queries $(\mathsf{id}^{(1)}, \mathsf{m}^{(1)}), \ldots, (\mathsf{id}^{(q)}, \mathsf{m}^{(q)})$. Suppose that, in the $\mathsf{aggUF\text{-}qCMA}$ security game, A outputs q ID/message/tag triplets $(\mathsf{id}^{(1)}, \mathsf{m}^{(1)}, \mathsf{t}^{(1)}), \ldots, (\mathsf{id}^{(1)}, \mathsf{m}^{(q)}, \mathsf{t}^{(q)})$ which he obtained, and (M, τ), where $M = \{(\mathsf{id}_{\sigma(i)}, \mathsf{m}_i)\}_{i\in[\ell]}$ $(1 \leq \ell \leq n)$ is a set of arbitrary ℓ pairs of IDs and messages and τ is an aggregate tag. Then, A wins, if $1 \leftarrow \mathsf{AVrfy}(K, M, \tau)$ and there is some $(\mathsf{id}, \mathsf{m}) \in M$ such that $\mathsf{id} \notin L_{\mathsf{Cor}}$ and $(\mathsf{id}, \mathsf{m}) \notin \{(\mathsf{id}^{(1)}, \mathsf{m}^{(1)}), \ldots, (\mathsf{id}^{(q)}, \mathsf{m}^{(q)})\}$. This is the same as the security definition of AMACs in [82], and ours is an extension of it.

4.4.2 Katz-Lindell Construction

We show that the Katz-Lindell construction [82] of AMACs meets $\mathsf{aggUF\text{-}qCMA}$ security. Let $\mathsf{MAC} = (\mathsf{Tag}_{\mathsf{MAC}}, \mathsf{Vrfy}_{\mathsf{MAC}})$ be a deterministic MAC scheme with

a key space $\mathcal{K}$. The Katz-Lindell construction $\mathsf{AMAC}_{\mathrm{KL}} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Agg}, \mathsf{AVrfy})$ is described as follows:

- $\mathsf{k}_{\mathsf{id}} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id})$: Output $\mathsf{k}_{\mathsf{id}} \xleftarrow{U} \mathcal{K}$ for an ID $\mathsf{id} \in \mathcal{ID}$.
- $\mathsf{t} \leftarrow \mathsf{Tag}(\mathsf{k}_{\mathsf{id}}, \mathsf{m})$: Output $\mathsf{t} \leftarrow \mathsf{Tag}_{\mathsf{MAC}}(\mathsf{k}_{\mathsf{id}}, \mathsf{m}) \in \mathcal{T}$ on a message $\mathsf{m} \in \mathcal{M}$.
- $\tau \leftarrow \mathsf{Agg}(\{(\mathsf{id}_{\sigma(1)}, \mathsf{t}_1), \ldots, (\mathsf{id}_{\sigma(\ell)}, \mathsf{t}_\ell)\})$: Output $\tau := \mathsf{t}_1 \oplus \cdots \oplus \mathsf{t}_\ell \in \mathcal{T}$.
- $1/0 \leftarrow \mathsf{AVrfy}(K, M, \tau)$: Verify an ID/message set $M = \{(\mathsf{id}_{\sigma(i)}, \mathsf{m}_i)\}_{i\in[\ell]}$ and an aggregate tag τ in the following way:
  1. $\tilde{\tau} \leftarrow \mathsf{Agg}(\{(\mathsf{id}_{\sigma(1)}, \tilde{\mathsf{t}}_1), \ldots, (\mathsf{id}_{\sigma(\ell)}, \tilde{\mathsf{t}}_\ell)\})$, where $\tilde{\mathsf{t}}_i \leftarrow \mathsf{Tag}(\mathsf{k}_{\mathsf{id}_{\sigma(i)}}, \mathsf{m}_i)$.
  2. Output 1 if $\tau = \tilde{\tau}$, and output 0 otherwise.

We show the following theorem which states quantum security of the construction $\mathsf{AMAC}_{\mathrm{KL}}$.

Theorem 4.1. *If a deterministic* MAC *meets* EUF-qCMA *security,* $\mathsf{AMAC}_{\mathrm{KL}}$ *satisfies* aggUF-qCMA *security.*

Proof. Let A be a QPT adversary against $\mathsf{AMAC}_{\mathrm{KL}}$. We prove the theorem by constructing a PPT algorithm F breaking the EUF-qCMA security of MAC, in the following way: Given a tagging oracle in EUF-qCMA game, it chooses $\mathsf{id}^* \in \mathcal{ID}$ uniformly at random and generates k_{id} for all $\mathsf{id} \in \mathcal{ID}$ and a list $L_{\mathsf{Cor}} \leftarrow \emptyset$. When A submits queries to $\mathsf{O}_{\mathsf{Cor}}$ and $\mathsf{O}_{\mathsf{Tag}}$, it simulates these oracles as follows:

- $\mathsf{O}_{\mathsf{Cor}}$: Take id as input. Abort this game if $\mathsf{id} = \mathsf{id}^*$. Return the corresponding key k_{id} and set $L_{\mathsf{Cor}} \leftarrow L_{\mathsf{Cor}} \cup \{\mathsf{id}\}$ if $\mathsf{id} \neq \mathsf{id}^*$.
- $\mathsf{O}_{\mathsf{Tag}}$: Take $(\mathsf{id}, \sum_{\mathsf{m}\in\mathcal{M}, t\in\mathcal{T}, z} \psi_{\mathsf{m},t,z} |\mathsf{m}, t, z\rangle)$ as input. If $\mathsf{id} = \mathsf{id}^*$, submit the given quantum query to the tagging oracle and return the received quantum superposition. If $\mathsf{id} \neq \mathsf{id}^*$, return $\sum_{\mathsf{m},t,z} \psi_{\mathsf{m},t,z} |\mathsf{m}, t \oplus \mathsf{Tag}(\mathsf{k}_{\mathsf{id}}, \mathsf{m}), z\rangle$.

When A outputs $(\mathsf{id}^{(1)}, \mathsf{m}^{(1)}, \mathsf{t}^{(1)}), \ldots, (\mathsf{id}^{(q)}, \mathsf{m}^{(q)}, \mathsf{t}^{(q)})$ and (M, τ), where $M = \{(\mathsf{id}_{\sigma(i)}, \mathsf{m}_i)\}_{i\in[\ell]}$, then F checks the following:

- For all $\mathsf{id}^{(i)} \neq \mathsf{id}^*$ $(i \in [q])$, we have $1 \leftarrow \mathsf{AVrfy}(\mathsf{k}_{\mathsf{id}^{(i)}}, (\mathsf{id}^{(i)}, \mathsf{m}^{(i)}), \mathsf{t}^{(i)})$, and
- there exists some ID/message pair $(\mathsf{id}^*, \mathsf{m}^*) \in M$ such that $(\mathsf{id}^*, \mathsf{m}^*) \notin \{(\mathsf{id}^{(1)}, \mathsf{m}^{(1)}), \ldots, (\mathsf{id}^{(q)}, \mathsf{m}^{(q)})\}$.

If the output of A meets these conditions, F sets $\mathsf{t}^* \leftarrow \tau$ and computes $\mathsf{t}^* \leftarrow \mathsf{t}^* \oplus \mathsf{Tag}(\mathsf{k}_{\mathsf{id}_{\sigma(i)}}, \mathsf{m}_i)$ for all $(\mathsf{id}_{\sigma(i)}, \mathsf{m}_i) \in M \backslash \{(\mathsf{id}^*, \mathsf{m}^*)\}$ $(i \in [q])$. Then, it outputs $(\mathsf{m}^*, \mathsf{t}^*)$ and all (m, t) such that $(\mathsf{id}^*, \mathsf{m}, \mathsf{t}) \in \{(\mathsf{id}^{(i)}, \mathsf{m}^{(i)}, \mathsf{t}^{(i)})\}_{i\in[q]}$. If the output of A does not meet the conditions above, F aborts this game.

The output of F is a forgery in $\mathsf{EUF\text{-}qCMA}$ security game, since the one-more forgery $(\mathsf{m}^*, \mathsf{t}^*)$ is not in $\{(\mathsf{m}^{(1)}, \mathsf{t}^{(1)}), \ldots, (\mathsf{m}^{(q)}, \mathsf{t}^{(q)})\}$ and the other pairs can be obtained in the straightforward way. Besides, the probability that A wins without finding a forgery for MACs is at most $1/|\mathcal{T}|$. Thus, we obtain $\mathsf{Adv}^{\text{agguf-qcma}}_{\mathsf{AMAC}_{\mathrm{KL}},\mathsf{A}}(\lambda) \leq n \cdot \mathsf{Adv}^{\text{euf-qcma}}_{\mathsf{MAC},\mathsf{F}}(\lambda) + 1/|\mathcal{T}|$, and the proof is completed. □

4.5 Quantum-Secure SAMAC

We define a model of history-free SAMACs and formalize the security in the quantum security model because all existing SAMACs [44, 62, 128] are history-free. The ordinary SAMACs generate each aggregate tag depending on the local message of a tagging user, a sequence of previous messages, and an aggregate-so-far tag. On the other hand, history-free SAMACs generate each aggregate tag depending only on a local message and an aggregate-so-far tag.

4.5.1 Quantum Security of SAMAC

First, we define the model of SAMACs which was formalized in [44]. An SAMAC scheme consists of a tuple of three polynomial-time algorithms $(\mathsf{KGen}, \mathsf{STag}, \mathsf{SVrfy})$: Let λ be a security parameter, let $n = \mathsf{poly}(\lambda)$ be the number of tagging users, and a permutation $\sigma : [n] \to [n]$ denotes order information. $\mathcal{ID} = \{\mathsf{id}_i\}_{i\in[n]} \in \left(\{0,1\}^{O(\lambda)}\right)^n$ is an ID space, $\mathcal{K} = \mathcal{K}(\lambda)$ is a key space, $\mathcal{M} = \mathcal{M}(\lambda)$ is a message space, and $\mathcal{T} = \mathcal{T}(\lambda)$ is a tag space.

Key Generation. KGen is a randomized algorithm which, on input a security parameter 1^λ and an ID $\mathsf{id} \in \mathcal{ID}$, outputs a secret key $\mathsf{k}_{\mathsf{id}} \in \mathcal{K}$. We write $\mathsf{k}_{\mathsf{id}} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id})$.

Tagging. STag is an algorithm which, on input a secret key $\mathsf{k}_{\mathsf{id}} \in \mathcal{K}$, a message $\mathsf{m} \in \mathcal{M}$, and an aggregate-so-far tag $\tau' \in \mathcal{T}$, outputs an aggregate tag $\tau \in \mathcal{T}$. We write $\tau \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}}, \mathsf{m}, \tau')$. Note that the first tagging user generates an aggregate tag on a local message and an empty symbol $\emptyset_\tau \in \mathcal{T}$ as an aggregate-so-far tag.

Verification. SVrfy is a deterministic algorithm which, on input a set of secret keys $K = \{\mathsf{k}_{\mathsf{id}_i}\}_{i\in[n]}$, a sequence of arbitrary ℓ ID/message pairs $M = ((\mathsf{id}_{\sigma(i)}, \mathsf{m}_i))_{i\in[\ell]}$, an aggregate-so-far tag $\tau' \in \mathcal{T}$, and an aggregate tag $\tau \in \mathcal{T}$, outputs 1 (accept) or 0 (reject). We write $1/0 \leftarrow \mathsf{SVrfy}(K, (M, \tau'), \tau)$.

We require that SAMAC schemes $(\mathsf{KGen}, \mathsf{STag}, \mathsf{SVrfy})$ meet correctness in the following way: For any set $K = \{\mathsf{k}_{\mathsf{id}_i}\}_{i\in[n]}$ of secret keys $(\forall \mathsf{id}_i \in \mathcal{ID}$, $\mathsf{k}_{\mathsf{id}_i} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id}_i))$, any sequence $M = ((\mathsf{id}_{\sigma(i)}, \mathsf{m}_i))_{i\in[\ell]}$ of ID/message pairs, and any aggregate-so-far tag $\tau' \in \mathcal{T}$, it holds that $1 = \mathsf{SVrfy}(K, (M, \tau'), \tau)$, where $\tau \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(\ell)}}, \mathsf{m}_\ell, \mathsf{STag}(\ldots \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(1)}}, \mathsf{m}_1, \tau') \ldots))$.

Next, we define the quantum security of SAMACs: *sequential aggregate unforgeability against quantum chosen message attacks*, which we call $\mathsf{saggUF\text{-}qCMA}$ *security.*

We define a sequential aggregation algorithm SeqAgg_K and a closure $\mathsf{Closure}$ in order to define $\mathsf{saggUF\text{-}qCMA}$ security. For an SAMAC scheme $\mathsf{SAMAC} = (\mathsf{KGen}, \mathsf{STag}, \mathsf{SVrfy})$, a deterministic or randomized algorithm SeqAgg_K with secret keys $K = \{\mathsf{k}_{\mathsf{id}_i}\}_{i\in[n]}$ is defined as follows:

Definition 4.3 (Sequential Aggregation Algorithm)**.** *Given a permutation $\sigma : [n] \to [n]$, a sequence $\boldsymbol{m} = (\mathsf{m}_1, \ldots, \mathsf{m}_\ell)$ of messages, an aggregate-so-far tag $\tau' \in \mathcal{T}$, and a sequence $\boldsymbol{r} = (r_1, \ldots, r_\ell)$ of random coins, a sequential aggregation algorithm outputs the aggregate tag*

$$\tau \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(\ell)}}, \mathsf{m}_\ell, \mathsf{STag}(\ldots \mathsf{m}_2, \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(1)}}, \mathsf{m}_1, \tau'; r_1)\ldots); r_\ell)$$

on the given messages/tag sequence $((\mathsf{m}_1, \ldots, \mathsf{m}_\ell), \tau'; \boldsymbol{r})$. Then, we write $\tau \leftarrow \mathsf{SeqAgg}_K(\sigma, \boldsymbol{m}, \tau')$ as the sequential aggregation algorithm.

And, we define $\mathsf{Closure}$ in the same way as the closure defined in [44].

Definition 4.4 (Closure)**.** *We define a set* $\mathsf{Trivial}$ *to formalize* $\mathsf{Closure}$*. Let L_{Tag} be a set of pairs $((M, \tau'), \tau)$, where $M = ((\mathsf{id}_{\sigma(i)}, \mathsf{m}_i))_{i\in[\ell]}$ is a sequence of ID/message pairs, τ' is an aggregate-so-far tag, and τ is an aggregate tag on (M, τ'). Let L_{Cor} be a set of corrupted IDs.* $\mathsf{Trivial}$ *is defined as follows:*

$$\begin{aligned}\mathsf{Trivial}_{L_{\mathsf{Tag}}, L_{\mathsf{Cor}}}(M, \tau) :=& \{M\} \cup \bigcup_{((\hat{M}, \tau), \hat{\tau}) \in L_{\mathsf{Tag}}} \mathsf{Trivial}_{L_{\mathsf{Tag}}, L_{\mathsf{Cor}}}(M \parallel \hat{M}, \hat{\tau}) \\ &\cup \bigcup_{\substack{\forall \bar{\mathsf{m}} \in \mathcal{M}, \bar{\tau} \in \mathcal{T}, \\ \mathsf{id} \in L_{\mathsf{Cor}}}} \mathsf{Trivial}_{L_{\mathsf{Tag}}, L_{\mathsf{Cor}}}(M \parallel (\mathsf{id}, \bar{\mathsf{m}}), \bar{\tau}).\end{aligned}$$

$\mathsf{Closure}$ *is defined as follows: Let $\emptyset_{\mathsf{m}}$ be an empty symbol in $\mathcal{M}$ and let $\emptyset_\tau$ be an empty symbol in $\mathcal{T}$, then let* $\mathsf{Closure}(L_{\mathsf{Tag}}, L_{\mathsf{Cor}}) := \{\mathsf{Trivial}_{L_{\mathsf{Tag}}, L_{\mathsf{Cor}}}(\emptyset_{\mathsf{m}}, \emptyset_\tau)\}$.

Then, we define $\mathsf{saggUF\text{-}qCMA}$ security by using SeqAgg_K and $\mathsf{Closure}$.

Definition 4.5 ($\mathsf{saggUF\text{-}qCMA}$ security)**.** *An SAMAC scheme* $\mathsf{SAMAC} = (\mathsf{KGen}, \mathsf{STag}, \mathsf{SVrfy})$ *meets* $\mathsf{saggUF\text{-}qCMA}$ *security, if for any QPT algorithm* A *against* SAMAC, $\mathsf{Adv}^{\text{sagguf-qcma}}_{\mathsf{SAMAC},\mathsf{A}}(\lambda) := \Pr[\mathsf{A}\ \textit{wins}]$ *is negligible, where* $[\mathsf{A}\ \textit{wins}]$ *is the event that* A *wins in the following game:*

Setup: *Generate secret keys $\mathsf{k}_{\mathsf{id}_i} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id}_i)$ for all $\mathsf{id}_i \in \mathcal{ID}$. Set a list $L_{\mathsf{Cor}} \leftarrow \emptyset$.*

Corrupt: *When* A *submits a query $\mathsf{id} \in \mathcal{ID}$, a corrupt oracle $\mathsf{O}_{\mathsf{Cor}}$ returns the corresponding key k_{id} and sets $L_{\mathsf{Cor}} \leftarrow L_{\mathsf{Cor}} \cup \{\mathsf{id}\}$.*

Tagging: A *submits a permutation* $\sigma : [n] \to [n]$ *(classical data) and a superposition of message/previous-tag pairs*

$$\sum_{\boldsymbol{m}\in\mathcal{M}^{\ell},\tau'\in\mathcal{T},t\in\mathcal{T},z} \psi_{\boldsymbol{m},\tau',t,z}\,|(\boldsymbol{m},\tau'),t,z\rangle$$

to tagging oracle $\mathsf{O}_{\mathsf{Tag}}$*, where* ℓ *is an integer such that* $1 \leq \ell \leq n$*, a permutation* $\sigma : [n] \to [n]$ *is order-information of IDs, and* $\boldsymbol{m} = (\mathsf{m}_i)_{i\in[\ell]}$ *is a sequence of messages. Then,* $\mathsf{O}_{\mathsf{Tag}}$ *chooses randomness* $\boldsymbol{r} = (r_1, \ldots, r_\ell)$ *used in* STag *algorithm, where it does not need to choose* r *if* STag *is deterministic, and returns*

$$\sum_{\boldsymbol{m}\in\mathcal{M}^{\ell},\tau'\in\mathcal{T},t\in\mathcal{T},z} \psi_{\boldsymbol{m},\tau',t,z}\,|(\boldsymbol{m},\tau'),t\oplus\mathsf{SeqAgg}_K(\sigma,\boldsymbol{m},\tau';\boldsymbol{r}),z\rangle\,.$$

A *submits at most* q *queries to* $\mathsf{O}_{\mathsf{Tag}}$ *and it is not allowed to issue queries to* $\mathsf{O}_{\mathsf{Cor}}$ *after querying to* $\mathsf{O}_{\mathsf{Tag}}$*.*

Output: A *outputs* $(q+1)$ *tuples of ID/message sequences, aggregate-so-far tags, and aggregate tags* $((M_1,\tau'_1),\tau_1),\ldots,((M_{q+1},\tau'_{q+1}),\tau_{q+1})$*.* A *wins if the following holds:*

- *For all* $i \in [q+1]$*,* $1 \leftarrow \mathsf{SVrfy}(K,(M_i,\tau'_i),\tau_i)$ *holds.*
- *For all* $i \in [q+1]$*,* $(M_i,\tau'_i) \notin \mathsf{Closure}(L^{(i)}_{\mathsf{Tag}}, L_{\mathsf{Cor}})$ *holds, where* $L^{(i)}_{\mathsf{Tag}} := \left\{((M_j,\tau'_j),\tau_j)\right\}_{j\in[q+1]} \setminus \{((M_i,\tau'_i),\tau_i)\}$*.*

We explain that Definition 4.5 can be viewed as an extension from both security notions of [24] and [44].

- Consider a special case where the number of IDs is 1 (i.e., $n = 1$) in Definition 4.5. Suppose that, in the $\mathsf{saggUF\text{-}qCMA}$ security game, a QPT adversary A outputs q tuples of ID/message pairs, aggregate-so-far tags, and aggregate tags $((\mathsf{id}_1,\mathsf{m}_1),\tau'_1,\tau_1)$, …, $((\mathsf{id}_1,\mathsf{m}_{q+1}),\tau'_{q+1},\tau_{q+1})$ for the same ID id_1. Then, A wins if $1 \leftarrow \mathsf{SVrfy}(\mathsf{k}_{\mathsf{id}_1},(\mathsf{m}_i,\tau'_i),\tau_i)$ and $((\mathsf{id}_1,\mathsf{m}_i),\tau'_i) \notin \mathsf{Closure}(L^{(i)}_{\mathsf{Tag}},\emptyset_\tau)$ for all $i \in [q+1]$, where $\mathsf{Closure}(L^{(i)}_{\mathsf{Tag}},\emptyset_\tau) = \{(\mathsf{id}_1,\mathsf{m}_j),\tau'_j)\}_{j\in[q+1]}\backslash\{(\mathsf{id}_1,\mathsf{m}_i),\tau'_i)\}$. This is the same as Definition 4.1 since we can view $\mathsf{m}_i\|\tau'_i$ as messages for all $i \in [q+1]$, and the outputted messages $\mathsf{m}_i\|\tau'_i$ are different one another. Hence, Definition 4.5 is regarded as an extension of the quantum security of MACs in [24].
- Consider a special case where PPT algorithm A obtains valid q tuples of ID/message pairs, aggregate-so-far tags, and aggregate tags $((M_1,\tau'_1),\tau_1)$, …, $((M_q,\tau'_q),\tau_q)$ by having access to the oracle $\mathsf{O}_{\mathsf{Tag}}$ with classical queries $(M_1,\tau'_1),\ldots,(M_q,\tau'_q)$. Suppose that, in the $\mathsf{saggUF\text{-}qCMA}$ security game, a PPT algorithm A finally outputs q tuples of ID/message

pairs, aggregate-so-far tags, and aggregate tags $((M_1, \tau'_1), \tau_1), \ldots, ((M_q, \tau'_q), \tau_q)$ which he obtained, and $((M_{q+1}, \tau'_{q+1}), \tau_{q+1})$, where $M_{q+1} = ((\mathsf{id}_{\sigma(i)}, \mathsf{m}_i))_{i\in[\ell]}$ $(1 \leq \ell \leq n)$ is a sequence of arbitrary ℓ pairs of IDs and messages and τ_{q+1} is an aggregate tag on (M_{q+1}, τ'_{q+1}). Then, A wins if we have $1 \leftarrow \mathsf{SVrfy}(K, (M_{q+1}, \tau'_{q+1}), \tau_{q+1})$ and $(M_{q+1}, \tau'_{q+1}) \notin \mathsf{Closure}(L_{\mathsf{Tag}}^{(q+1)}, L_{\mathsf{Cor}})$. This is the same as the security definition of SAMACs in [44], and ours is an extension of it.

In terms of quantum security mentioned above, we analyze security of known SAMACs, and the results are summarized in Table 4.1. In particular, we can break the security of SAMACs of [44, 128] by using quantum algorithms proposed in [24, 80]. In the next section, we propose secure constructions of SAMACs in terms of quantum security mentioned above.

We propose two generic constructions SAMAC_1 and SAMAC_2 of (history-free) SAMACs and show that these constructions meet $\mathsf{saggUF\text{-}qCMA}$ security.

4.5.2 Quantum Algorithms against Existing SAMACs

We describe the attack against the existing sequential aggregate authentication schemes of [44, 128].

The Attack against the Scheme of [44]

The algorithm breaking the scheme of [44] follows the quantum attack against CBC-MAC of [80]. First, we define Simon's algorithm used by the one against the scheme of [44]. Simon's algorithm is a quantum algorithm solving the following problem.

Definition 4.6 (Simon's Problem)**.** *Given a Boolean function $f : \{0,1\}^n \to \{0,1\}^n$ and the promise that there exists $s \in \{0,1\}^n$ such that for any $(x, y) \in \{0,1\}^n$, $[f(x) = f(y)] \Leftrightarrow [x \oplus y \in \{0^n, s\}]$, the goal is to find s.*

Simon's algorithm is as follows:

1. Set the following $2n$-qubit: $\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle |0\rangle$.
2. Submit a quantum query $\frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle |f(x)\rangle$ to the function f.
3. Measure the second register in the computational basis and obtain a value $f(z)$. Then, from the promise $f(x) = f(x \oplus s)$, the first register is as follows:
$$\frac{1}{\sqrt{2}}(|z\rangle + |z \oplus s\rangle).$$
4. Apply the Hadamard transformation to the first register and get
$$\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{y\cdot z}(1 + (-1)^{y\cdot s}) |y\rangle .$$

5. Measure the register and obtain a vector y.

The obtained vector y meets $y \cdot s = 0$ since if the amplitude of y such that $y \cdot s = 1$ is 0. By replying the above process, we have $O(n)$ vectors y such that $y \cdot s = 0$. Therefore, we can recover s.

Let $\varepsilon(f, s) := \max_{t \in \{0,1\}^n \setminus \{0,s\}} \Pr_x[f(x) = f(x \oplus t)]$ for a function $f : \{0,1\}^n \to \{0,1\}^n$ meeting the promise of Simon's algorithm ($f(x \oplus s) = f(x)$ for all x). From [80], the success probability of Simon's algorithm is as follows.

Proposition 4.1 (Theorem 1 in [80]). *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a function such that $f(x \oplus s) = f(x)$ for all x, and let c be a positive integer. If $\varepsilon(f, s) \leq p_0 < 1$ holds for probability p_0, then Simon's algorithm returns s with cn queries, with probability at least $1 - \left(2\left(\frac{1+p_0}{2}\right)^c\right)^n$.*

Moreover, the following existing result shows that if we select a suitable random value t, $f(x \oplus t) = f(x)$ holds with high probability.

Proposition 4.2 (Theorem 2 in [80]). *After cn steps of Simon's algorithm, if t is orthogonal to all vectors u_i returned by each step of the algorithm, then $\Pr_x[f(x \oplus t) = f(x)] \geq p_0$ with probability at least $1 - \left(2\left(\frac{1+p_0}{2}\right)^c\right)^n$.*

Next, we describe the SAMAC scheme $\mathsf{SAMAC}_{ex} = (\mathsf{KGen}, \mathsf{STag}, \mathsf{SVrfy})$ of [44] as follows: Let $(\mathsf{Tag}, \mathsf{Vrfy})$ be a deterministic MAC with a key space $\mathcal{K}_{\mathsf{MAC}}$ and a tag space $\mathcal{T}$, and let $\mathsf{PRP} : \mathcal{K}_{\mathsf{PRP}} \times \mathcal{T} \to \mathcal{T}$ be a pseudorandom permutation.

- $\mathsf{k}_{\mathsf{id}} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id})$: Generate keys $\mathsf{k}_{\mathsf{MAC}} \xleftarrow{U} \mathcal{K}_{\mathsf{MAC}}$ and $\mathsf{k}_{\mathsf{PRP}} \xleftarrow{U} \mathcal{K}_{\mathsf{PRP}}$. Output $\mathsf{k}_{\mathsf{id}} := (\mathsf{k}_{\mathsf{MAC}}, \mathsf{k}_{\mathsf{PRP}})$.
- $\tau \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}}, \mathsf{m}, \tau')$:
  Compute $t \leftarrow \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m})$, and then output $\tau \leftarrow \mathsf{PRP}(\mathsf{k}_{\mathsf{PRP}}, t \oplus \tau')$.
- $1/0 \leftarrow \mathsf{SVrfy}(K, (M, \tau'), \tau)$:
  Compute $\tilde{\tau} \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(\ell)}}, \mathsf{m}_\ell, \mathsf{STag}(\ldots, \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(1)}}, \mathsf{m}_1, \tau') \ldots))$. Output 1 if $\tau = \tilde{\tau}$, or output 0 otherwise.

Finally, we describe the attack against SAMAC_{ex}. We fix two arbitrary messages $\mathsf{m}_0, \mathsf{m}_1 \in \mathcal{M}$ ($\mathsf{m}_0 \neq \mathsf{m}_1$), and the function of Simon's problem is defined as follows:

$$\begin{aligned} f : \{0,1\} \times \mathcal{M} &\to \mathcal{T} \\ (b, \tau') &\mapsto \mathsf{PRP}(\mathsf{k}_{\mathsf{PRP}}, \tau' \oplus \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_b)) \end{aligned}$$

For $s = 1 \parallel \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_0) \oplus \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_1)$, the function f meets the promise of Simon's problem:

$$f(0, \tau') = \mathsf{PRP}(\mathsf{k}_{\mathsf{PRP}}, \tau' \oplus \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_1)),$$

$$f(1,\tau') = \mathsf{PRP}(\mathsf{k}_{\mathsf{PRP}}, \tau' \oplus \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_0)),$$
$$f(b,\tau') = f(b \oplus 1, \tau' \oplus \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_0) \oplus \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_1)).$$

Then, we can generate the following forgery against SAMAC_{ex}:

1. Fix $\mathsf{m}_0, \mathsf{m}_1$ as the messages of a message block, and let a previous tag $\tau' = 0^n \in \mathcal{T}$ denote a n-bit string of 0.
2. Submit a classical query $\mathsf{m}_0 \parallel 0^n$ to the tagging oracle of saggUF-qCMA security game, and receive the aggregate tag τ.
3. By using Simon's algorithm with $O(n)$ quantum queries, obtain $s = \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_0) \oplus \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_1)$.
4. Output a forgery $(\mathsf{m}_1 \parallel \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_0) \oplus \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_1), \tau)$ as a valid aggregate tag.

The above forgery is valid, since $\mathsf{m}_1 \parallel \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_0) \oplus \mathsf{Tag}(\mathsf{k}_{\mathsf{MAC}}, \mathsf{m}_1)$ has never been queried.

The Attack against the Scheme of [128]

We describe two schemes presented in [128]. Let $\mathbb{F}_p$ be a finite field with a prime power p. The first construction is as follows:

- $\mathsf{k}_{\mathsf{id}} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id})$: Output a secret key $\mathsf{k}_{\mathsf{id}} := (a, b) \xleftarrow{U} \mathbb{F}_p^2$.
- $\tau \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}}, \mathsf{m}, \tau')$: On input a message $\mathsf{m} \in \mathbb{F}_p$ and an aggregate-so-far tag $\tau' \in \mathbb{F}_p$, output a tag $\tau := a \cdot \mathsf{m} + b + \tau' \in \mathbb{F}_p$.
- $1/0 \leftarrow \mathsf{SVrfy}(K, (M, \tau'), \tau)$:
  Compute $\tilde{\tau} \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(\ell)}}, \mathsf{m}_\ell, \mathsf{STag}(\ldots, \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(1)}}, \mathsf{m}_1, \tau')\ldots))$. Output 1 if $\tau = \tilde{\tau}$, or output 0 otherwise.

And, the second construction is described as follows:

- $\mathsf{k}_{\mathsf{id}} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id})$: Output a secret key $\mathsf{k}_{\mathsf{id}} := (a, b, c) \xleftarrow{U} \mathbb{F}_p^3$.
- $\tau \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}}, \mathsf{m}, \tau')$: On input a message $\mathsf{m} \in \mathbb{F}_p$, an ID $\mathsf{id} \in \mathbb{F}_p$, and an aggregate-so-far tag $\tau' = (s', t') \in \mathbb{F}_p^2$, output a tag $\tau := (a \cdot \mathsf{m} + b + s', a \cdot \mathsf{id} + c + t') \in \mathbb{F}_p^2$.
- $1/0 \leftarrow \mathsf{SVrfy}(K, (M, \tau'), \tau)$:
  Compute $\tilde{\tau} \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(\ell)}}, \mathsf{m}_\ell, \mathsf{STag}(\ldots, \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(1)}}, \mathsf{m}_1, \tau')\ldots))$. Output 1 if $\tau = \tilde{\tau}$, or output 0 otherwise.

Regarding both schemes, we can view aggregate tags as the values of pairwise independent hash functions $h(x) = ax + b$ with $a, b \in \mathbb{F}_p$. In the straightforward way, we can apply the quantum algorithm in the proof of Lemma 6.3 in [24]. In this case, adversaries can get secret keys $(a, b) \in \mathbb{F}_p^2$ with non-negligible probability and generate forgeries obviously even if they submit only one quantum query. Therefore, the schemes of [128] do not meet the one-time security formalized in Section 4.5.1.

4.5.3 SAMAC from Quantum-Secure Pseudorandom Function

We construct a generic construction SAMAC_1 starting from any QPRF. The idea is as follows: Regarding [44], it is shown that there exists an SAMAC if there exists a partial invertible MAC which can recover partial messages from MAC tags and the other parts of messages. The paper [44] generally presented a partial invertible MAC from the ordinary MACs and pseudorandom permutations. However, in order to construct SAMACs, it is enough to use a PRF. This is because it is known that (quantum) PRFs can be used as EUF-(q)CMA secure MACs [16, 24]. Hence, it is possible to construct a quantum-secure SAMAC if a PRF meets the quantum security.

Let $\mathsf{PRF} : \mathcal{K} \times \mathcal{M} \times \mathcal{T} \to \mathcal{T}$ be a QPRF. Then, $\mathsf{SAMAC}_1 = (\mathsf{KGen}, \mathsf{STag}, \mathsf{SVrfy})$ is constructed as follows:

- $\mathsf{k}_{\mathsf{id}} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id})$: Choose a secret key $\mathsf{k} \in \mathcal{K}$ uniformly at random, and output $\mathsf{k}_{\mathsf{id}} := \mathsf{k}$.
- $\tau \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}}, \mathsf{m}, \tau')$: Compute $\tau \leftarrow \mathsf{PRF}(\mathsf{k}_{\mathsf{id}}, \mathsf{m} \parallel \tau')$, and output τ.
- $1/0 \leftarrow \mathsf{SVrfy}(K, (M, \tau'), \tau)$: Given a sequence $M = ((\mathsf{id}_{\sigma(i)}, \mathsf{m}_i))_{i \in [\ell]}$ and an aggregate-so-far tag τ', do the following:
  1. $\tilde{\tau} \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(\ell)}}, \mathsf{m}_\ell, \mathsf{STag}(\ldots \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(1)}}, \mathsf{m}_1, \tau') \ldots))$.
  2. Output 1 if $\tau = \tilde{\tau}$, or output 0 otherwise.

Theorem 4.2. *If* PRF *is a quantum-secure pseudorandom function, then* SAMAC_1 *satisfies* $\mathsf{saggUF\text{-}qCMA}$ *security.*

Proof. Let A be a QPT adversary against SAMAC_1, let $|\tau|$ be the bit-length of aggregate tags, and let q be the number of queries which A issues to $\mathsf{O}_{\mathsf{Tag}}$.

We consider any QPT adversary A which generates one-more forgery on an ID/message sequence including a target subsequence $M^*_{j,k} := ((\mathsf{id}^*_j, \mathsf{m}^*_j), \ldots, (\mathsf{id}^*_k, \mathsf{m}^*_k))$. Target subsequence is defined as follows: We assume that A generates a forgery on a sequence $((M_{i^*}, \tau'_{i^*}), \tau_{i^*})$ $(i^* \in [q+1])$. The target subsequence $M^*_{j,k}$ is included in M_{i^*} and satisfies the following:

- It is not in $\mathsf{Trivial}_{L^{(i^*)}_{\mathsf{Tag}}, L_{\mathsf{Cor}}}(\emptyset_{\mathsf{m}}, \emptyset_\tau)$.
- It contains only not corrupted IDs.

- There do not exist j', k' such that $1 \leq j \leq j' \leq k' \leq k$ and $((\mathsf{id}^*_{j'}, \mathsf{m}_{j'}), \ldots, (\mathsf{id}^*_{k'}, \mathsf{m}_{k'})) \in \mathsf{Trivial}_{L^{(i^*)}_{\mathsf{Tag}}, L_{\mathsf{Cor}}}(\emptyset_{\mathsf{m}}, \emptyset_{\tau})$.
- There do not exist j', k' such that $1 \leq j' \leq j \leq k'$ and $((\mathsf{id}^*_{j'}, \mathsf{m}_{j'}), \ldots, (\mathsf{id}^*_{k'}, \mathsf{m}_{k'})) \in \mathsf{Trivial}_{L^{(i^*)}_{\mathsf{Tag}}, L_{\mathsf{Cor}}}(\emptyset_{\mathsf{m}}, \emptyset_{\tau})$.
- There do not exist j', k' such that $1 \leq j' \leq k \leq k' \leq \ell^*$ and $((\mathsf{id}^*_{j'}, \mathsf{m}_{j'}), \ldots, (\mathsf{id}^*_{k'}, \mathsf{m}_{k'})) \in \mathsf{Trivial}_{L^{(i^*)}_{\mathsf{Tag}}, L_{\mathsf{Cor}}}(\emptyset_{\mathsf{m}}, \emptyset_{\tau})$, where ℓ^* is the maximum of the length of an ID/message sequence.

And then, we classify the event that A wins in the security game as some events by using target subsequence, and prove that the probabilities that these events occur are negligible. Regarding A's output $\{((M_i, \tau'_i), \tau_i)\}_{i \in [q+1]}$, we consider the following events:

- [Coll]: A outputs $\{((M_i, \tau'_i), \tau_i)\}_{i \in [q+1]}$ by finding a collision pair $(\mathsf{m} \parallel \tau', \hat{\mathsf{m}} \parallel \hat{\tau}')$ of SAMAC_1 for an ID $\mathsf{id} \in \mathcal{ID}$.
- [Suff]: A outputs $\{((M_i, \tau'_i), \tau_i)\}_{i \in [q+1]}$ such that there exists a target sequence $M^*_{j,k}$ in a sequence M_{i^*} ($i^* \in [q+1]$), which is a suffix of an ID/message sequence in A's output.
- [Pref]: A outputs $\{((M_i, \tau'_i), \tau_i)\}_{i \in [q+1]}$ such that there exists a target sequence $M^*_{j,k}$ in a sequence M_{i^*} ($i^* \in [q+1]$), which is a prefix of an ID/message sequence in A's output.
- [New]: A outputs $\{((M_i, \tau'_i), \tau_i)\}_{i \in [q+1]}$ such that there exists a target sequence $M^*_{j,k}$ in a sequence M_{i^*} ($i^* \in [q+1]$), which is neither suffix nor prefix of an ID/message sequence in A's output.

Then, we have the following advantage:

$$\begin{aligned}\mathsf{Adv}^{\text{sagguf-qcma}}_{\mathsf{SAMAC}_1,\mathsf{A}}(\lambda) \leq\ & \Pr[\mathsf{Coll}] + \Pr[\mathsf{Suff} \mid \neg\mathsf{Coll}] \\ & + \Pr[\mathsf{Pref} \mid \neg\mathsf{Coll} \wedge \neg\mathsf{Suff}] + \Pr[\mathsf{New} \mid \neg\mathsf{Coll} \wedge \neg\mathsf{Suff} \wedge \neg\mathsf{Pref}].\end{aligned}$$

Proof of [Coll]: By using A which outputs a forgery meeting the condition of [Coll], we construct a PPT algorithm D_c breaking a PRF in the following way: It is given the oracle $\mathsf{O}_{\mathsf{PRF}}$ in the security game of QPRFs.

Setup: Set secret keys as follows:

1. $\mathsf{id}^* \xleftarrow{U} \mathcal{ID}$ and assign $\mathsf{O}_{\mathsf{PRF}}$ to the PRF of id^*.
2. For all $\mathsf{id} \in \mathcal{ID} \backslash \{\mathsf{id}^*\}$, $\mathsf{k}_{\mathsf{id}} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id})$.

Corrupt: For each query id, return the key k_{id} and set $L_{\mathsf{Cor}} \leftarrow L_{\mathsf{Cor}} \cup \{\mathsf{id}\}$.

Tagging: For each query $\left(\sigma, \sum \psi_{\boldsymbol{m},\tau',t,z} |(\boldsymbol{m}, \tau'), t, z\rangle\right)$, simulate as follows:

1. Compute each $\mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(i)}}, \cdot, \cdot)$ algorithm, in the following way:
   - If $\mathsf{id}_{\sigma(i)} = \mathsf{id}^*$, generate a tag by using $\mathsf{O}_{\mathsf{PRF}}(\cdot)$.
   - If $\mathsf{id}_{\sigma(i)} \neq \mathsf{id}^*$, generate a tag by using $\mathsf{PRF}(\mathsf{k}_{\mathsf{id}_{\sigma(i)}}, \cdot)$.
2. Return $\sum \psi_{\boldsymbol{m},\tau',t,z} |(\boldsymbol{m}, \tau'), t \oplus \mathsf{SeqAgg}_K(\sigma, \boldsymbol{m}, \tau'), z\rangle$.

Output: When A outputs $\{((M_i, \tau_i'), \tau_i)\}_{i\in[q+1]}$, do the following:

1. For all (M, τ') including id^*, compute aggregate tags generated by id^*.
2. Find a pair $(m, \tau'), (\hat{m}, \hat{\tau}')$ such that $(m, \tau') \neq (\hat{m}, \hat{\tau}')$ and $\mathsf{O}_{\mathsf{PRF}}(m \parallel \tau') = \mathsf{O}_{\mathsf{PRF}}(\hat{m} \parallel \hat{\tau}')$.
3. If there exists such a pair, output 1. Otherwise, output 0.

D_c simulates the environment of A completely since it has secret keys and the oracle in the security game of PRFs. If A can find a collision of PRF, D_c can also break the security of PRF obviously. Because the probability that A finds a collision in the straightforward way is $O(q^3 \cdot 2^{-|\tau|})$ from [138], we have $\Pr[\mathsf{Coll}] \leq n(q+1) \cdot \mathsf{Adv}^{\mathrm{qpr}}_{\mathsf{PRF},\mathsf{D}_c}(\lambda) + O(q^3 \cdot 2^{-|\tau|})$. ■

Proof of $[\mathsf{Suff}|\neg\mathsf{Coll}]$: We consider the case where id^*_{j-1} is corrupted for a target sequence $M^*_{j,k}$, or the case where id^*_j is the first order of another sequence including $M^*_{j,k}$. In these cases, $M^*_{j,k}$ is not any suffix of other ID/message-sequences. Thus, the event $[\mathsf{Suff}]$ does not happen. If id^*_{j-1} is not corrupted, $M^*_{j-1,k}$ must be the target subsequence from the condition that event $[\mathsf{Coll}]$ does not happen. By replying this, however, the obtained $M^*_{1,k}$ does not meet the condition of target subsequences. From this contradiction, event $[\mathsf{Suff}|\neg\mathsf{Coll}]$ does not happen. That is, $\Pr[\mathsf{Suff}|\neg\mathsf{Coll}] = 0$ holds. ■

Proof of $[\mathsf{Pref}|\neg\mathsf{Coll} \wedge \neg\mathsf{Suff}]$: We construct D_p breaking a QPRF in the same way as the algorithm above D_c except for the process in **Output** phase. When A outputs $\{((M_i, \tau_i'), \tau_i)\}_{i\in[q+1]}$ in **Output** phase, it does the following:

1. Find a pair $((M^*, \tau'^*), \tau^*)$ such that M^* includes a target sequence $M^*_{j,k}$ meeting the condition of $[\mathsf{Pref}|\neg\mathsf{Coll} \wedge \neg\mathsf{Suff}]$, and id^* equals to id^*_k of $M^*_{j,k}$.
2. Generate an aggregate tag $\bar{\tau}^*$ on (M^*, τ'^*) by using $\mathsf{O}_{\mathsf{PRF}}$.
3. If $\tau^* = \bar{\tau}^*$, output 1. Otherwise, output 0.

D_p simulates the view of A and breaks the quantum security of PRF. Then, we have $\Pr[\mathsf{Pref}|\neg\mathsf{Coll} \wedge \neg\mathsf{Suff}] \leq n(q+1) \cdot \mathsf{Adv}^{\mathrm{qpr}}_{\mathsf{PRF},\mathsf{D}_p}(\lambda) + 2^{-|\tau|/2}$. ■

Proof of $[\mathsf{New}|\neg\mathsf{Coll} \wedge \neg\mathsf{Suff} \wedge \neg\mathsf{Pref}]$: In the same way as the proof in $[\mathsf{Pref}|\neg\mathsf{Coll} \wedge \neg\mathsf{Suff}]$, we can show that the event happens with negligible probability. It is possible to construct a PPT algorithms D_n in the same way as D_p except for the way to choose the target sequence. That is, D_n chooses a target

sequence which is neither suffix nor prefix of another ID/message-sequence and checks whether it is a valid tag. Hence, we have $\Pr[\mathsf{New}|\neg\mathsf{Coll}\wedge\neg\mathsf{Suff}\wedge\neg\mathsf{Pref}] \leq n(q+1)\cdot \mathsf{Adv}^{\mathrm{qpr}}_{\mathsf{PRF},\mathsf{D}_n}(\lambda) + \frac{1}{2^{|\tau|/2}}$. ■

From the above, we obtain the following advantage:

$$\mathsf{Adv}^{\text{sagguf-qcma}}_{\mathsf{SAMAC}_1,\mathsf{A}}(\lambda) \leq 3n(q+1)\cdot \mathsf{Adv}^{\mathrm{qpr}}_{\mathsf{PRF},\mathsf{D}}(\lambda) + O(q^3\cdot 2^{-|\tau|}).$$

Therefore, the proof is completed. □

In order to obtain quantum-secure constructions of SAMACs based on SAMAC_1, we can apply the quantum-secure PRF of [136, 124] to SAMAC_1, since those are deterministic. More specifically, we can apply NMAC/HMAC to SAMAC_1 as a quantum PRF, since these MACs are shown to be quantum PRFs in [124].

4.5.4 SAMAC from Randomized Pseudorandom Generator

We construct an SAMAC scheme SAMAC_2 starting from any randomized PRG and any PRF. This scheme is based on the GGM (quantum) pseudorandom function [55, 136]. The difference between the GGM construction and ours is that a deterministic PRG is used in the GGM construction, whereas a randomized PRG is used in SAMAC_2.

Although one may think that we can realize quantum-secure SAMAC schemes by applying randomized PRGs to the GGM pseudorandom function, there exists a problem. This problem is that each tagging user has to append a randomness to his/her aggregate tag. Namely, a tagging user generates an aggregate tag $\tau_1 = (r_1, GGM(\mathsf{k}_1, \mathsf{m}_1\|\tau'; r_1))$, and the next user generates his/her tag $\tau_2 = (r_1, r_2, GGM(\mathsf{k}_2, \mathsf{m}_2\|\tau_1; r_2))$ so that a verifier can check whether $(\mathsf{m}_1, \mathsf{m}_2, \tau')$ and $GGM(\mathsf{k}_2, \mathsf{m}_2\|\tau_1; r_2)$ are valid. Here, a function $GGM(\cdot)$ is the GGM pseudorandom function, r_1, r_2 are randomness used in underlying PRGs, $\mathsf{k}_1, \mathsf{k}_2$ are the seeds of PRGs, $\mathsf{m}_1, \mathsf{m}_2$ are local messages, and τ' is an aggregate-so-far tag. In this case, the size of aggregate tags increases every time tagging users generate aggregate tags. Therefore, the size depends on the number of tagging users.

In order to resolve this problem, we utilize a value $r \leftarrow \mathsf{PRF}(\mathsf{k}_{\mathsf{PRF}}, c)$ as randomness, where $\mathsf{PRF}(\mathsf{k}_{\mathsf{PRF}}, \cdot)$ is a classical PRF, and c is a counter value which is a component of tags. This counter value is shared among tagging users and updated after sending an aggregate tag to a verifier. And, each counter value is used only once for a sequential aggregate tag. The value r is used as follows: r is the randomness used in randomized PRGs. The tag-size does not depend on the number of tagging users, since it is possible to obtain each r from a counter c and each PRF PRF.

Note that it is natural to use (counter) values shared among tagging users in the model of SAMACs, since users are synchronized basically and the same situation using common values has been considered in previous works such

as counter-based aggregate MACs [44], and synchronized aggregate signatures [4, 68]. We use the following primitives and parameters:

- Let $G : \mathcal{X} \times \mathcal{R} \to \mathcal{X}^2$ be a randomized PRG with a set $\mathcal{R}$ of randomness used in G. Then, we write $G(x; r) = (G_0(x; r), G_1(x; r))$, where G_0, G_1 are functions from $\mathcal{X} \times \mathcal{R}$ to $\mathcal{X}$.
- Let $\mathsf{PRF} : \mathcal{K} \times \mathcal{C} \to \mathcal{R}^\mu$ be a (classical) PRF.
- Let c be a counter value in a space $\mathcal{C}$.
- $L_c \leftarrow \emptyset$ is a list of counter values and shared among tagging users.

$\mathsf{SAMAC}_2 = (\mathsf{KGen}, \mathsf{STag}, \mathsf{SVrfy})$ is constructed as follows:

- $\mathsf{k}_{\mathsf{id}} \leftarrow \mathsf{KGen}(1^\lambda, \mathsf{id})$: Choose $x \in \mathcal{X}$ and $\mathsf{k}_{\mathsf{PRF}} \in \mathcal{K}$ uniformly at random. Output $\mathsf{k}_{\mathsf{id}} := (x, \mathsf{k}_{\mathsf{PRF}})$.
- $\tau \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}}, \mathsf{m}, \tau')$: Generate an aggregate tag as follows.
  1. Split τ' into (c, y').
  2. $(r_i)_{i\in[\mu]} \leftarrow \mathsf{PRF}(\mathsf{k}_{\mathsf{PRF}}, c) \in \mathcal{R}^\mu$.
  3. $(z_i)_{i\in[\mu]} \leftarrow \mathsf{m} \parallel y' \in \{0,1\}^\mu$.
  4. $y \leftarrow G_{z_1}(\ldots G_{z_{\mu-1}}(G_{z_\mu}(x; r_\mu); r_{\mu-1}) \ldots; r_1)$.
  5. Output $\tau := (c, y)$.
- $1/0 \leftarrow \mathsf{SVrfy}(K, (M, \tau'), \tau)$: Verify a message/previous-tag pair (M, τ') and an aggregate tag τ, as follows.
  1. $\tilde{\tau} \leftarrow \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(\ell)}}, \mathsf{m}_\ell, \mathsf{STag}(\ldots \mathsf{STag}(\mathsf{k}_{\mathsf{id}_{\sigma(1)}}, \mathsf{m}_1, \tau') \ldots))$.
  2. Output 1 if $\tau = \tilde{\tau}$ and $c \notin L_c$, and output 0 otherwise.

The following theorem holds regarding SAMAC_2.

Theorem 4.3. *If G is a randomized pseudorandom generator and* PRF *is a pseudorandom function, then* SAMAC_2 *satisfies* saggUF-qCMA *security.*

Proof. Let A be a QPT adversary against SAMAC_2. In the process of STag algorithm, let $F\left(x, (z_i)_{i\in[\mu]}; (r_i)_{i\in[\mu]}\right) := G_{z_1}(\ldots G_{z_{\mu-1}}(G_{z_\mu}(x; r_\mu); r_{\mu-1}) \ldots; r_1)$ be a PRF, where x is a key, and $((z_i)_{i\in[\mu]}; (r_i)_{i\in[\mu]})$ is the input of F. If F is a QPRF, the resulting SAMAC SAMAC_2 meets saggUF-qCMA security from Theorem 4.2. To this end, we show that the function F is a QPRF if G is a randomized PRG.

First, we consider that for $i \in [\mu]$, a QPT algorithm $\mathsf{A}_{\mathsf{PRF}}$ against F is given an oracle $F_i\left((z_j, r_j)_{j\in[\mu]}\right) := G_{z_1}(\ldots G_{z_i}(P((z_j, r_j)_{j\in[i+1,\mu]}); r_i) \ldots; r_1)$, where $P : \{0,1\}^{\mu-i} \times \mathcal{R}^{\mu-i} \to \mathcal{X}$ is a random function. Notice that the case of $i = \mu$ is the same as the game that $\mathsf{A}_{\mathsf{PRF}}$ is given the truly PRF F. Let p_i be

the probability $\Pr[\mathsf{A}_{\mathsf{PRF}}^{F_i} \to 1]$ for $i \in \{0, 1, \ldots, \mu\}$ and let $\epsilon = |p_0 - p_\mu|$ denote the advantage of $\mathsf{A}_{\mathsf{PRF}}$. Then, we have $\epsilon = \left|\sum_{i\in\{0,1,\ldots,\mu-1\}}(p_i - p_{i+1})\right|$.

Next, we construct an algorithm D which distinguishes a random function $\mathsf{RF} : \{0,1\}^{\mu-1} \times \mathcal{R}^{\mu-1} \to \mathcal{X}^2$ and a function $G \circ \mathsf{RF}$ for a random function $\mathsf{RF} : \{0,1\}^{\mu-1} \times \mathcal{R}^{\mu-1} \to \mathcal{X} \times \mathcal{R}$. D breaking PRG G is as follows:

- Choose $i \in \{0, 1, \ldots, \mu - 1\}$ at random.
- Let $P^{(i)} : \{0,1\}^{\mu-i-1} \times \mathcal{R}^{\mu-i-1} \to \mathcal{X}^2$ be the oracle $P^{(i)}(z; r) = P(0^i z; 0^i r)$.
- Write $P^{(i)}$ as $(P_0^{(i)}, P_1^{(i)})$ where $P_b^{(i)} : \{0,1\}^{\mu-i-1} \times \mathcal{R}^{\mu-i-1} \to \mathcal{X}$ for each $b \in \{0,1\}$ is the left-hand side ($b = 0$) or the right-hand side ($b = 1$) of $P^{(i)}(\cdot)$.
- Construct the oracle $F((z_j)_{j\in[\mu]})$ as follows: Choose $(r_j)_{j\in[\mu]} \in \mathcal{R}^\mu$ at random, and compute
$$G_{z_1}(\ldots, G_{z_i}(P_{z_{i+1}}^{(i)}\left((z_j)_{j\in[i+2,\mu]}; (r_j)_{j\in[i+2,\mu]}\right); r_i)\ldots; r_1).$$
- When $\mathsf{A}_{\mathsf{PRF}}^F$ outputs the guessing bit, output this bit.

Let D_i be an algorithm D which chooses $i \in \{0, \ldots, \mu - 1\}$. We analyze the algorithm D_i. If the given P is a random function, then $P^{(i)}(z; r) = P(0^i z; 0^i r)$ is also truly random, and D_i simulates the environment of Game_i. If the given P is $G \circ \mathsf{RF}$, D_i simulates Game_{i+1} since P_b is $G_b \circ \mathsf{RF}$ for $b \in \{0,1\}$. For each $i \in \{0, \ldots, \mu - 1\}$, we have

$$\Pr_{P=\mathsf{RF}}\left[\mathsf{D}_i^P(1^\lambda) \to 1\right] - \Pr_{P=G\circ\mathsf{RF}}\left[\mathsf{D}_i^P(1^\lambda) \to 1\right] = p_i - p_{i+1}.$$

Then, we obtain the following advantage:

$$\begin{aligned}
\mathsf{Adv}_{G,\mathsf{D}}^{\mathrm{prg}}(\lambda) &= \left|\Pr_{P=\mathsf{RF}}\left[\mathsf{D}^P(1^\lambda) \to 1\right] - \Pr_{P=G\circ\mathsf{RF}}\left[\mathsf{D}^P(1^\lambda) \to 1\right]\right| \\
&= \frac{1}{\mu}\left|\sum_{i\in\{0,\ldots,\mu-1\}}\left(\Pr_{P=\mathsf{RF}}\left[\mathsf{D}_i^P(1^\lambda) \to 1\right] - \Pr_{P=G\circ\mathsf{RF}}\left[\mathsf{D}_i^P(1^\lambda) \to 1\right]\right)\right| \\
&= \frac{1}{\mu}\left|\sum_{i\in\{0,\ldots,\mu-1\}}(p_i - p_{i+1})\right| = \frac{\epsilon}{\mu}.
\end{aligned}$$

Therefore, $\epsilon = \mu \cdot \mathsf{Adv}_{G,\mathsf{D}}^{\mathrm{prg}}(\lambda)$ holds, and F is a QPRF.

Thus, we can replace the QPRF of a targeted tagging user by a random function, and it is possible to prove Theorem 4.3 in the same way as the proof of Theorem 4.2. Hence, from the union bound, we obtain

$$\mathsf{Adv}_{\mathsf{SAMAC}_2,\mathsf{A}}^{\text{sagguf-qcma}}(\lambda) \leq 3n(q+1)\mu \cdot \mathsf{Adv}_{G,\mathsf{D}}^{\mathrm{prg}}(\lambda) + n \cdot \mathsf{Adv}_{\mathsf{PRF},\mathsf{D}'}^{\mathrm{qr}}(\lambda) + O\left(q^3 \cdot 2^{-|\tau|/2}\right).$$

Therefore, the proof of Theorem 4.3 is completed. □

Chapter 5

Quantum-Secure Signcryption

5.1 Background of Signcryption

The notion of signcryption was introduced by Zheng [140]. In the model of signcryption, there are two kinds of setting, the *two-user setting* and *multi-user setting*. The two-user setting is a simple model of signcryption in which there is a single sender and a single receiver. In contrast, the multi-user setting is the model where there are multiple senders and receivers. It is important to realize signcryption in the multi-user setting, since it is a realistic model and the security of two-user setting does not imply that of the multi-user setting. Furthermore, there are two kinds of security for signcryption, the *insider security* and *outsider security*. In the outsider security, an external adversary only knows public information (i.e., public parameters and public-keys of entities). On the other hand, in the insider security, an internal adversary can know some private-keys. Note that the insider security is stronger, and hence it is sufficient and reasonable to consider the insider security.

The strongest security definition, which consists of strong insider confidentiality and strong insider integrity in the multi-user setting, was first formalized by Libert and Quisquater [88]. In this thesis, as IND-CCA security and sUF-CMA security (see Section 2.6.1 and 2.7.1, respectively) in this security model, we call multi-user indistinguishability against insider chosen ciphertext attack (MU-IND-iCCA security), and multi-user strong unforgeability against insider chosen message attack (MU-sUF-iCMA security), respectively. Currently, there are several constructions known for signcryption schemes satisfying the strongest security, [13, 88, 100] in the random oracle model, and [34, 100, 104, 126] in the standard model (i.e., without random oracles). The construction in [126] is a direct construction, and constructions in [34, 100, 104] are generic constructions. Note that although [100] requires a key registration, it is desirable to construct signcryption without the key registration. In addition, there is no quantum-secure signcryption scheme with MU-IND-iCCA security and MU-IND-iCMA security. Hence, of all existing ones, [34, 104]

are the most desirable constructions because we can apply (post-quantum) primitives to these ones without the key registration.

5.2 Contribution

Our purpose is to propose quantum-secure signcryption schemes with short key-size and ciphertext-size. Our schemes satisfy both MU-IND-iCCA security and MU-sUF-iCMA security. Existing constructions with these securities are generic constructions of [34, 104] in the standard model, and quantum-secure signcryption scheme can be obtained by applying lattice-based primitives to these ones. Hence, we aim at constructing quantum-secure ones with shorter key-size and ciphertext-size than the existing ones.

We present two signcryption schemes:

- One is a lattice-based construction in the standard model. More concretely, we construct a basic lattice-based construction satisfying both MU-IND-iCCA security and MU-sUF-iCMA security. Furthermore, in order to improve the efficiency of the basic construction, we construct a hybrid signcryption scheme obtained by combining the basic one and any DEM scheme with both indistinguishability against one-time attacks (IND-OT security) and one-to-one property. And then, we show that this scheme also satisfies both MU-IND-iCCA security and MU-sUF-iCMA security.

- The other is a generic construction secure in the QROM, which is constructed from any PKE with indistinguishability against chosen plaintext attacks (IND-CPA security) and any lossy identification scheme with several properties. We can obtain concrete constructions by applying concrete lattice-based constructions to these primitives because there exist the following lattice-based ones: IND-CCA secure PKE [113, 90] and lossy identification scheme [95, 84].

It is important to consider both schemes. Constructions in the standard model are stronger than ones in the QROM. That is, if a post-quantum signcryption scheme meets MU-IND-iCCA security and MU-sUF-iCMA security in the standard models, it also satisfies these securities in the QROM. Besides, generally, constructions in the (Q)ROM are more efficient than ones in the standard model in terms of key-size and ciphertext-size. In fact, standardized PKE/DS schemes fulfill security in the classical ROM. Therefore, focusing on security, we construct the signcryption scheme in the standard model while we present the one in the QROM from the viewpoint of practicality.

Furthermore, we compare our schemes with existing ones. Our schemes mean the lattice-based hybrid signcryption scheme and the scheme in the QROM which is constructed by applying suitable lattice-based constructions to our generic construction, and existing schemes mean constructions obtained

by applying lattice-based primitives to existing generic constructions [34, 104]. We show that the key-size and ciphertext-size of our schemes are shorter than those of the existing ones.

5.3 Lattice-based Signcryption

5.3.1 Basic Construction

In this section, we propose a lattice-based construction of signcryption. The idea for our construction is as follows: To achieve both of MU-IND-iCCA security and MU-sUF-iCMA security, we use a tag-based encryption (TBE) [101], a DS [101], and collision-resistant hash functions [102].

Although our construction is based on sign-then-encrypt methodology, it is shown that the construction, by combining IND-CCA secure PKE (or IND-Tag-CCA secure TBE) and sUF-CMA secure DS in a trivial way of this methodology, cannot achieve MU-sUF-iCMA security while they can meet MU-IND-iCCA security, according to [11, 100]. This is because the insider adversary, who has a receiver's public-key, can unsigncrypt a ciphertext ct by using the signcrypt oracle and obtain a valid pair of messages and signatures which passes verification of the DS. Hence, the adversary can make a forgery in the MU-sUF-iCMA game by encrypting the pair again.

To resolve the problem above, we utilize the following idea in the sign-then-encrypt paradigm: We generate a signature σ not only for a message m and a receiver's public-key pk_R, but also for outputs $(\bar{\boldsymbol{c}}_0, \bar{\boldsymbol{c}}_1)$ of tag-based trapdoor (or one-way) functions based on LWE such as $g_{\boldsymbol{A}}(\boldsymbol{s}; \boldsymbol{x}) := \boldsymbol{s}^\top \boldsymbol{A} + \boldsymbol{x}^\top \bmod q$ for a parameter $\boldsymbol{A} \in \mathbb{Z}_q^{n \times m}$, a secret vector $\boldsymbol{s} \in \mathbb{Z}_q^n$, and an error vector $\boldsymbol{x} \in \mathbb{Z}^m$. Let $\bar{\boldsymbol{c}}_0 := g_{\boldsymbol{A}}(\boldsymbol{s}; \boldsymbol{x}_0)$ and let $\bar{\boldsymbol{c}}_1 := g_{\boldsymbol{U}}(\boldsymbol{s}; \boldsymbol{x}_1)$. And, it encrypts the signature and the message by computing $\boldsymbol{c}_0 = \bar{\boldsymbol{c}}_0 + \sigma \bmod q$ and $\boldsymbol{c}_1 = \bar{\boldsymbol{c}}_1 + \mu \lfloor \frac{q}{2} \rfloor \bmod q$. Then, the adversary needs to generate false $\bar{\boldsymbol{c}}_0^*$ or $\bar{\boldsymbol{c}}_1^*$ to break $\mathsf{MU\text{-}sUF\text{-}iCMA}$ security. However, he cannot generate such a forgery unless he breaks the $\mathsf{sUF\text{-}CMA}$ security, since $\bar{\boldsymbol{c}}_0^*$ and $\bar{\boldsymbol{c}}_1^*$ are signed.

$\mathsf{LB\text{-}SCS} = (\mathsf{Setup}, \mathsf{KGen}_R, \mathsf{KGen}_S, \mathsf{SC}, \mathsf{USC})$ is constructed as follows.

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$: Given a security parameter λ, set the following parameters: Let $n \geq \lambda$ be a positive integer, and $q = \mathsf{poly}(n)$ be a prime. Let $k := \lceil \log q \rceil$, and let $m := \bar{m} + nk$, where $\bar{m} = \Theta(nk)$ is a positive integer. $\{0,1\}^\ell$ is a message-space, where ℓ is the bit-length of messages. The following lattice-based primitives are set:
  - Let $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ be the full-rank differences encoding proposed in [3].

– A gadget matrix $\boldsymbol{G}$ is defined as follows:

$$\boldsymbol{G} := \begin{bmatrix} \boldsymbol{g}^\top & & 0 \\ & \ddots & \\ 0 & & \boldsymbol{g}^\top \end{bmatrix} \in \mathbb{Z}_q^{n \times nk}, \text{ where } \boldsymbol{g}^\top = (2^0, 2^1, \ldots, 2^k).$$

– $\boldsymbol{A}_0, \ldots, \boldsymbol{A}_{nk} \xleftarrow{U} \mathbb{Z}_q^{n \times nk}$, $\boldsymbol{B} \xleftarrow{U} \mathbb{Z}_q^{n \times m}$, $\boldsymbol{U} \xleftarrow{U} \mathbb{Z}_q^{n \times \ell}$, $\boldsymbol{u}_s \xleftarrow{U} \mathbb{Z}_q^n$.

– Let $f_{\boldsymbol{A}} : \{0,1\}^* \to \mathbb{Z}_q^n$ be a lattice-based collision-resistant hash (CRH) function $f_{\boldsymbol{A}}(\boldsymbol{x}) := \boldsymbol{A}\boldsymbol{x} \bmod q$ [102] with a matrix-parameter $\boldsymbol{A}$. Notice that it is possible to evaluate $f_{\boldsymbol{A}}$ for an input with any bit-length by using the technique in [125].

Let α be a positive integer such that $\alpha^{-1} = O(nk) \cdot \omega(\sqrt{\log n})$ and let $\delta = O(nk) \cdot \omega(\sqrt{\log n})^2$ be a positive integer. Let p be a positive integer such that $p > \alpha q$, and $d = O(\sqrt{nk}) \cdot \omega(\sqrt{\log n})$ be a positive integer.

Output $\mathsf{pp} = (\lambda, n, q, k, \bar{m}, m, \ell, \boldsymbol{G}, H, \boldsymbol{A}_0, \ldots, \boldsymbol{A}_{nk}, \boldsymbol{B}, \boldsymbol{U}, \boldsymbol{u}_s, f, \alpha, \delta, p, d)$.

- $(\mathsf{pk}_R, \mathsf{sk}_R) \leftarrow \mathsf{KGen}_R(\mathsf{pp})$: Generate a receiver's key-pair as follows.
  1. $\bar{\boldsymbol{A}}_R \xleftarrow{U} \mathbb{Z}_q^{n \times \bar{m}}$, $\boldsymbol{T}_R \leftarrow \mathcal{D}^{\bar{m} \times nk}_{\mathbb{Z}, \omega(\sqrt{\log n})}$.
  2. $\boldsymbol{A}_R \leftarrow \left[\bar{\boldsymbol{A}}_R \mid -\bar{\boldsymbol{A}}_R \boldsymbol{T}_R\right] \in \mathbb{Z}_q^{n \times m}$.
  3. Output $\mathsf{pk}_R := \boldsymbol{A}_R$ and $\mathsf{sk}_R := \boldsymbol{T}_R$.
- $(\mathsf{pk}_S, \mathsf{sk}_S) \leftarrow \mathsf{KGen}_S(\mathsf{pp})$: Generate a sender's key-pair as follows.
  1. $\bar{\boldsymbol{A}}_S \xleftarrow{U} \mathbb{Z}_q^{n \times \bar{m}}$, $\boldsymbol{T}_S \leftarrow \mathcal{D}^{\bar{m} \times nk}_{\mathbb{Z}, \omega(\sqrt{\log n})}$.
  2. $\boldsymbol{A}_S \leftarrow \left[\bar{\boldsymbol{A}}_S \mid \boldsymbol{G} - \bar{\boldsymbol{A}}_S \boldsymbol{T}_S\right] \in \mathbb{Z}_q^{n \times m}$.
  3. Output $\mathsf{pk}_S := \boldsymbol{A}_S$ and $\mathsf{sk}_S := \boldsymbol{T}_S$.
- $\mathsf{ct} \leftarrow \mathsf{SC}(\mathsf{pp}, \mathsf{pk}_R, \mathsf{sk}_S, \mathsf{m})$: To signcrypt $\mathsf{m} \in \{0,1\}^\ell$, do the following:
  1. $\boldsymbol{r}_e, \boldsymbol{r}_s \leftarrow \mathcal{D}^m_{\mathbb{Z}, d}$, $\boldsymbol{t} \leftarrow f_{\bar{\boldsymbol{A}}_R}(\mathsf{pk}_S) + f_{\boldsymbol{B}}(\boldsymbol{r}_e) \in \mathbb{Z}_q^n$.
  2. $\boldsymbol{A}_{R,\boldsymbol{t}} \leftarrow \left[\bar{\boldsymbol{A}}_R \mid H(\boldsymbol{t})\boldsymbol{G} - \bar{\boldsymbol{A}}_R \boldsymbol{T}_R\right] \in \mathbb{Z}_q^{n \times m}$.
  3. $\boldsymbol{x}_0^{(0)} \leftarrow \mathcal{D}^{\bar{m}}_{\mathbb{Z}, \alpha q}$ and $\boldsymbol{x}_0^{(1)} \leftarrow \mathcal{D}^{nk}_{\mathbb{Z}, s}$, where $s^2 = (\|\boldsymbol{x}_0^{(0)}\|^2 + \bar{m}\alpha^2 q^2) \cdot \omega(\sqrt{\log n})^2$. Then $\boldsymbol{x}_0^\top \leftarrow \boldsymbol{x}_0^{(0)\top} \| \boldsymbol{x}_0^{(1)\top}$.
  4. $\boldsymbol{s} \xleftarrow{U} \mathbb{Z}_q^n$, $\boldsymbol{x}_1 \leftarrow \mathcal{D}^\ell_{\mathbb{Z}, \alpha q}$.
  5. $\bar{\boldsymbol{c}}_0 \leftarrow \boldsymbol{s}^\top \boldsymbol{A}_{R,\boldsymbol{t}} + p\boldsymbol{x}_0^\top \in \mathbb{Z}_q^m$, $\bar{\boldsymbol{c}}_1 \leftarrow \boldsymbol{s}^\top \boldsymbol{U} + p\boldsymbol{x}_1^\top \in \mathbb{Z}_q^\ell$.
  6. $\bar{c} = (\bar{\boldsymbol{c}}_0, \bar{\boldsymbol{c}}_1, \boldsymbol{r}_e)$.
  7. Generate a signature on $\mathsf{m} \| \mathsf{pk}_R \| \bar{c}$ as follows:
     – $\boldsymbol{h} \leftarrow f_{\boldsymbol{A}_S}(\mathsf{m} \| \mathsf{pk}_R \| \bar{c}) + f_{\boldsymbol{B}}(\boldsymbol{r}_s) \in \mathbb{Z}_q^n$.

- $\boldsymbol{A}_{S,\boldsymbol{h}} \leftarrow \left[\boldsymbol{A}_S \mid \boldsymbol{A}_0 + \sum_{i=1}^{nk} h_i \cdot \boldsymbol{A}_i\right] \in \mathbb{Z}_q^{n\times(m+nk)}$, where $\boldsymbol{h} = (h_1, \ldots, h_{nk})^\top \in \{0,1\}^{nk}$.
- $\boldsymbol{e} \leftarrow \mathsf{SampleD}(\boldsymbol{T}_S, \boldsymbol{A}_{S,\boldsymbol{h}}, \boldsymbol{u}_s, \delta)$.

8. $\boldsymbol{c}_0 \leftarrow \bar{\boldsymbol{c}}_0 + \boldsymbol{r}_s \in \mathbb{Z}_q^m$, $\boldsymbol{c}_1 \leftarrow \bar{\boldsymbol{c}}_1 + p \cdot \mathsf{m}\lfloor q/2 \rfloor \in \mathbb{Z}_q^\ell$.
9. Output $\mathsf{ct} = (\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{r}_e, \boldsymbol{e})$.

- $\mathsf{m}/\perp \leftarrow \mathsf{USC}(\mathsf{pp}, \mathsf{pk}_S, \mathsf{sk}_R, \mathsf{ct})$: To unsigncrypt $\mathsf{ct} = (\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{r}_e, \boldsymbol{e})$, do the following:

1. $\boldsymbol{t} \leftarrow f_{\bar{\boldsymbol{A}}_R}(\mathsf{pk}_S) + f_{\boldsymbol{B}}(\boldsymbol{r}_e) \in \mathbb{Z}_q^n$, $\boldsymbol{A}_{R,\boldsymbol{t}} \leftarrow \left[\bar{\boldsymbol{A}}_R \mid H(\boldsymbol{t})\boldsymbol{G} - \bar{\boldsymbol{A}}_R\boldsymbol{T}_R\right]$.
2. $(\boldsymbol{z}, \boldsymbol{r}_s) \leftarrow \mathsf{Invert}(\boldsymbol{T}_R, \boldsymbol{A}_{R,\boldsymbol{t}}, \boldsymbol{c}_0)$.
3. $\boldsymbol{Y} := [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\ell] \in \mathbb{Z}^{m\times\ell}$, where let $\boldsymbol{U} := [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_\ell]$ and for $i \in [\ell]$, $\boldsymbol{y}_i \leftarrow \mathsf{SampleD}(\boldsymbol{T}_R, \boldsymbol{A}_{R,\boldsymbol{t}}, \boldsymbol{u}_i, d)$.
4. $\boldsymbol{v}^\top \leftarrow p^{-1}(\boldsymbol{c}_1^\top - \boldsymbol{c}_0^\top\boldsymbol{Y}) = \boldsymbol{x}_1^\top + \mathsf{m}^\top\lfloor q/2 \rfloor - \boldsymbol{x}_0^\top\boldsymbol{Y} \in \mathbb{Z}_q^\ell$.
5. For each $i \in [\ell]$, let $\mathsf{m}_i = 0$ if v_i is closer to 0 (modulo q) than to $q/2$, and let $\mathsf{m}_i = 1$ otherwise. Then, $\mathsf{m} \leftarrow (\mathsf{m}_1, \ldots, \mathsf{m}_\ell)^\top$.
6. $\bar{\boldsymbol{c}}_0 \leftarrow \boldsymbol{c}_0 - \boldsymbol{r}_s \in \mathbb{Z}_q^m$, $\bar{\boldsymbol{c}}_1 \leftarrow \boldsymbol{c}_1 - p \cdot \mathsf{m}\lfloor q/2 \rfloor \in \mathbb{Z}_q^\ell$, $\bar{c} \leftarrow (\bar{\boldsymbol{c}}_0, \bar{\boldsymbol{c}}_1, \boldsymbol{r}_e)$.
7. $\boldsymbol{h} \leftarrow f_{\bar{\boldsymbol{A}}_S}(\mathsf{m}\|\mathsf{pk}_R\|\bar{c}) + f_{\boldsymbol{B}}(\boldsymbol{r}_s) \in \mathbb{Z}_q^n$, $\boldsymbol{A}_{S,\boldsymbol{h}} \leftarrow \left[\boldsymbol{A}_S \mid \boldsymbol{A}_0 + \sum_{i=1}^{nk} h_i \cdot \boldsymbol{A}_i\right]$.
8. Output m if $\boldsymbol{A}_{S,\boldsymbol{h}} \cdot \boldsymbol{e} = \boldsymbol{u}_s \bmod q \wedge \|\boldsymbol{e}\| \leq \delta\sqrt{m+nk}$. Output $\perp$ otherwise.

As the security of LB-SCS, Theorem 5.1 and 5.2 hold.

Theorem 5.1. *If* $\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}$ *assumption holds for* $\alpha \geq 2\sqrt{n}/q$*, and* CRH f *meets collision-resistance, then* LB-SCS *meets* MU-IND-iCCA *security.*

Theorem 5.2. *If* $\mathsf{SIS}_{n,q,\beta,m+nk}$ *assumption holds for* $\beta = O((nk)^{5/2}) \cdot \omega(\sqrt{\log n})^3$*, and* CRH f *meets collision-resistance, then* LB-SCS *meets* MU-sUF-iCMA *security.*

Proof of Theorem 5.1

Let A be a PPT algorithm against LB-SCS and let q_u be the number of queries issued to $\mathsf{USC}^{\mathsf{O}}$ oracle. For values $\boldsymbol{x}$ generated in Challenge phase, we write $\boldsymbol{x}^*$. For $i \in \{0,1,2,3,4\}$, we consider the following security games.

- Game_0: This game is the ordinary MU-IND-iCCA game.
- Game_1: This game is the same as Game_0 except that when A submits an unsigncrypt query $(\mathsf{pk}_S, (\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{r}_e, \boldsymbol{e}))$ such that $\boldsymbol{t}^* = f_{\bar{\boldsymbol{A}}_R}(\mathsf{pk}_S) + f_{\boldsymbol{B}}(\boldsymbol{r}_e)$, $\mathsf{USC}^{\mathsf{O}}$ oracle returns $\perp$.

- Game_2: This game is the same as Game_1 except for replacing the parameter $\boldsymbol{B}$ of a collision resistant hash function $f_{\boldsymbol{B}}$ by a parameter $\boldsymbol{B}$ with a lattice-trapdoor $\boldsymbol{T}_{\boldsymbol{B}}$.
- Game_3: This game is the same as Game_2 except that a component $\boldsymbol{c}_0^*$ of a challenge ciphertext is generated as follows:
  1. $\boldsymbol{s} \xleftarrow{U} \mathbb{Z}_q^n$, $\bar{\boldsymbol{x}}_0 \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q}^{\bar{m}}$, $\boldsymbol{x}_0' \leftarrow \mathcal{D}_{\mathbb{Z},\alpha q\sqrt{\bar{m}}\cdot\omega(\sqrt{\log n})}^{nk}$.
  2. $\hat{\boldsymbol{c}}_0 \leftarrow \boldsymbol{s}^\top \bar{\boldsymbol{A}}_R + p\bar{\boldsymbol{x}}_0^\top \in \mathbb{Z}_q^{\bar{m}}$.
  3. $\boldsymbol{c}_0' \leftarrow -p\hat{\boldsymbol{c}}_0^\top \boldsymbol{T}_R + p\boldsymbol{x}_0'^\top = p\boldsymbol{s}^\top(-\bar{\boldsymbol{A}}_R\boldsymbol{T}_R) + p(-\bar{\boldsymbol{x}}_0^\top \boldsymbol{T}_R + \boldsymbol{x}_0'^\top) \in \mathbb{Z}_q^{nk}$.
  4. $\bar{\boldsymbol{c}}_0^{*\top} := \hat{\boldsymbol{c}}_0^\top \| \boldsymbol{c}_0'^\top \in \mathbb{Z}_q^m$.
- Game_4: This game is the same as Game_3 except that $(\hat{\boldsymbol{c}}_0, \bar{\boldsymbol{c}}_1^*) \in \mathbb{Z}_q^{\bar{m}} \times \mathbb{Z}_q^\ell$ are chosen uniformly at random.

And then, we define the following events for $i \in \{0,1,2,3,4\}$:

- W_i: This is the event that A outputs $b' \in \{0,1\}$ such that $b = b'$ in Game_i.
- CR_i: This is the event that A submits an unsigncrypt query $(\mathsf{pk}_S, (\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{r}_e, \boldsymbol{e}))$ such that $(\mathsf{pk}_S, \boldsymbol{r}_e) \neq (\mathsf{pk}_S^*, \boldsymbol{r}_e^*) \wedge \boldsymbol{t}^* = f_{\bar{\boldsymbol{A}}_R}(\mathsf{pk}_S) + f_{\boldsymbol{B}}(\boldsymbol{r}_e) \in \mathbb{Z}_q^n$ in Game_i.
- F_i: This is the event that A submits an unsigncrypt query $(\mathsf{pk}_S, (\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{e}, \boldsymbol{r}_e))$ such that $(\mathsf{pk}_S, \boldsymbol{r}_e) = (\mathsf{pk}_S^*, \boldsymbol{r}_e^*) \wedge \|\boldsymbol{e}\| \leq \delta\sqrt{m+nk} \wedge \boldsymbol{A}_{S,\boldsymbol{h}} = \boldsymbol{u}_s \bmod q$, where $\boldsymbol{h} = f_{\bar{\boldsymbol{A}}_S}(\mathsf{m}\|\mathsf{pk}_R\|\bar{c}) + f_{\boldsymbol{B}}(\boldsymbol{r}_e) \in \mathbb{Z}_q^n$ in Game_i.

Then, we have

$$\begin{aligned}\mathsf{Adv}^{\text{mu-ind-icca}}_{\mathsf{LB\text{-}SCS},\mathsf{A}}(\lambda) &= \left|\Pr[W_0] - \frac{1}{2}\right| \\ &\leq |\Pr[W_0] - \Pr[W_1]| + |\Pr[W_1] - \Pr[W_2]| \\ &\quad + |\Pr[W_2] - \Pr[W_3]| + \left|\Pr[W_3] - \frac{1}{2}\right|.\end{aligned}$$

In addition, let *Bad* be the event that A outputs an unsigncrypt query $(\mathsf{pk}_S^*, (\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{r}_e^*, \boldsymbol{e}))$ such that $\mathsf{USC}^{\mathsf{O}}$ oracle returns $\mathsf{m} \neq \perp$. Then, because $\Pr[W_1 \mid \mathit{Bad}] = \Pr[CR_1] + \Pr[F_1]$ holds, we get

$$\begin{aligned}|\Pr[W_0] - \Pr[W_1]| &\leq \Pr[W_1 \mid \mathit{Bad}] \\ &= \Pr[CR_1] + \Pr[F_1] \\ &\leq \Pr[CR_1] + |\Pr[F_1] - \Pr[F_2]| + |\Pr[F_2] - \Pr[F_3]| \\ &\quad + |\Pr[F_3] - \Pr[F_4]| + \Pr[F_4].\end{aligned}$$

Proof of $\Pr[CR_1] \leq \mathsf{Adv}^{\text{cr}}_{\mathsf{CRH},\mathsf{B}}(\lambda)$: We construct a PPT algorithm C breaking the collision-resistance of $f_{\bar{\boldsymbol{A}}_R} + f_{\boldsymbol{B}}$ as follows:

Setup: Take as input $(\boldsymbol{B}, \bar{\boldsymbol{A}}_R) \in \mathbb{Z}_q^{n\times(m+\bar{m})}$. Sample $\boldsymbol{T}_R \leftarrow \mathcal{D}^{\bar{m}\times nk}_{\sqrt{\log n}}$ and compute $\boldsymbol{A}_R \leftarrow \left[\bar{\boldsymbol{A}}_R | - \bar{\boldsymbol{A}}_R\boldsymbol{T}_R\right] \in \mathbb{Z}_q^{n\times m}$. $\boldsymbol{B}$ is added to a public parameter. Send $(\mathsf{pp}, \mathsf{pk}_R)$ to A.

Queries 1: Given an unsigncrypt query $(\mathsf{pk}_S, \mathsf{ct})$, simulate $\mathsf{USC}^{\mathsf{O}}$ oracle by using $\mathsf{sk}_R = \boldsymbol{T}_R$.

Challenge: When A submits $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{pk}_S^*, \mathsf{sk}_S^*)$, compute $(\boldsymbol{c}_0^*, \boldsymbol{c}_1^*, \boldsymbol{r}_e^*, \boldsymbol{e}^*) \leftarrow \mathsf{SC}(\mathsf{pp}, \mathsf{sk}_R, \mathsf{sk}_S^*, \mathsf{m}_b)$ and return $\mathsf{ct}^* := (\boldsymbol{c}_0^*, \boldsymbol{c}_1^*, \boldsymbol{r}_e^*, \boldsymbol{e}^*)$, where $b \xleftarrow{U} \{0,1\}$.

Queries 2: Given an unsigncrypt query $(\mathsf{pk}_S, \mathsf{ct})$, simulate $\mathsf{USC}^{\mathsf{O}}$ oracle in the same way as the process of Queries 1 phase.

Output: When A outputs the guessing bit $b' \in \{0,1\}$, find a submitted query $(\mathsf{pk}_S, (\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{r}_e, \boldsymbol{e}))$ such that $\boldsymbol{t}^* = f_{\bar{\boldsymbol{A}}_R}(\mathsf{pk}_S) + f_{\boldsymbol{B}}(\boldsymbol{r}_e) \in \mathbb{Z}_q^n$ and $(\mathsf{pk}_S, \boldsymbol{r}_e) \neq (\mathsf{pk}_S^*, \boldsymbol{r}_e^*)$. Then, output $((\mathsf{pk}_S, \boldsymbol{r}_e), (\mathsf{pk}_S^*, \boldsymbol{r}_e^*))$.

C simulates the view of A completely, and the output is a collision of $f_{\bar{\boldsymbol{A}}_R} + f_{\boldsymbol{B}}$ clearly. Thus, we have the inequality. ■

Proof of $\Pr[W_1] = \Pr[W_2]$ and $\Pr[F_1] = \Pr[F_2]$: We show $\Pr[W_1] = \Pr[W_2]$. The statistical distance between distributions of $\boldsymbol{B}$ in the two games is negligible from the proof in Section 5.2 in [101]. Hence, we have the inequality. In the same way as this, we get $\Pr[F_1] = \Pr[F_2]$. ■

Proof of $|\Pr[W_2] - \Pr[W_3]| \leq \mathsf{negl}(\lambda)$ and $|\Pr[F_2] - \Pr[F_3]| \leq \mathsf{negl}(\lambda)$: The difference between Game_2 and Game_3 is as follows: In Game_3, the error term of $\boldsymbol{c}_0'$ is the form $\bar{\boldsymbol{x}}_0^\top \boldsymbol{t}_i + \boldsymbol{x}_0'^\top$ for $i \in [nk]$ while in Game_2, the error term is drawn from $\mathcal{D}_{\mathbb{Z},s}$, where $\boldsymbol{T}_R = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{nk}]$, $s^2 = (\|\boldsymbol{x}_0^{(0)}\|^2 + \bar{m}(\alpha q)^2) \cdot \omega(\sqrt{\log n})^2$, and $\boldsymbol{x}_0^{(0)}, \bar{\boldsymbol{x}}_0$ are drawn from $\mathcal{D}^{\bar{m}}_{\mathbb{Z},\alpha q}$. Because $\boldsymbol{t}_i$ is an independent discrete Gaussian over $\Lambda^\perp(\bar{\boldsymbol{A}}_R)$, the statistical distance between the error terms in the two games is negligible by applying Corollary 3.10 in [113]. Hence, we have $|\Pr[W_2] - \Pr[W_3]| \leq \mathsf{negl}(\lambda)$ and $|\Pr[F_2] - \Pr[F_3]| \leq \mathsf{negl}(\lambda)$. ■

Proof of $|\Pr[W_3] - \Pr[W_4]| \leq \mathsf{Adv}_{\mathsf{D}}^{\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}}(\lambda)$ and $|\Pr[F_3] - \Pr[F_4]| \leq \mathsf{Adv}_{\mathsf{D}'}^{\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}}(\lambda)$: First, we prove $|\Pr[W_3] - \Pr[W_4]| \leq \mathsf{Adv}_{\mathsf{D}}^{\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}}(\lambda)$ by constructing a PPT algorithm D solving $\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}$ problem in the following way:

Setup: Given samples $(\bar{\boldsymbol{A}}_R, \boldsymbol{U}, \hat{\boldsymbol{c}}_0, \hat{\boldsymbol{c}}_1) \in \mathbb{Z}_q^{n\times(\bar{m}+\ell)} \times \mathbb{Z}_q^{\bar{m}+\ell}$ from the LWE oracle, compute $\boldsymbol{A}_R \leftarrow \left[\bar{\boldsymbol{A}}_R | - H(\boldsymbol{t}^*)\boldsymbol{G} - \bar{\boldsymbol{A}}_R\boldsymbol{T}_R\right] \in \mathbb{Z}_q^{n\times m}$. $(\boldsymbol{B}, \boldsymbol{U})$ are components of pp. Send pp and $\mathsf{pk}_R := \boldsymbol{A}_R$ to A.

Queries 1: Given an unsigncrypt query $(\mathsf{pk}_S, \mathsf{ct})$, simulate $\mathsf{USC}^{\mathsf{O}}$ oracle by using $\mathsf{sk}_R = \boldsymbol{T}_R$.

Challenge: When A submits $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{pk}_S^*, \mathsf{sk}_S^*)$, do the following:

1. $b \xleftarrow{U} \{0,1\}$.
2. $\boldsymbol{r}_e^* \leftarrow \mathsf{SampleD}(\boldsymbol{T}_{\boldsymbol{B}}, \boldsymbol{B}, (\boldsymbol{t}^* - f_{\bar{\boldsymbol{A}}_R}(\mathsf{pk}_S^*)) \mod q, d)$.
3. $\bar{\boldsymbol{c}}_0^{*\top} \leftarrow p(\hat{\boldsymbol{c}}_0^\top \| \hat{\boldsymbol{c}}_0^\top \boldsymbol{T}_R + \boldsymbol{x}_0'^\top)$, where $\boldsymbol{x}_0' \leftarrow \mathcal{D}^{nk}_{\mathbb{Z},\alpha q\sqrt{\bar{m}}\cdot\omega(\sqrt{\log n})}$.
4. $\bar{\boldsymbol{c}}_1^* \leftarrow p\hat{\boldsymbol{c}}_1 \in \mathbb{Z}_q^\ell$.
5. Compute $(\boldsymbol{c}_0^*, \boldsymbol{c}_1^*, \boldsymbol{r}_e^*, \boldsymbol{e}^*)$ in the same way as SC algorithm with $(\boldsymbol{r}_e^*, \bar{\boldsymbol{c}}_0^*, \bar{\boldsymbol{c}}_1^*)$.
6. Return $\mathsf{ct}^* := (\boldsymbol{c}_0^*, \boldsymbol{c}_1^*, \boldsymbol{r}_e^*, \boldsymbol{e}^*)$.

Queries 2: In the same way as the process of Queries 1 phase, simulate $\mathsf{USC}^{\mathsf{O}}$ oracle.

Output: When A outputs $b' \in \{0,1\}$, output 1 if $b = b'$, and output 0 otherwise.

In Challenge phase, D simulates the view of A in Game_3 (resp. Game_4) if it is given LWE samples (resp. uniformly random samples). In addition, in Queries 1 and Queries 2 phases, D can simulate $\mathsf{USC}^{\mathsf{O}}$ oracle since it has a secret key $\boldsymbol{T}_R$, and $H(\boldsymbol{t}) - H(\boldsymbol{t}^*) = H(\boldsymbol{t} - \boldsymbol{t}^*) \in \mathbb{Z}_q^{n\times n}$ is invertible for any $\boldsymbol{t} = f_{\bar{\boldsymbol{A}}_R}(\mathsf{pk}_S) + f_{\boldsymbol{B}}(\boldsymbol{r}_e) \in \mathbb{Z}_q^n$. Hence, we have $|\Pr[W_3] - \Pr[W_4]| \leq \mathsf{Adv}_{\mathsf{D}}^{\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}}(\lambda)$.

In addition, we show $|\Pr[F_3] - \Pr[F_4]| \leq \mathsf{Adv}_{\mathsf{D}'}^{\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}}(\lambda)$ by constructing a PPT algorithm D' solving $\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}$. D' is the same as D except that in Output phase, D' checks whether A issues an unsigncrypt query $(\mathsf{pk}_S^*, (\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{r}_e, \boldsymbol{e}))$ such that $(\mathsf{pk}_S, \boldsymbol{r}_e) = (\mathsf{pk}_S^*, \boldsymbol{r}_e^*) \ \wedge \ \|\boldsymbol{e}\| \leq \delta\sqrt{m+nk} \ \wedge \ \boldsymbol{A}_{S,\boldsymbol{h}} \cdot \boldsymbol{e} = \boldsymbol{u}_s \bmod q$, where $\boldsymbol{h} = f_{\bar{\boldsymbol{A}}_S}(\mathsf{m}\|\mathsf{pk}_R\|\bar{c}) + f_{\boldsymbol{B}}(\boldsymbol{r}_e) \in \mathbb{Z}_q^n$. If so, it outputs 1. Otherwise, it outputs 0. Then, we have the inequality. ■

Proof of $\Pr[W_4] = 1/2$ and $\Pr[F_4] = \mathsf{negl}(\lambda)$: We have $\Pr[W_4] = 1/2$ since ct^* and uniformly random $b \in \{0,1\}$ are independent each other. Next, we show $\Pr[F_4] = \mathsf{negl}(\lambda)$. In order to submit an unsignrypt query $(\mathsf{pk}_S^*, (\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{r}_e, \boldsymbol{e}))$ such that $(\mathsf{pk}_S, \boldsymbol{r}_e) = (\mathsf{pk}_S^*, \boldsymbol{r}_e^*) \ \wedge \ \|\boldsymbol{e}\| \leq \delta\sqrt{m+nk} \ \wedge \ \boldsymbol{A}_{S,\boldsymbol{h}} \cdot \boldsymbol{e} = \boldsymbol{u}_s \bmod q$, A needs to know $\boldsymbol{r}_s^*$. Although a component $\boldsymbol{c}_0^*$ of a challenge ciphertext and $\boldsymbol{h}^*$ hide $\boldsymbol{r}_s^*$, these values are uniformly random in Game_4. Thus, A cannot find $\boldsymbol{r}_s^*$, and we have the equation. ■

From the discussion above, we obtain

$$\mathsf{Adv}^{\text{mu-ind-icca}}_{\mathsf{LB\text{-}SCS},\mathsf{A}}(\lambda) \leq 2 \cdot \mathsf{Adv}_{\mathsf{D}}^{\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}}(\lambda) + \mathsf{Adv}^{\text{cr}}_{\mathsf{CRH},\mathsf{B}}(\lambda) + \mathsf{negl}(\lambda).$$

The proof is completed. □

Proof of Theorem 5.2

Let A be a PPT adversary against LB-SCS and let q_s be the number of queries submitted to SC^{O} oracle. Let $M := \mathsf{m}\|\mathsf{pk}_R\|\bar{c}$ and let $x^{(i)}$ for $i \in [q_s]$ be a value x generated in the i-th oracle access.

The adversary A is classified into several types in the following way:

Type-1. A generates a forgery by finding a collision of $f_{\bar{\boldsymbol{A}}_S} + f_{\boldsymbol{B}}$.

Type-2. A generates a forgery without finding a collision of $f_{\bar{\boldsymbol{A}}_S} + f_{\boldsymbol{B}}$.

Type-2-(a). A generates a forgery without any unsigncrypt queries.

Type-2-(b). A generates a forgery by using an unsigncrypt query.

First, we consider a Type-1 adversary. We construct a PPT algorithm F_{cr} breaking the collision-resistance of $f_{\bar{\boldsymbol{A}}_S} + f_{\boldsymbol{B}}$ in the following way:

Setup: Take as input parameters $(\bar{\boldsymbol{A}}_S, \boldsymbol{B}) \in \mathbb{Z}_q^{n\times(\bar{m}+m)}$ of f. In the same way as Setup and KGen_S algorithms, generate pp and $(\mathsf{pk}_S, \mathsf{sk}_S)$. Then, send pk_S to A.

Queries: Given a signcrypt query $(\mathsf{pk}_R, \mathsf{m})$, return $\mathsf{ct} \leftarrow \mathsf{SC}(\mathsf{pp}, \mathsf{pk}_R, \mathsf{sk}_S, \mathsf{m})$.

Output: When A outputs $(\mathsf{pk}_R, \mathsf{sk}_R^*, \mathsf{ct}^*)$, find a pair $(M^{(i)}, \boldsymbol{r}_s^{(i)})$ such that $f_{\bar{\boldsymbol{A}}_S}(M^*) + f_{\boldsymbol{B}}(\boldsymbol{r}_s^*) = f_{\bar{\boldsymbol{A}}_S}(M^{(i)}) + f_{\boldsymbol{B}}(\boldsymbol{r}_s^{(i)})$ and $(M^{(i)}, \boldsymbol{r}_s^{(i)}) \neq (M^*, \boldsymbol{r}_s^*)$.

F_{cr} simulates the view of A completely since $(\mathsf{pp}, \mathsf{pk}_S)$ in the game above is the same as those in the ordinary MU-sUF-iCMA game, and it can simulate SC^{O} oracle by using the generated key sk_S. The output of F_{cr} is clearly the collision of $f_{\bar{\boldsymbol{A}}_S} + f_{\boldsymbol{B}}$. Hence, the success probability of the Type-1 adversary is negligible if $f_{\bar{\boldsymbol{A}}_S} + f_{\boldsymbol{B}}$ meets collision-resistance.

Next, we consider the Type-2-(a) adversary. We construct a PPT algorithm $\mathsf{S}_{\mathrm{new}}$ solving the SIS problem, in the following way.

Setup: Given a SIS challenge $(\bar{\boldsymbol{A}}_S \| \boldsymbol{A}', \boldsymbol{u}) \in \mathbb{Z}_q^{n\times(\bar{m}+nk)} \times \mathbb{Z}_q^n$, do the following:

- Choose q_s values $\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \ldots, \boldsymbol{h}^{(q_s)} \in \{0,1\}^{nk}$ uniformly at random. Compute the set P of all strings $\mathsf{p} \in \{0,1\}^{\leq nk}$ with the property that p is a shortest string for which no $\boldsymbol{h}^{(i)}$ has p as a prefix. In the same way as this, P represents the set of maximal subtrees of $\{0,1\}^{\leq nk}$ that do not contain any of $\boldsymbol{h}^{(i)}$. Notice that the size of P is at most $(nk-1)q_s+1$. Choose some $\mathsf{p} \in P$ uniformly at random and let $t = |\mathsf{p}| \leq nk$.
- $(\boldsymbol{A}_S = \bar{\boldsymbol{A}}_S \| \boldsymbol{A}', \boldsymbol{A}_0, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_{nk}, \boldsymbol{u}_s = \boldsymbol{u})$ is constructed as follows: For $i \in \{0, 1, \ldots, nk\}$, choose $\boldsymbol{T}_{S,i} \leftarrow \mathcal{D}^{\bar{m}\times nk}_{\mathbb{Z},\omega(\sqrt{\log n})}$ and let

$$\boldsymbol{A}_i = \boldsymbol{H}_i\boldsymbol{G} - \bar{\boldsymbol{A}}_S\boldsymbol{T}_{S,i}, \text{ where } \boldsymbol{H}_i = \begin{cases} H(0) = \boldsymbol{0} & i > t \\ (-1)^{\mathsf{p}_i} \cdot H(u_i) & i \in [t] \\ -\sum_{j\in[t]} \mathsf{p}_j \cdot \boldsymbol{H}_j & i = 0 \end{cases}$$

$u_1, \ldots, u_{nk} \in \mathbb{Z}_q^n$ are units whose nontrivial subset-sums are also units, and $H : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ is the FRD encoding of [3].

– Generate $(\mathsf{pp}, \mathsf{pk}_S)$ following Setup and KGen_S algorithms except for the parameters above.

$\mathsf{Queries}$: Given the j-th signcrypt query $(\mathsf{pk}_R^{(j)}, \mathsf{m}^{(j)})$ $(j \in [q_s])$, do the following:

1. Compute $\bar{c}^{(j)}$ following SC algorithm and let $M^{(j)} := \mathsf{m}^{(j)} \| \mathsf{pk}_R^{(j)} \| \bar{c}^{(j)}$.
2. $\boldsymbol{r}_s^{(j)} \leftarrow \mathsf{SampleD}(\boldsymbol{T}_{\boldsymbol{B}}, \boldsymbol{B}, (\boldsymbol{h}^{(j)} - f_{\bar{\boldsymbol{A}}_S}(M^{(j)})) \bmod q, d)$.
3. $\boldsymbol{A}_{S,\boldsymbol{h}^{(j)}} \leftarrow \left[\bar{\boldsymbol{A}}_S | \boldsymbol{A}' | \boldsymbol{H}\boldsymbol{G} - \bar{\boldsymbol{A}}_S(\boldsymbol{T}_{S,0} + \sum_{i\in[nk]} h_i^{(j)} \cdot \boldsymbol{T}_{S,i})\right] \in \mathbb{Z}_q^{n\times(m+nk)}$, where $\boldsymbol{T}_{\boldsymbol{h}} = \boldsymbol{T}_{S,0} + \sum_{i\in[nk]} h_i^{(j)} \cdot \boldsymbol{T}_{S,i}$ is a trapdoor of $\boldsymbol{A}_{S,\boldsymbol{h}^{(j)}}$ and $\boldsymbol{H}$ is invertible because the prefix of $\boldsymbol{h}^{(j)}$ is not p.
4. $\boldsymbol{e}^{(j)} \leftarrow \mathsf{SampleD}(\boldsymbol{T}_{\boldsymbol{h}^{(j)}}, \boldsymbol{A}_{S,\boldsymbol{h}^{(j)}}, \boldsymbol{u}_s, \delta)$.
5. Compute $\mathsf{ct}^{(j)}$ by using $(\boldsymbol{e}^{(j)}, \boldsymbol{r}_s^{(j)})$ and return $\mathsf{ct}^{(j)}$.

Output: When A outputs a forgery $(\mathsf{pk}_R^*, \mathsf{sk}_R^*, \mathsf{ct}^* = (\boldsymbol{c}_0^*, \boldsymbol{c}_1^*, \boldsymbol{r}_e^*, \boldsymbol{e}^*))$, abort if $\mathsf{USC}(\mathsf{pp}, \mathsf{pk}_S, \mathsf{sk}_R^*, \mathsf{ct}^*) = \perp$. Compute $\boldsymbol{h}^* \leftarrow f_{\bar{\boldsymbol{A}}_S}(M^*) + f_{\boldsymbol{B}}(\boldsymbol{r}_s^*)$ if $\mathsf{USC}(\mathsf{pp}, \mathsf{pk}_S, \mathsf{sk}_R^*, \mathsf{ct}^*) \neq \perp$. Then, extract the solution of $\mathsf{SIS}_{n,q,\beta,m+nk}$ as follows: Compute $\boldsymbol{z} \in \mathbb{Z}^{m+nk}$ such that

$$\underbrace{\left[\bar{\boldsymbol{A}}_S | \boldsymbol{A}'\right]}_{\boldsymbol{A}_S} \underbrace{\begin{bmatrix} \boldsymbol{I}_m & \\ & \boldsymbol{I}_{nk} \end{bmatrix} \begin{bmatrix} & -\boldsymbol{T}_S^* \\ & \end{bmatrix} \boldsymbol{e}^*}_{\boldsymbol{z}} = \boldsymbol{u}_s \bmod q,$$

where $\boldsymbol{T}_S = \boldsymbol{T}_{S,0} + \sum_{i\in[nk]} h_i^* \cdot \boldsymbol{T}_{S,i}$. Then, $(\boldsymbol{z}^\top, 0)$ is the solution of the given SIS instance $\left[\bar{\boldsymbol{A}}_S | \boldsymbol{A}' | \boldsymbol{u}\right]$.

We analyze $\mathsf{S}_{\mathrm{new}}$. For each signcrypt query, we have

$$\boldsymbol{A}_{S,\boldsymbol{h}} = [\boldsymbol{A}_S | \boldsymbol{A}_0 + \sum_{i\in[nk]} h_i \cdot \boldsymbol{A}_i] = [\bar{\boldsymbol{A}}_S | \boldsymbol{A}' | \boldsymbol{H}\boldsymbol{G} - \bar{\boldsymbol{A}}_S(\boldsymbol{T}_{S,0} + \sum_{i\in[nk]} h_i \cdot \boldsymbol{T}_{S,i})].$$

$\boldsymbol{T}_S = \boldsymbol{T}_{S,0} + \sum_{i\in[nk]} h_i \cdot \boldsymbol{T}_{S,i}$ is a trapdoor for $\boldsymbol{A}_{S,\boldsymbol{h}}$. Besides, by Lemma 2.9 in [101], we have

$$s_1(\boldsymbol{T}_S) = \sqrt{nk+1} \cdot O(\sqrt{m} + \sqrt{nk}) \cdot \omega(\sqrt{\log n}) = O(nk) \cdot \omega(\sqrt{\log n}).$$

with overwhelming probability. We can generate $\boldsymbol{e} \leftarrow \mathcal{D}_{\Lambda_{\boldsymbol{u}_s}^{\perp}(\boldsymbol{A}_{S,\boldsymbol{h}}),\delta}$ properly since δ is large enough. Thus, $\mathsf{S}_{\mathrm{new}}$ simulates SC^{O} oracle completely.

Then, because $\|\boldsymbol{e}^*\| \leq \delta\sqrt{m+nk} = O((nk)^{3/2}) \cdot \omega(\sqrt{\log n})^2$ and $s_1(\boldsymbol{T}_S^*) = O(nk) \cdot \omega(\sqrt{\log n})$ with overwhelming probability, we have $\|\boldsymbol{z}\| = O((nk)^{5/2}) \cdot \omega(\sqrt{\log n})^3$. That is, $\|\boldsymbol{z}\|$ is at most $\beta - 1$. Hence, the output of $\mathsf{S}_{\mathrm{new}}$ is a valid solution of SIS problem. The success probability of A is at most $((nk-1)q_s + 1) \cdot \mathsf{Adv}_{\mathsf{S}_{\mathrm{new}}}^{\mathsf{SIS}_{n,q,\beta,m+nk}}(\lambda) + \mathsf{negl}(\lambda)$.

Finally, we consider a $\mathsf{Type\text{-}2\text{-}}(b)$ adversary. Namely, this adversary outputs a forgery on $M^{(j)} \| \boldsymbol{r}_s^{(j)}$ which have been used in the j-th oracle access. We construct a PPT algorithm $\mathsf{S}_{\mathrm{pre}}$ finding a solution of $\mathsf{SIS}_{n,q,\beta,m+nk}$ as follows:

Setup: Given an SIS challenge $\bar{\boldsymbol{A}}_S \| \boldsymbol{A}' \in \mathbb{Z}_q^{n \times (\bar{m}+nk)}$, do the following:

- Choose q_s values $\boldsymbol{h}^{(1)}, \boldsymbol{h}^{(2)}, \ldots, \boldsymbol{h}^{(q_s)} \in \{0,1\}^{nk}$ uniformly at random. Choose $\boldsymbol{h}^* \xleftarrow{U} \{\boldsymbol{h}^{(i)}\}_{i \in [n]}$.
- $(\boldsymbol{A}_S, \boldsymbol{A}_0, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_{nk})$ is constructed in the same way as the above algorithm $\mathsf{S}_{\mathrm{new}}$.
- Generate $(\mathsf{pp}, \mathsf{pk}_S)$ following Setup and KGen_S algorithms except for the parameters above.

$\mathsf{Queries}$: Given the j-th signcrypt query $(\mathsf{pk}_R^{(j)}, \mathsf{m}^{(j)})$, do the following:

1. Compute $\bar{c}^{(j)}$ following SC algorithm and let $M^{(j)} := \mathsf{m}^{(j)} \| \mathsf{pk}_R^{(j)} \| \bar{c}^{(j)}$.
2. $\boldsymbol{r}_s^{(j)} \leftarrow \mathsf{SampleD}(\boldsymbol{T}_{\boldsymbol{B}}, \boldsymbol{B}, (\boldsymbol{h}^{(j)} - f_{\bar{\boldsymbol{A}}_S}(M)) \bmod q, d)$.
3. $\boldsymbol{A}_{S,\boldsymbol{h}^{(j)}} \leftarrow \left[\bar{\boldsymbol{A}}_S | \boldsymbol{A}' | \boldsymbol{H}\boldsymbol{G} - \bar{\boldsymbol{A}}_S (\boldsymbol{T}_{S,0} + \sum_{i \in [nk]} h_i^{(j)} \cdot \boldsymbol{T}_{S,i})\right] \in \mathbb{Z}_q^{n \times (m+nk)}$, where $\boldsymbol{T}_{\boldsymbol{h}^{(j)}} = \boldsymbol{T}_{S,0} + \sum_{i \in [nk]} h_i^{(j)} \cdot \boldsymbol{T}_{S,i}$ is a trapdoor of $\boldsymbol{A}_{S,\boldsymbol{h}^{(j)}}$.
4. $\boldsymbol{e}^{(j)} \leftarrow \mathsf{SampleD}(\boldsymbol{T}_{\boldsymbol{h}}, \boldsymbol{A}_{S,\boldsymbol{h}^{(j)}}, \boldsymbol{u}_s, \delta)$.
5. Compute $\mathsf{ct}^{(j)}$ by using $(\boldsymbol{e}^{(j)}, \boldsymbol{r}_s^{(j)})$ and return $\mathsf{ct}^{(j)}$.

Output: When A outputs a forgery $(\mathsf{pk}_R^*, \mathsf{sk}_R^*, \mathsf{ct}^* = (\boldsymbol{c}_0^*, \boldsymbol{c}_1^*, \boldsymbol{r}_e^*, \boldsymbol{e}^*))$, abort if $\mathsf{USC}(\mathsf{pp}, \mathsf{pk}_S, \mathsf{sk}_R^*, \mathsf{ct}^*) = \perp$ or $\boldsymbol{h}^* \neq f_{\bar{\boldsymbol{A}}_S}(M^*) + f_{\boldsymbol{B}}(\boldsymbol{r}_s^*)$. Otherwise, extract the solution of $\mathsf{SIS}_{n,q,\beta,m+nk}$ as follows: Compute $\boldsymbol{z} \in \mathbb{Z}^{m+nk}$ such that

$$\underbrace{\left[\bar{\boldsymbol{A}}_S | \boldsymbol{A}'\right]}_{\boldsymbol{A}_S} \underbrace{\begin{bmatrix} \boldsymbol{I}_m & & -\boldsymbol{T}_S^* \\ & \boldsymbol{I}_{nk} & \end{bmatrix} (\boldsymbol{e}^* - \boldsymbol{e}^{(i)})}_{\boldsymbol{z}} = \boldsymbol{0} \bmod q,$$

where $\boldsymbol{T}_S = \boldsymbol{T}_{S,0} + \sum_{i \in [nk]} h_i \cdot \boldsymbol{T}_{S,i}$. Then, output $\boldsymbol{z} \in \mathbb{Z}^{m+nk}$.

Because $\|\boldsymbol{e}^*\|, \|\boldsymbol{e}^{(i)}\| \leq \delta \cdot \sqrt{m+nk}$ and $s_1(\boldsymbol{T}_S^*) = O(nk) \cdot \omega(\sqrt{\log n})$ with overwhelming probability, we have $\|\boldsymbol{z}\| = O((nk)^{5/2}) \cdot \omega(\sqrt{\log n})^3$. Thus, the output of $\mathsf{S}_{\mathrm{pre}}$ is a solution to $\mathsf{SIS}_{n,q,\beta,m+nk}$.

From the discussion above, we obtain

$$\begin{aligned} \mathsf{Adv}^{\text{mu-suf-icma}}_{\mathsf{LB\text{-}SCS},\mathsf{A}}(\lambda) \leq 2\left((nk-1)q_s + 1\right) \cdot & \mathsf{Adv}^{\mathsf{SIS}_{n,q,\beta,m+nk}}_{\mathsf{S}}(\lambda) \\ & + \mathsf{Adv}^{\mathrm{cr}}_{\mathsf{CRH},\mathsf{B}}(\lambda) + \mathsf{negl}(\lambda). \end{aligned}$$

The proof is completed. □

5.3.2 Lattice-based Hybrid Signcryption

In this section, we propose a lattice-based hybrid signcryption obtained by combining LB-SCS and a DEM. In LB-SCS, the ciphertext size for a message is $|\mathsf{m}| \log q$ for the bit-length of a message $|\mathsf{m}|$ and a modulus q. By combining LB-SCS with a DEM, the ciphertext-size for a message m is reduced to $|\mathsf{m}|$.

Let $\mathsf{DEM} = (\mathsf{DEM.Enc}, \mathsf{DEM.Dec})$ be an IND-OT secure DEM meeting *one-to-one* property [1]. Our hybrid signcryption $\mathsf{HSC} = (\mathsf{Setup}, \mathsf{KGen}_R, \mathsf{KGen}_S, \mathsf{SC}, \mathsf{USC})$ is as follows. The Setup, KGen_R, and KGen_S algorithms of HSC are the same as those of LB-SCS.

- $\mathsf{ct} \leftarrow \mathsf{SC}(\mathsf{pp}, \mathsf{pk}_R, \mathsf{sk}_S, \mathsf{m})$: To signcrypt $\mathsf{m} \in \{0,1\}^{|\mathsf{m}|}$, do the following:
  1. $K \xleftarrow{U} \{0,1\}^{\ell}$, where ℓ is the bit-length of a DEM's symmetric key.
  2. $\boldsymbol{r}_e, \boldsymbol{r}_s \leftarrow D^m_{\mathbb{Z},d}$, $\boldsymbol{t} = f_{\bar{\boldsymbol{A}}_R}(\mathsf{pk}_S) + f_{\boldsymbol{B}}(\boldsymbol{r}_e) \in \mathbb{Z}^n_q$,
  3. $\boldsymbol{A}_{R,\boldsymbol{t}} \leftarrow [\bar{\boldsymbol{A}}_R \mid H(\boldsymbol{t})\boldsymbol{G} - \bar{\boldsymbol{A}}\boldsymbol{T}_R] \in Z^{n \times m}_q$.
  4. $\boldsymbol{x}^{(0)}_0 \leftarrow \mathcal{D}^{\bar{m}}_{\mathbb{Z},\alpha q}$ and $\boldsymbol{x}^{(1)}_0 \leftarrow \mathcal{D}^{nk}_{\mathbb{Z},s}$, where $s^2 = (\|\boldsymbol{x}^{(0)}_0\|^2 + \bar{m}\alpha^2 q^2) \cdot \omega(\sqrt{\log n})^2$. Then $\boldsymbol{x}^{\top}_0 \leftarrow \boldsymbol{x}^{(0)\top}_0 \| \boldsymbol{x}^{(1)\top}_0$.
  5. $\boldsymbol{s} \xleftarrow{U} \mathbb{Z}^n_q$, $\boldsymbol{x}_1 \leftarrow \mathcal{D}^{\ell}_{\mathbb{Z},\alpha q}$.
  6. $\bar{\boldsymbol{c}}_0 = \boldsymbol{s}^{\top}\boldsymbol{A}_{R,\boldsymbol{t}} + p\boldsymbol{x}^{\top}_0 \in \mathbb{Z}^m_q$, $\bar{\boldsymbol{c}}_1 = \boldsymbol{s}^{\top}\boldsymbol{U} + p\boldsymbol{x}^{\top}_1 \in \mathbb{Z}^{\ell}_q$.
  7. $\bar{c} = (\bar{\boldsymbol{c}}_0, \bar{\boldsymbol{c}}_1, \boldsymbol{r}_e)$.
  8. Generate a signature on $\mathsf{m} \| K \| \mathsf{pk}_R \| \bar{c}$.
     - $\boldsymbol{h} \leftarrow f_{\bar{\boldsymbol{A}}_S}(\mathsf{m} \| K \| \mathsf{pk}_R \| \bar{c}) + f_{\boldsymbol{B}}(\boldsymbol{r}_s) \in \mathbb{Z}^n_q$,
     - $\boldsymbol{A}_{S,\boldsymbol{h}} \leftarrow \left[\boldsymbol{A}_S \mid \boldsymbol{A}_0 + \sum_{i=1}^{nk} h_i \cdot \boldsymbol{A}_i\right] \in \mathbb{Z}^{m+nk}_q$,
     - $\boldsymbol{e} \leftarrow \mathsf{SampleD}(\boldsymbol{T}_S, \boldsymbol{A}_{S,\boldsymbol{h}}, \boldsymbol{u}_s, \delta)$.
  9. $\boldsymbol{c}_0 \leftarrow \bar{\boldsymbol{c}}_0 + \boldsymbol{r}_s \in \mathbb{Z}^m_q$, $\boldsymbol{c}_1 \leftarrow \bar{\boldsymbol{c}}_1 + p \cdot K\lfloor \frac{q}{2} \rfloor \in \mathbb{Z}^{\ell}_q$,
  10. $\boldsymbol{c}_2 \leftarrow \mathsf{DEM.Enc}(K, \mathsf{m})$,
  11. Output $\mathsf{ct} = (\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{r}_e, \boldsymbol{e})$.
- $\mathsf{m}/\perp \leftarrow \mathsf{USC}(\mathsf{pp}, \mathsf{pk}_S, \mathsf{sk}_R, \mathsf{ct})$: To unsigncrypt $\mathsf{ct} = (\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{r}_e, \boldsymbol{e})$, do the following:
  1. $\boldsymbol{t} \leftarrow f_{\bar{\boldsymbol{A}}_R}(\mathsf{pk}_S) + f_{\boldsymbol{B}}(\boldsymbol{r}_e)$, $\boldsymbol{A}_{R,\boldsymbol{t}} \leftarrow [\bar{\boldsymbol{A}}_R \mid H(\boldsymbol{t})\boldsymbol{G} - \bar{\boldsymbol{A}}\boldsymbol{T}_R]$.
  2. $(\boldsymbol{z}, \boldsymbol{r}_s) = \mathsf{Invert}(\boldsymbol{T}_R, \boldsymbol{A}_{R,\boldsymbol{t}}, \boldsymbol{c}_0)$.
  3. $\boldsymbol{Y} := [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{\ell}] \in \mathbb{Z}^{m \times \ell}$, where let $\boldsymbol{U} := [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{\ell}]$ and for $i \in [\ell]$, $\boldsymbol{y}_i \leftarrow \mathsf{SampleD}(\boldsymbol{T}_R, \boldsymbol{A}_{R,\boldsymbol{t}}, \boldsymbol{u}_i, d)$.
  4. $\boldsymbol{v}^{\top} \leftarrow p^{-1}(\boldsymbol{c}^{\top}_1 - \boldsymbol{c}^{\top}_0\boldsymbol{Y}) = \boldsymbol{x}^{\top}_1 + \mathsf{m}^{\top}\lfloor q/2 \rfloor - \boldsymbol{x}^{\top}_0\boldsymbol{Y} \in \mathbb{Z}^{\ell}_q$.

[1] We say that a DEM meets *one-to-one* property if the DEM is deterministic and bijective.

5. For each $i \in [\ell]$, let $K_i = 0$ if v_i is closer to 0 (modulo q) than to $q/2$, and let $K_i = 1$ otherwise. Then, $K \leftarrow (K_1, \ldots, K_\ell)^\top$.
6. $\mathsf{m} \leftarrow \mathsf{DEM.Dec}(K, \boldsymbol{c}_2)$.
7. $\bar{\boldsymbol{c}}_0 \leftarrow \boldsymbol{c}_0 - \boldsymbol{r}_s \bmod q$, $\bar{\boldsymbol{c}}_1 \leftarrow \boldsymbol{c}_1 - p \cdot K\lfloor \frac{q}{2} \rfloor \bmod q$, $\bar{c} \leftarrow (\bar{\boldsymbol{c}}_0, \bar{\boldsymbol{c}}_1, \boldsymbol{r}_e)$.
8. $\boldsymbol{h} \leftarrow f_{\bar{A}_S}(\mathsf{m}\|K\|pk_R\|\bar{c}) + f_{\boldsymbol{B}}(\boldsymbol{r}_s)$, $\boldsymbol{A}_{S,h} = [\boldsymbol{A}_S \mid \boldsymbol{A}_0 + \sum_{i=1}^{nk} h_i \cdot \boldsymbol{A}_i]$.
9. Output m if $\boldsymbol{A}_{S,h} \cdot \boldsymbol{e} = \boldsymbol{u}_S \bmod q \wedge \|\boldsymbol{e}\| \leq \delta\sqrt{m+nk}$. Output $\perp$ otherwise.

The following theorems show the security of HSC.

Theorem 5.3. *If* $\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}$ *assumption holds for* $\alpha \geq 2\sqrt{n}/q$, CRH f *meets collision-resistance, and* DEM *meets* IND-OT *security and one-to-one property, then* HSC *satisfies* MU-IND-iCCA *security.*

Theorem 5.4. *If* $\mathsf{SIS}_{n,q,\beta,m+nk}$ *assumption holds for* $\beta = O((nk)^{5/2}) \cdot \omega(\sqrt{\log n})^3$, CRH f *meets collision-resistance, and* DEM *meets one-to-one property, then* HSC *satisfies* MU-sUF-iCMA *security.*

Proof of Theorem 5.3

Let A be a PPT adversary against HSC. We consider two security games Game_0 and Game_1 below. Let W_0 and W_1 be the events that A submits $b' \in \{0,1\}$ such that $b = b'$ in Game_0 and Game_1, respectively.

Game_0: This game is the ordinary MU-IND-iCCA security game. Thus, we have $\mathsf{Adv}^{\text{mu-ind-icca}}_{\mathsf{HSC,A}}(\lambda) = |\Pr[W_0] - 1/2|$. ■

Game_1: This game is the same as Game_0 except that the symmetric-key K is chosen uniformly at random.

We obtain $|\Pr[W_0] - \Pr[W_1]| \leq 2 \cdot \mathsf{Adv}^{\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}}_{\mathsf{D}}(\lambda) + \mathsf{Adv}^{\text{cr}}_{\mathsf{CRH,B}}(\lambda) + \mathsf{negl}(\lambda)$ in the same way as the proof of Theorem 5.1. The reason is as follows: From the *one-to-one* property of DEM, A cannot issue a valid unsigncrypt query $\mathsf{ct} = (\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{r}_e, \boldsymbol{e})$ such that $(\mathsf{m}, K) \neq (\mathsf{m}_b, K^*)$ and $\boldsymbol{c}_2 \neq \boldsymbol{c}_2^*$, where (m, K) is used in ct. In addition, A cannot generate a valid ct by substituting $(\boldsymbol{c}_0^*, \boldsymbol{c}_1^*, \boldsymbol{r}_e^*, \boldsymbol{e}^*)$ from the MU-IND-iCCA security of LB-SCS. Therefore, we have $|\Pr[W_0] - \Pr[W_1]| = 2 \cdot \mathsf{Adv}^{\text{mu-ind-icca}}_{\mathsf{LB\text{-}SCS,A}}(\lambda)$. Then, $|\Pr[W_0] - \Pr[W_1]| \leq 4 \cdot \mathsf{Adv}^{\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}}_{\mathsf{D}}(\lambda) + 2 \cdot \mathsf{Adv}^{\text{cr}}_{\mathsf{CRH,B}}(\lambda) + \mathsf{negl}(\lambda)$ holds.

In addition, we show $|\Pr[W_1] - 1/2| \leq \mathsf{Adv}^{\text{ind-ot}}_{\mathsf{DEM,D}'}(\lambda)$. We can simulate Setup, Queries 1, and Queries 2 phases, following the algorithms of HSC. In the Challenge phase, we encrypt the ciphertext $\boldsymbol{c}_2^*$ by sending messages to the challenger in IND-OT game. In Output phase, the guessing bit $b' \in \{0,1\}$ in IND-OT game is the same as the output of A in MU-IND-iCCA game.

From the above discussion,

$$\mathsf{Adv}^{\text{mu-ind-icca}}_{\mathsf{HSC,A}}(\lambda) \leq 4 \cdot \mathsf{Adv}^{\mathsf{LWE}_{n,q,\mathcal{D}_{\alpha q}}}_{\mathsf{D}}(\lambda) + 2 \cdot \mathsf{Adv}^{\text{cr}}_{\mathsf{CRH,B}}(\lambda) + \mathsf{Adv}^{\text{ind-ot}}_{\mathsf{DEM,D}'}(\lambda) + \mathsf{negl}(\lambda),$$

and the proof is completed. ■

Proof of Theorem 5.4

It is possible to prove Theorem 5.4 in the same way as Theorem 5.2. The reason is as follows:

- If an adversary tries to make a forgery on a queried message m, he has to make a new symmetric-key K which was not used in the $\mathsf{Queries}$ phase, because DEM meets *one-to-one* property. However, K has to be signcrypted in the same way as LB-SCS. Hence, he needs to break $\mathsf{SIS}_{n,q,\beta,m+nk}$ problem.

- Suppose that an adversary tries to make a forgery on a message m which was never queried. However, in HSC, the message has to be signcrypted in the same way as $\mathsf{LB\text{-}SCS}$. Hence, he has to break the $\mathsf{SIS}_{n,q,\beta,m+nk}$ problem in this case as well.

Therefore, by using the same proof technique in Theorem 5.2, we finally obtain

$$\mathsf{Adv}^{\text{mu-suf-icma}}_{\mathsf{HSC},\mathsf{A}}(\lambda) \leq 2\left((nk-1)q_s+1\right)\cdot\mathsf{Adv}^{\mathsf{SIS}_{n,q,\beta,m+nk}}_{\mathsf{S}}(\lambda) + \mathsf{Adv}^{\text{cr}}_{\mathsf{CRH},\mathsf{B}}(\lambda) + \mathsf{negl}(\lambda)$$

□

5.4 Signcryption in the Quantum Random Oracle Model

We construct a signcryption scheme in the QROM starting from an IND-CPA secure PKE, an IND-OT secure DEM, and a lossy identification scheme. Although our construction is based on the sign-then-encrypt methodology, it is shown in [11, 100] that the construction, by combining IND-CCA secure PKE and sUF-CMA secure DS in a trivial way of this methodology, cannot achieve MU-sUF-iCMA security while they can meet MU-IND-iCCA security. The reason is as follows: Any inside adversary can obtain a valid pair of a message and a signature from a ciphertext ct by using his/her decryption key sk_R. Hence, the adversary can make a forgery in the MU-sUF-iCMA game by encrypting the pair again. To resolve this problem, we generate a signature on $\mathsf{m}\|r$, where m is a message and r is a random value used in the underlying PKE scheme. By doing this, even if an adversary decrypts m and r, he has to generate a forgery of the underlying signature scheme.

We use the following primitives:

- Let $\mathsf{PKE} = (\mathsf{KGen}^{asy}, \mathsf{Enc}^{asy}, \mathsf{Dec}^{asy})$ be a PKE scheme with a message space $\mathcal{M}^{asy}$, a randomness space $\mathcal{R}^{asy}$, and a ciphertext space $\mathcal{C}^{asy}$.

- Let $\mathsf{LIDS} = (\mathsf{KGen}^{ids}, \mathsf{LossyKGen}^{ids}, \mathsf{P}^{ids}, \mathcal{C}^{ids}, \mathsf{V}^{ids})$ be a (commitment-recoverable) lossy identification scheme with a commitment space $\mathcal{W}$, a response space $\mathcal{Z}$, and a commitment-recoverable algorithm Rec^{ids}.
- Let $\mathsf{DEM} = (\mathsf{Enc}^{sym}, \mathsf{Dec}^{sym})$ be a DEM scheme with a key space $\mathcal{K}^{sym}$ and a message space $\mathcal{M}^{sym}$, where $\mathcal{M}^{sym}$ is the same as the message space $\mathcal{M}$ of SCS-QRO.
- Let $\mathsf{H} : \mathcal{M}^{asy} \times \mathcal{C}^{asy} \to \mathcal{K}^{sym}$, $\mathsf{G} : \mathcal{M}^{asy} \to \mathcal{R}^{asy}$, and $\mathsf{H}_S : \{0,1\}^* \to \mathcal{C}^{ids}$ be random oracles.

$\mathsf{SCS\text{-}QRO} = (\mathsf{Setup}, \mathsf{KGen}_R, \mathsf{KGen}_S, \mathsf{SC}, \mathsf{USC})$ is constructed as follows.

- $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$: Let pp^{asy} and pp^{ids} be public parameters of PKE and LIDS, respectively. Let $\kappa_m := \kappa_m(\lambda)$. Output $\mathsf{pp} := (1^\lambda, \mathsf{pp}^{asy}, \mathsf{pp}^{ids}, \kappa_m)$.
- $(\mathsf{pk}_R, \mathsf{sk}_R) \leftarrow \mathsf{KGen}_R(\mathsf{pp})$: Generate $(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}) \leftarrow \mathsf{KGen}^{asy}(1^\lambda; \mathsf{pp}^{asy})$[2], and choose $s \xleftarrow{U} \mathcal{M}^{asy}$. Output $\mathsf{pk}_R := \mathsf{pk}^{asy}$ and $\mathsf{sk}_R := (\mathsf{sk}^{asy}, s)$.
- $(\mathsf{pk}_S, \mathsf{sk}_S) \leftarrow \mathsf{KGen}_S(\mathsf{pp})$: Generate $(\mathsf{pk}^{ids}, \mathsf{sk}^{ids}) \leftarrow \mathsf{KGen}^{ids}(1^\lambda; \mathsf{pp}^{ids})$. Output $\mathsf{pk}_S := \mathsf{pk}^{ids}$ and $\mathsf{sk}_S := \mathsf{sk}^{ids}$.
- $\mathsf{ct} \leftarrow \mathsf{SC}(\mathsf{pp}, \mathsf{pk}_R, \mathsf{sk}_S, \mathsf{m})$: Compute a ciphertext on $\mathsf{m} \in \mathcal{M}$ as follows:
  1. $r \xleftarrow{U} \mathcal{M}^{asy}$, $\kappa \leftarrow 0$.
  2. $e \leftarrow \mathsf{Enc}^{asy}(\mathsf{pk}_R, r; \mathsf{G}(r))$.
  3. Do the following while $\mathsf{Z} = \perp$ and $\kappa \leq \kappa_m$:
     - $(\mathsf{W}, \mathsf{st}) \leftarrow \mathsf{P}_1^{ids}(\mathsf{sk}_S)$.
     - $\mathsf{c} \leftarrow \mathsf{H}_S(\mathsf{W}, \mathsf{m}, r, \mathsf{pk}_R, \mathsf{pk}_S)$.
     - $\mathsf{Z} \leftarrow \mathsf{P}_2^{ids}(\mathsf{sk}_S, \mathsf{W}, \mathsf{c}, \mathsf{st})$.
  4. $d \leftarrow \mathsf{Enc}^{sym}(\mathsf{k}, \mathsf{m} \| \mathsf{c} \| \mathsf{Z})$, where $\mathsf{k} = \mathsf{H}(r, e)$.
  5. Output $\mathsf{ct} := (e, d)$.
- $\mathsf{m}/\perp \leftarrow \mathsf{USC}(\mathsf{pp}, \mathsf{pk}_S, \mathsf{sk}_R, \mathsf{ct})$: Unsigncrypt $\mathsf{ct} = (e, d)$ as follows:
  1. $r' \leftarrow \mathsf{Dec}^{asy}(\mathsf{sk}_R, e)$. Output $\perp$ if $r' = \perp$.
  2. $\mathsf{k} \leftarrow \mathsf{H}(r', e)$ if $e = \mathsf{Enc}^{asy}(\mathsf{pk}_R, r'; \mathsf{G}(r'))$. $\mathsf{k} \leftarrow \mathsf{H}(s, e)$ otherwise.
  3. $M' \leftarrow \mathsf{Dec}^{sym}(\mathsf{k}, d)$. Output $\perp$ if $M' = \perp$.
  4. Parse $M' = \mathsf{m}' \| \mathsf{c}' \| \mathsf{Z}'$.
  5. $\mathsf{W}' \leftarrow \mathsf{Rec}^{ids}(\mathsf{pk}_S, \mathsf{c}', \mathsf{Z}')$.
  6. Output m' if $\mathsf{H}_S(\mathsf{W}', \mathsf{m}', r', \mathsf{pk}_R, \mathsf{pk}_S) = \mathsf{c}'$, and output $\perp$ otherwise.

[2] In this section, for a key generation algorithm KGen of PKE or LIDS, we write $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda; \mathsf{pp})$, where λ is a security parameter, and pp is a public parameter based on λ.

Theorem 5.5 and 5.6 show the security of SCS-QRO.

Theorem 5.5. *If* PKE *meets* IND-CPA *security and* DEM *meets* IND-OT *security and one-to-one property, then* SCS-QRO *satisfies* MU-IND-iCCA *security in the quantum random oracle model.*

Theorem 5.6. *If* LIDS *meets* naHVZK*,* α *bits* min-entropy*,* CUR *property,* key-indistinguishability*, and* lossy-soundness*, and* DEM *meets one-to-one property, then* SCS-QRO *meets* MU-sUF-iCMA *security in the quantum random oracle model.*

In the proofs of Theorem 5.5 and Theorem 5.6, we use the following notations: Let A be a PPT adversary against SCS-QRO. Let q_s and q_u be the number of queries which A issues to SC^{O} and $\mathsf{USC}^{\mathsf{O}}$ oracles, respectively. Let q_h, q_{h_s}, and q_g be the number of queries which A submits to random oracles H, H_S, and G, respectively.

Proof of Theorem 5.5

For $i \in \{0, 1, \ldots, 9\}$, we consider security games Game_i, let W_i be the event that A outputs $b' \in \{0,1\}$ such that $b = b'$ in Game_i, let Find_i be the event that a random oracle H, H_S, or G submits a query to a semi-classical oracle O_S^{SC} and it returns $|1\rangle$ in Game_i, and let $\neg\mathsf{Find}_i$ be the event that O_S^{SC} always returns $|0\rangle$ in Game_i.

Game_0: This game is the same as the ordinary MU-IND-iCCA game. So, we have $\mathsf{Adv}^{\text{mu-ind-icca}}_{\text{SCS-QRO},\mathsf{A}}(\lambda) = \left|\Pr[W_0] - \frac{1}{2}\right|$. ■

For any $(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}) \leftarrow \mathsf{KGen}^{asy}(1^\lambda; \mathsf{pp}^{asy})$ and any $r \in \mathcal{M}^{asy}$, a set of "bad" random coins is defined as

$$\mathcal{R}^{asy}_{bad}(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}, r) := \{\hat{r} \in \mathcal{R}^{asy} \mid \mathsf{Dec}^{asy}(\mathsf{sk}^{asy}, \mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r; \hat{r})) \neq r\},$$

and a set of "good" random coins is defined as $\mathcal{R}^{asy}_{good} = \mathcal{R}^{asy} \backslash \mathcal{R}^{asy}_{bad}(\mathsf{pk}^{asy})$. Then, let

$$\delta(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}, r) := \frac{\left|\mathcal{R}^{asy}_{bad}(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}, r)\right|}{|\mathcal{R}^{asy}|},$$

and let $\delta(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}) := \max_{r \in \mathcal{M}^{asy}} \delta(\mathsf{pk}^{asy}, \mathsf{sk}^{asy})$. Then, we have the expectation $\delta = \mathbf{E}[\delta(\mathsf{pk}_R, \mathsf{sk}_R)]$ which is taken over $(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}) \leftarrow \mathsf{KGen}^{asy}(1^\lambda; \mathsf{pp}^{asy})$.

Game_1: This game is the same as Game_0 except that $\mathsf{H}(s, e)$ in $\mathsf{USC}^{\mathsf{O}}$ oracle is replaced by $\mathsf{H}_q(e)$, where $\mathsf{H}_q : \mathcal{M}^{asy} \to \mathcal{R}^{asy}$ is a random oracle.

We apply Lemma 6 in [78] in the straightforward way. That is, $\mathsf{H}(s, \cdot)$ and $\mathsf{H}_q(\cdot)$ are viewed as $F_0(\cdot)$ and $F_1(\cdot)$ oracles in this lemma, respectively. Then, we have $|\Pr[W_0] - \Pr[W_1]| \leq 2q_h/\sqrt{\mathcal{K}^{sym}}$. ■

Game_2: This game is the same as Game_1 except that the random oracle $\mathsf{G}(r)$ is replaced by $\mathsf{G}'(r)$ which samples uniformly random values from a set $\mathcal{R}^{asy}_{good}$ of "good" random coins.

G and G' can be viewed as functions F and N in the generic search problem, respectively. Thus, we get the following equations: $\Pr[F(r) = 1] = \delta(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}, r)$, $\Pr[F(r) = 0] = 1 - \delta(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}, r)$, and $\Pr[N(r) = 0] = 1$. Then, we get

$$|\Pr[W_1 \mid (\mathsf{pk}^{asy}, \mathsf{sk}^{asy})] - \Pr[W_2 \mid (\mathsf{pk}^{asy}, \mathsf{sk}^{asy})]| \leq 2q_g\sqrt{\delta(\mathsf{pk}^{asy}, \mathsf{sk}^{asy})}.$$

Hence, by averaging this equation over $(\mathsf{pk}^{asy}, \mathsf{sk}^{asy}) \leftarrow \mathsf{KGen}^{asy}(1^\lambda; \mathsf{pp}^{asy})$, we obtain $|\Pr[W_1] - \Pr[W_2]| \leq 2q_g\sqrt{\delta}$. ■

Game_3: This game is the same as Game_2 except that the random oracles $\mathsf{H}_S(\mathsf{W}, \mathsf{m}, r, \mathsf{pk}_R, \mathsf{pk}_S)$ and $\mathsf{H}(r, e)$ return $\mathsf{H}'_S(\mathsf{W}, \mathsf{m}, \mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r; \mathsf{G}'(r)), \mathsf{pk}_R, \mathsf{pk}_S)$ and $\mathsf{H}_q(\mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r; \mathsf{G}'(r)))$, respectively.

Since the random oracle G' samples only "good" random coins, the encryption algorithm $\mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, \cdot; \mathsf{G}'(\cdot))$ is injective. The statistical distance between values of $(\mathsf{H}_S, \mathsf{H})$ and $(\mathsf{H}'_S, \mathsf{H}_q)$ is negligible. Hence, we have $\Pr[W_3] = \Pr[W_2]$. ■

Game_4: This game is the same as Game_3 except that $\mathsf{USC}^{\mathsf{O}}$ oracle is changed as follows:

1. $\mathsf{k} \leftarrow \mathsf{H}_q(e)$.
2. $M' \leftarrow \mathsf{Dec}^{sym}(\mathsf{k}, d)$ and output $\perp$ if $M' = \perp$.
3. Parse $M' = \mathsf{m}'\|\mathsf{c}'\|\mathsf{Z}'$.
4. $\mathsf{W}' \leftarrow \mathsf{Rec}^{ids}(\mathsf{pk}_S, \mathsf{c}', \mathsf{Z}')$.
5. Output m' if $\mathsf{c}' = \mathsf{H}_S(\mathsf{W}', \mathsf{m}, e, \mathsf{pk}_R, \mathsf{pk}_S)$, and output $\perp$ otherwise.

That is, the modified $\mathsf{USC}^{\mathsf{O}}$ oracle does not use a receiver's secret key.

We consider several cases for unsigncrypt queries (e, d). In the case of $e = \mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r; \mathsf{G}'(r))$ and $\mathsf{c} = \mathsf{H}_S(\mathsf{W}, \mathsf{m}, r, \mathsf{pk}_R, \mathsf{pk}_S)$, $\mathsf{USC}^{\mathsf{O}}$ oracles in both games return a message $\mathsf{m} \neq \perp$. In the case of $e = \mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r; \mathsf{G}'(r))$ and $\mathsf{c} \neq \mathsf{H}_S(\mathsf{W}, \mathsf{m}, r, \mathsf{pk}_R, \mathsf{pk}_S)$, $\mathsf{USC}^{\mathsf{O}}$ oracles in both games return an invalid symbol $\perp$. In the case of $e \neq \mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r; \mathsf{G}'(r))$ and $\mathsf{c} \neq \mathsf{H}_S(\mathsf{W}, \mathsf{m}, r, \mathsf{pk}_R, \mathsf{pk}_S)$, both $\mathsf{USC}^{\mathsf{O}}$ oracles return $\perp$. In the case of $e \neq \mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r; \mathsf{G}'(r))$ and $\mathsf{c} = \mathsf{H}_S(\mathsf{W}, \mathsf{m}, r, \mathsf{pk}_R, \mathsf{pk}_S)$, the $\mathsf{USC}^{\mathsf{O}}$ oracle in Game_3 returns $\perp$ while the $\mathsf{USC}^{\mathsf{O}}$ oracle in Game_4 returns m. However, since G' returns only "good" random coins, this case of $e \neq \mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r; \mathsf{G}'(r))$ does not happen. Hence, we have $\Pr[W_4] = \Pr[W_3]$. ■

Game_5: This game is the same as Game_4 except that we replace G' by an ideal random oracle G.

In the same way as the proof in the game-hop of Game_1, we get the inequality $|\Pr[W_4] - \Pr[W_5]| \leq 2q_g\sqrt{\delta}$. ■

$\ddot{\mathsf{G}}$ is a random oracle which $\ddot{\mathsf{G}}(r)$ is sampled from $\mathcal{R}^{asy}$ uniformly at random if $r = r^*$, and $\ddot{\mathsf{G}}(r) := \mathsf{G}(r)$ otherwise. $\ddot{\mathsf{H}}$ and $\ddot{\mathsf{H}}_S$ are defined in the same way as $\ddot{\mathsf{G}}$.

Game_6: This game is the same as Game_5 except that we replace G, H, and H_S by $\ddot{\mathsf{G}}\backslash\{r^*\}$, $\ddot{\mathsf{H}}\backslash\{r^*\}$, and $\ddot{\mathsf{H}}_S\backslash\{r^*\}$, respectively.

From Theorem 1 in [9], we have

$$\begin{aligned}&|\Pr[W_5] - \Pr[W_6 \wedge \neg\mathsf{Find}_6]| + \left|\Pr[W_6 \wedge \neg\mathsf{Find}_6] - \frac{1}{2}\right| \\ \leq &\sqrt{2(q_g + q_h + q_{h_s})\Pr[\mathsf{Find}_6]} + \left|\Pr[W_6 \wedge \neg\mathsf{Find}_6] - \frac{1}{2}\right|.\end{aligned}$$

We show $|\Pr[W_6 \wedge \neg\mathsf{Find}_6] - 1/2| \leq \mathsf{Adv}^{\text{ind-ot}}_{\mathsf{DEM},\mathsf{D}^{sym}}(\lambda)$. The following PPT algorithm D^{sym} against DEM is constructed as follows: At the beginning of the game, it generates $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{pk}_R, \mathsf{sk}_R) \leftarrow \mathsf{KGen}_R(\mathsf{pp})$ and sends $(\mathsf{pp}, \mathsf{pk}_R)$ to A. D^{sym} can simulate $\mathsf{USC}^{\mathsf{O}}$ oracle and random oracles since $\mathsf{USC}^{\mathsf{O}}$ oracle does not use any secret key from the game-hop of Game_4 and $2q'$-wise independent hash functions can be used as random oracles ($q' \in \{q_g, q_h, q_{h_s}\}$), from Theorem 6.1 in [137]. When A submits $(\mathsf{m}_0, \mathsf{m}_1, \mathsf{pk}^*_S, \mathsf{sk}^*_S)$, D^{sym} computes $(e^*, \mathsf{c}^*, \mathsf{Z}^*)$ by following SC algorithm. It gets d^* by issuing $(\mathsf{m}_0\|\mathsf{c}^*\|\mathsf{Z}^*, \mathsf{m}_1\|\mathsf{c}^*\|\mathsf{Z}^*)$ to the challenger of $\mathsf{IND\text{-}OT}$ game, and returns (e^*, d^*). When A outputs $b' \in \{0,1\}$, D^{sym} also submits b'. Notice that A cannot submit a valid query $\mathsf{ct} = (e, d)$ such that $d \neq d^*$ and $(\mathsf{m}, r) = (\mathsf{m}_b, r^*)$ from the *one-to-one* property of DEM and the assumption of $[W_6 \wedge \neg\mathsf{Find}_6]$, where (m, r) is used in the query ct.

From the above, D^{sym} simulates the environment of A in Game_6. If A guesses the signcrypted message m_b, D^{sym} also wins in $\mathsf{IND\text{-}OT}$ game. Hence, we have the inequality $|\Pr[W_6 \wedge \neg\mathsf{Find}_6] - 1/2| \leq \mathsf{Adv}^{\text{ind-ot}}_{\mathsf{DEM},\mathsf{D}^{sym}}(\lambda)$. ■

Game_7: This game is the same as Game_6 except that we replace $\hat{r}^* \leftarrow \mathsf{G}(r^*)$, $\mathsf{k}^* \leftarrow \mathsf{H}(r^*, e^*)$, and $\mathsf{c}^* \leftarrow \mathsf{H}_S(\mathsf{W}^*, \mathsf{m}_b, r^*, \mathsf{pk}_R, \mathsf{pk}^*_S)$ by $\hat{r}^* \xleftarrow{U} \mathcal{M}^{asy}$, $\mathsf{k}^* \xleftarrow{U} \mathcal{K}^{sym}$, and $\mathsf{c}^* \xleftarrow{U} \mathcal{C}^{ids}$, respectively.

We consider only whether event Find happens or not. In the two games Game_6 and Game_7, A is not given the values $(\mathsf{G}(r^*), \mathsf{H}(r^*, \cdot), \mathsf{H}_S(\cdot, \cdot, r^*, \cdot, \cdot))$. Hence, $\Pr[\mathsf{Find}_7] = \Pr[\mathsf{Find}_6]$ holds. ■

Game_8: This game is the same as Game_7 except that $\ddot{\mathsf{G}}$, $\ddot{\mathsf{H}}$, and $\ddot{\mathsf{H}}_S$ are replaced by G, H, and H_S, respectively.

Since $\mathsf{G}(r^*)$, $\mathsf{H}(r^*, \cdot)$, and $\mathsf{H}_S(\cdot, \cdot, r^*, \cdot, \cdot))$ are not used in the two games, this replacement does not influence the view of A, and $\Pr[\mathsf{Find}_8] = \Pr[\mathsf{Find}_7]$ holds. ■

Game_9: This game is the same as Game_8 except that in $\mathsf{Challenge}$ phase, we

replace r^* by r'^*. Notice that random oracles $\mathsf{G}\backslash\{r^*\}$, $\mathsf{H}\backslash\{r^*\}$, and $\mathsf{H}_S\backslash\{r^*\}$ are used in Game_9.

By Corollary 1 in [9], we have $\Pr[\mathsf{Find}_9] \leq 4(q_g+q_h+q_{h_s})/\,|\mathcal{M}^{asy}|$. Next, we show $|\Pr[\mathsf{Find}_8] - \Pr[\mathsf{Find}_9]| \leq \mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathsf{D}^{asy}}(\lambda)$. A PPT adversary D^{asy} against PKE is constructed in the following way: Given pk^{asy}, it chooses $r^*, r'^* \xleftarrow{U} \mathcal{M}^{asy}$ and submits these as the challenge messages in IND-CPA game. After receiving the challenge ciphertext e^*, D^{asy} sets $(\mathsf{pp}, \mathsf{pk}_R)$ and sends $(\mathsf{pp}, \mathsf{pk}_R)$ to A. When A submits queries to random oracles $\mathsf{G}, \mathsf{H}, \mathsf{H}_S$, it submits them to a semi-classical oracle $\mathsf{O}^{SC}_{\{r^*\}}$ outputs 1 if this oracle returns $|1\rangle$. D^{asy} can simulate $\mathsf{USC}^{\mathsf{O}}$ oracle by following the game-hop of Game_4. When A outputs $b' \in \{0,1\}$, and $\mathsf{O}^{SC}_{\{r^*\}}$ never returns $|1\rangle$, then D^{asy} outputs 0.

D^{asy} simulates the environment of A in Game_8 or Game_9 if it is given $e^* = \mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r^*)$ or $e^* = \mathsf{Enc}^{asy}(\mathsf{pk}^{asy}, r'^*)$ as input, respectively. Hence, we have $|\Pr[\mathsf{Find}_8] - \Pr[\mathsf{Find}_9]| \leq \mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathsf{D}}(\lambda)$. ■

From the above discussion, we obtain the advantage

$$\mathsf{Adv}^{\text{mu-ind-icca}}_{\mathsf{SCS\text{-}QRO},\mathsf{A}}(\lambda) \leq \sqrt{2(q_g+q_h+q_{h_s})\mathsf{Adv}^{\text{ind-cpa}}_{\mathsf{PKE},\mathsf{D}^{asy}}(\lambda) + 8\frac{(q_g+q_h+q_{h_s})^2}{|\mathcal{M}^{asy}|}} + \mathsf{Adv}^{\text{ind-ot}}_{\mathsf{DEM},\mathsf{D}^{sym}}(\lambda) + 2q_g\sqrt{\delta} + \frac{2q_h}{\sqrt{\mathcal{M}^{asy}}},$$

and the proof is completed. □

Proof of Theorem 5.6

For $i \in \{0,1,2,3,4\}$, we consider security games Game_i, and let W_i be the event that A wins in Game_i.

Game_0: This game is the same as the ordinary MU-sUF-iCMA security game. Thus, we have $\mathsf{Adv}^{\text{mu-suf-icma}}_{\mathsf{SCS\text{-}QRO},\mathsf{A}}(\lambda) = \Pr[W_0]$. ■

Game_1: This game is the same as Game_0 except that SC^{O} oracle is modified as follows: Let $c \leftarrow 0$ be a counter.

1. $r \xleftarrow{U} \mathcal{M}^{asy}$, $c \leftarrow c+1$.
2. $e \leftarrow \mathsf{Enc}^{asy}(\mathsf{pk}_R, r; \mathsf{G}(r))$.
3. $(\mathsf{W}_{M,c}, \mathsf{c}_{M,c}, \mathsf{Z}_{M,c}) \leftarrow \mathsf{GetTrans}^{ids}(M,c)$, where $M \leftarrow \mathsf{m}\|r\|\mathsf{pk}_R\|\mathsf{pk}_S$.
4. Output $\perp$ if $(\mathsf{W}_{M,c}, \mathsf{c}_{M,c}, \mathsf{Z}_{M,c}) = (\perp,\perp,\perp)$.
5. $d \leftarrow \mathsf{Enc}^{sym}(\mathsf{k}, \mathsf{m}\|\mathsf{c}_{M,c}\|\mathsf{Z}_{M,c})$, where $\mathsf{k} = \mathsf{H}(r,e)$.
6. Output $\mathsf{ct} := (e,d)$.

Here, $\mathsf{GetTrans}^{ids}(M, c)$ is defined as follows: Let $\mathsf{RF} : \{0,1\}^* \to \mathcal{R}^{ids}$ be a random function.

1. $\kappa \leftarrow 0$.
2. While $\mathsf{Z}_{M,c} = \perp$ and $\kappa \leq \kappa_m$:
   - $\kappa \leftarrow \kappa + 1$.
   - $(\mathsf{W}_{M,c}, \mathsf{st}) \leftarrow \mathsf{P}_1^{ids}(\mathsf{sk}_S; \mathsf{RF}(0\|M\|\kappa\|c))$.
   - $\mathsf{c}_{M,c} \leftarrow \mathsf{H}_S(\mathsf{W}_{M,c}\|M)$.
   - $\mathsf{Z}_{M,c} \leftarrow \mathsf{P}_2^{ids}(\mathsf{sk}_S, \mathsf{W}_{M,c}, \mathsf{c}_{M,c}, \mathsf{st}; \mathsf{RF}(1\|M\|\kappa\|c))$.
3. If $\mathsf{Z}_{M,c} = \perp$, $(\mathsf{W}_{M,c}, \mathsf{c}_{M,c}) \leftarrow (\perp, \perp)$.
4. Output $(\mathsf{W}_{M,c}, \mathsf{c}_{M,c}, \mathsf{Z}_{M,c})$.

This modification is conceptual because we just add a counter c so that $(\mathsf{W}_{M,c}, \mathsf{c}_{M,c}, \mathsf{Z}_{M,c})$ can be viewed as a random value. Hence, we have $\Pr[W_1] = \Pr[W_0]$. ■

Game_2: This game is the same as Game_1 except that $(\mathsf{W}_{M,c}, \mathsf{c}_{M,c}, \mathsf{Z}_{M,c})$ on $M = \mathsf{m}\|r\|\mathsf{pk}_R\|\mathsf{pk}_S$ are generated by the simulator of LIDS and the random oracle H_S is programmed by following this modification. Concretely, $\mathsf{GetTrans}^{ids}(M, c)$ is modified as follows: Let S^{ids} be the simulator of LIDS.

1. $\kappa \leftarrow 0$.
2. While $\mathsf{Z}_{M,c} = \perp$ and $\kappa \leq \kappa_m$:
   - $\kappa \leftarrow \kappa + 1$.
   - $(\mathsf{W}_{M,c}, \mathsf{c}_{M,c}, \mathsf{Z}_{M,c}) \leftarrow \mathsf{S}^{ids}(\mathsf{pk}_S; \mathsf{RF}(M\|\kappa\|c))$.
3. If $\mathsf{Z}_{M,c} = \perp$, $(\mathsf{W}_{M,c}, \mathsf{c}_{M,c}, \mathsf{Z}_{M,c}) \leftarrow (\perp, \perp, \perp)$.
4. Output $(\mathsf{W}_{M,c}, \mathsf{c}_{M,c}, \mathsf{Z}_{M,c})$.

Besides, $\mathsf{H}_S(\mathsf{W}\|M)$ is modified as follows:

1. For $i \in [q_s]$, do the following:
   - $(\mathsf{W}_{M,c}, \mathsf{c}_{M,c}, \mathsf{Z}_{M,c}) \leftarrow \mathsf{GetTrans}^{ids}(M, c)$.
   - If $\mathsf{W} = \mathsf{W}_{M,c}$, return $\mathsf{c} \leftarrow \mathsf{c}_{M,c}$.
2. Return $\mathsf{c} \leftarrow \mathsf{H}'_S(\mathsf{W}\|M)$, where H'_S is a random oracle which A cannot access directly.

By the naHVZK of LIDS, the statistical distance between (c, Z) in Game_1 and Game_2 is at most $\kappa_m \varepsilon_{\mathsf{zk}}$. Since (c, Z) must be a valid signature on M, we need to patch the values of the quantum random oracle H_S like the modification above. Hence, we have $|\Pr[W_1] - \Pr[W_2]| \leq \kappa_m q_s \cdot \varepsilon_{\mathsf{zk}}$. ■

Game_3: This game is the same as Game_2 except that the challenger aborts if $\mathsf{c}^* \neq \mathsf{H}_S(\mathsf{W}^*, M^*)$ holds for the values $(\mathsf{W}^*, \mathsf{c}^*, \mathsf{Z}^*)$ obtained from the output ct^* of A.

First, we consider the case in which A generates a forgery on $M^* = \mathsf{m}^* \| r^* \| \mathsf{pk}_R^* \| \mathsf{pk}_S$ which is not queried to SC^{O} oracle, and $\mathsf{W}^* = \mathsf{W}_{M^*,c}$ holds for $c \in [q_s]$. By the α bits min-entropy of LIDS, $\mathsf{W}_{M^*,c}$ is not revealed for all $c \in [q_s]$. Thus, the success probability in this case is at most $q_s \cdot 2^{-\alpha+1}$.

Next, we consider the case in which A generates $(\mathsf{W}_{M^*,c^*}, \mathsf{c}_{M^*,c^*}, \mathsf{Z}_{M^*,c^*})$ for M^* and c^* which were used in SC^{O} oracle. By the CUR of LIDS, it is computationally difficult for A to generate Z_{M^*,c^*} which was not used in SC^{O} oracle. Hence, the upper bound that A generates such a response is $\mathsf{Adv}^{\mathrm{cur}}_{\mathsf{LIDS},\mathsf{C}}(\lambda)$.

Therefore, we obtain $|\Pr[W_2] - \Pr[W_3]| \leq q_s \cdot 2^{-\alpha+1} + \mathsf{Adv}^{\mathrm{cur}}_{\mathsf{LIDS},\mathsf{C}}(\lambda)$. ■

Game_4: This game is the same as Game_3 except for replacing pk with a lossy public key $\mathsf{pk}_{\mathsf{ls}}$.

By using $2q'$-wise independent hash functions for $q' \in \{q_{h_s}, q_h, q_g\}$ in order to simulate the random oracles H_S, H, and G, it is possible to construct a PPT adversary D against LIDS such that

$$|\Pr[W_3] - \Pr[W_4]| \leq \mathsf{Adv}^{\mathrm{loss}}_{\mathsf{LIDS},\mathsf{D}}(\lambda).$$

In addition, we show $\Pr[W_4]$ is negligible in λ if LIDS fulfills lossy-soundness. For any $(\mathsf{pk}_S, \mathsf{sk}_S) \leftarrow \mathsf{KGen}_S(\mathsf{pp})$ and any $W \in \mathsf{W}$, a set of "good" challenges is defined as $\mathcal{C}^{ids}_{good}(\mathsf{pk}_S, \mathsf{W}) := \{\mathsf{c} \in \mathcal{C}^{ids} \mid \exists \mathsf{Z} \in \mathcal{Z}, \mathsf{V}^{ids}(\mathsf{pk}_S, \mathsf{W}, \mathsf{c}, \mathsf{Z}) = 1\}$. $\mathcal{C}^{ids}_{good}(\mathsf{pk}_S, \mathsf{W})$ contains only challenges c such that there exists a response Z satisfying $\mathsf{V}^{ids}(\mathsf{pk}_S, \mathsf{W}, \mathsf{c}, \mathsf{Z}) = 1$.

We construct a PPT algorithm F solving the generic search problem with bounded probabilities as follows: At the beginning of the game, it generates $\mathsf{pk}_S = \mathsf{pk}^{ids}_{\mathsf{ls}} \leftarrow \mathsf{LossyKGen}(\mathsf{pp})$ and chooses a $2q_{h_s}$-wise independent hash function f_{H_S}. For each $\mathsf{W} \in \mathcal{W}$, F does the following:

1. Compute $\mathcal{C}^{ids}_{good}(\mathsf{pk}_S, \mathsf{W}) \subseteq \mathcal{C}^{ids}$.
2. Let $\gamma_{\mathsf{pk}^{ids}_{\mathsf{ls}}}(\mathsf{W}) := |\mathcal{C}^{ids}_{good}(\mathsf{pk}_S, \mathsf{W})|/|\mathcal{C}^{ids}|$.
3. For all M, let $\gamma_{\mathsf{pk}^{ids}_{\mathsf{ls}}}(\mathsf{W}\|M) := \gamma_{\mathsf{pk}^{ids}_{\mathsf{ls}}}(\mathsf{W})$.

Then, it outputs $\gamma_{\mathsf{pk}^{ids}_{\mathsf{ls}}}(\mathsf{W}\|M)$ for all $\mathsf{W} \in \mathcal{W}$ and all M. F simulates oracles SC^{O} in the same way as Game_3 and simulates H and G by using a $2q_h$-wise independent hash function and a $2q_g$-wise independent hash function, respectively. It simulates $\mathsf{H}_S(\mathsf{W}\|M)$ as follows: If $F(\mathsf{W}\|M) = 1$, it returns $\mathsf{W} \in$

$\mathcal{C}^{ids}_{good}(\mathsf{pk}_S, \mathsf{W})$ chosen uniformly at random by using a random coin $f_{\mathsf{H}_S}(\mathsf{W}\|M)$. If $F(\mathsf{W}\|M) = 0$, it returns uniformly random $\mathsf{W} \in \mathcal{C}^{ids} \backslash \mathcal{C}^{ids}_{good}(\mathsf{pk}_S, \mathsf{W})$ by using a random coin $f_{\mathsf{H}_S}(\mathsf{W}\|M)$. In Output phase, when A outputs $(\mathsf{pk}^*_R, \mathsf{sk}^*_R, \mathsf{ct}^*)$, F computes M^* and $(\mathsf{W}^*, \mathsf{c}^*, \mathsf{Z}^*)$ by using sk^*_R. Finally, it returns $\mathsf{W}^*\|M^*$ if $\mathsf{H}_S(\mathsf{W}^*, \mathsf{m}^*, r^*, \mathsf{pk}^*_R, \mathsf{pk}_S) = \mathsf{c}^*$, and outputs $\perp$ otherwise. Then, c^* is a "good" challenge which implies $F(\mathsf{W}\|M) = 1$ holds with probability $|\mathcal{C}^{ids}_{good}(\mathsf{pk}_S, \mathsf{W})| / \left|\mathcal{C}^{ids}\right|$.

If the forgery $(\mathsf{c}^*, \mathsf{Z}^*)$ on M^* is valid, c^* is in $\mathcal{C}^{ids}_{good}(\mathsf{pk}_S, \mathsf{W}^*)$, that is, A finds $\mathsf{W}^*\|M^*$ such that $F(\mathsf{W}^*\|M^*) = 1$. Notice that since LIDS is commitment-recoverable, the condition of $\mathcal{C}^{ids}_{good}(\mathsf{pk}_S, \mathsf{W}^*)$ is identical to the verification by $\mathsf{H}_S(\mathsf{W}^*, \mathsf{m}^*, r^*, \mathsf{pk}^*_R, \mathsf{pk}_S) = \mathsf{c}^*$. Thus, for fixed $\mathsf{pk}^{ids}_{\mathsf{ls}}$, the success probability in Game_4 is at most $8(q_h+1)^2\gamma_{\mathsf{pk}^{ids}_{\mathsf{ls}}}$, where $\gamma_{\mathsf{pk}^{ids}_{\mathsf{ls}}} = \max_{\mathsf{W},M} \gamma_{\mathsf{pk}^{ids}_{\mathsf{ls}}}(\mathsf{W}\|M)$. The average of $8(q_{h_s}+1)^2\gamma_{\mathsf{pk}^{ids}_{\mathsf{ls}}}$ over $\mathsf{pk}^{ids}_{\mathsf{ls}} \leftarrow \mathsf{LossyKGen}(\mathsf{pp})$ is $8(q_h+1)^2\varepsilon_{\mathsf{ls}}$. Hence, we have $\Pr[W_4] \leq 8(q_{h_s}+1)^2\varepsilon_{\mathsf{ls}}$. ■

In addition, the *one-to-one* property of DEM also guarantees the strong unforgeability of $\mathsf{SCS\text{-}QRO}$ even though the adversary A tries to generate the same randomness as query/response pairs submitted to the signcrypt oracle.

From the discussion above, we obtain

$$\mathsf{Adv}^{\text{mu-suf-icma}}_{\mathsf{SCS\text{-}QRO},\mathsf{A}}(\lambda) \leq \kappa_m q_s \cdot \varepsilon_{\mathsf{zk}} + \frac{q_s}{2^{\alpha-1}} + \mathsf{Adv}^{\text{cur}}_{\mathsf{LIDS},\mathsf{C}}(\lambda) + \mathsf{Adv}^{\text{loss}}_{\mathsf{LIDS},\mathsf{D}}(\lambda) + 8(q_{h_s}+1)^2\varepsilon_{\mathsf{ls}}.$$

The proof is completed. □

5.5 Comparison of Signcryption Schemes

We compare our schemes with existing ones in terms of key-sizes (i.e., sizes of public-keys and secret-keys), and ciphertext-size in order to evaluate efficiency among the constructions. Our schemes are two constructions. One is the construction HSC in Section 5.3. The other is a scheme obtained by applying lattice-based primitives to $\mathsf{SCS\text{-}QRO}$ in Section 5.4. Concretely, we apply an IND-CPA secure PKE [90] and a lossy identification scheme [84] to the scheme.

To the best of our knowledge, $\mathsf{LB\text{-}SCS}$ and HSC are the first direct constructions of signcryption based on lattice problems without random oracles. Hence, there is no other lattice-based construction to compare efficiency with ours. However, since there are generic constructions of signcryption [34, 104] satisfying the strongest security (i.e., both MU-IND-iCCA security and MU-sUF-iCMA security) without (quantum) random oracles, we can obtain lattice-based signcryption schemes by applying suitable lattice-based primitives to the generic constructions. Specifically, we consider the following applications of lattice-based primitives.

- SCS_{TK}[34]: We apply IND-Tag-CCA secure Tag-based KEM ([101] and [31]), sUF-CMA secure DS ([101] and [31]), IND-CCA secure DEM.

- SCS_{KEM}[34]: We apply IND-CCA secure KEM ([101] and [22]), sUF-CMA secure DS ([101] and [31]), IND-OT secure DEM, sUF-OT-CMA secure MAC.
- SCS_{CHK}[104]: We apply IND-sID-CPA secure Identity-based Encryption [3], EUF-CMA secure DS [26], sUF-OT-CMA secure OTS [98].

In the description above, KEM is a key encapsulation mechanism, DS is a digital signature, MAC is a message authentication code, and OTS is a one-time signature. IND-Tag-CCA means indistinguishability against adaptive tag chosen ciphertext attacks, sUF-OT-CMA means strong unforgeability against one-time chosen message attacks, and IND-sID-CPA means indistinguishability against selective ID chosen plaintext attacks. Then, to fairly compare efficiency of lattice-based signcryption, we take into account the following:

1. In SCS_{TK}, SCS_{KEM}, and our schemes, we assume that the paradigm of authenticated encryption in [17] is used to obtain IND-CCA secure DEM (or IND-OT secure DEM). Namely, IND-CCA secure DEM is obtained from IND-CPA secure symmetric-key encryption (SKE) and sUF-CMA secure MAC. These SKE and MAC can be constructed from the AES meeting the 128-bit security and IND-CPA security. The key-sizes are set to be at least 512 bits, since it is necessary to have resistance against quantum computing by taking into account the power of the Grover's algorithm.
2. In SCS_{TK}, IND-Tag-CCA secure tag-based KEM is required. However, there is no lattice-based construction meeting this security. We construct this tag-based KEM by combining tag-based KEM achieving weaker security and a chameleon hash function [31] in a generic way.
3. In SCS_{KEM}, we construct IND-CCA secure KEM by the BK-transformation [22]. This reason is that even if we consider realizing CCA-secure KEM based on the lattice problems [110], the resulting signcryption will be less efficient than SCS_{TK} and SCS_{CHK} obviously.

Table 5.1 shows comparison in sizes of public/secret-keys and ciphertexts. We can see that the key-sizes and ciphertext-size of SCS-QRO are the shortest of all schemes though it is secure in the QROM. Regarding schemes in the standard model, Table 5.1 shows the following: Although it can be seen that SCS_{CHK} and our scheme HSC are the most efficient in terms of receiver's and sender's public-key sizes. The ciphertext-size of HSC is the shortest of all schemes in the standard model.

From the above discussion, the public-key sizes and ciphertext-size of our scheme HSC are shorter than those of existing ones, and there is no disadvantage for ours compared to other ones. SCS-QRO is the best of all schemes in terms of key-sizes and ciphertext-size.

Constructions	Receiver's key size (bit-length)		Sender's key size (bit-length)		Ciphertext size (bit-length)
	public key	secret key	public key	secret key	
SCS_{TK}	$3nmk$	$nmk \log d$	$3nmk$	$nmk \log d$	$(m+K)k$ $+3m \log d + \lvert\mathsf{m}\rvert$
SCS_{KEM}	$2nmk$				$(2m+K+n)k$ $+2m \log d + \lvert\mathsf{m}\rvert$ $+2\lvert MAC\rvert$
SCS_{CHK}	nmk		nmk		$(3m+n)k$ $+\lvert vk\rvert + \lvert\mathsf{m}\rvert k$
HSC (Our Scheme)					$(m+K)k$ $+2m \log d + \lvert\mathsf{m}\rvert$
SCS-QRO (Our Scheme)	nKk	$nK \log d$	$n\ell k$	$2n\ell \log d$	$(n+K)k + 2n\ell \log d$ $+n + \lvert\mathsf{m}\rvert$

Table 5.1: Comparison of sizes of public/secret keys and ciphertexts: A positive integer n is a security parameter, q is a prime, a positive integer $m = O(nk)$ is a dimension of a lattice, $d(\ll q)$ is a value of an element sampled from a Gaussian distribution in $\mathbb{Z}$, $\ell(\ll n)$ is a dimension of matrices, $|MAC|$ is the bit-length of MAC tags, K is the bit-length of symmetric keys of DEM, $|vk|$ is the bit-length of an OTS's verification-key, and $|\mathsf{m}|$ is the bit-length of a message.

Chapter 6

Conclusion

We dealt with quantum-secure cryptographic schemes of encryption, authentication, and signcryption which guarantees both securities of encryption and authentication.

First, we focused on the selective opening (SO) security of public key encryption (PKE). We proved that two hybrid encryption schemes satisfy simulation-based SO security against chosen ciphertext attacks (SIM-SO-CCA security) in the quantum random oracle model (QROM) or the quantum ideal cipher model (QICM). One is constructed from any key encapsulation mechanism (KEM) meeting indistinguishability against chosen ciphertext attacks (IND-CCA security) and any data encapsulation mechanism (DEM) meeting both simulatability and one-time integrity of ciphertexts (OT-INT-CTXT security). The other is constructed from a KEM based on Fujisaki-Okamoto transformation [48, 64] and any message authentication code (MAC) meeting strong unforgeability against one-time chosen message attacks (sUF-OT-CMA security). We obtain concrete constructions of the above PKE schemes in the following way:

- Regarding the PKE scheme starting from a KEM and a DEM, it is possible to construct the concrete ones from existing IND-CCA secure KEM/PKE schemes resistant to quantum computing and standardized DEMs such as CTR-DEM and CCM-DEM. In particular, we can transform all KEM/PKE schemes submitted to the post-quantum cryptography (PQC) standardization project to SIM-SO-CCA secure PKE in the QICM by combining with the standardized DEM.

- Concerning the scheme starting from an FO-based KEM and a MAC, we can obtain the concrete ones by combining concrete FO-based KEM/PKE schemes submitted to the PQC standardization project and quantum-secure MACs. Notice that most submitted KEM constructions are categorized as FO-based KEM such as $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}^{\not\perp}_m$, $\mathsf{QFO}^{\not\perp}$, and $\mathsf{QFO}^{\not\perp}_m$, and standardized MACs such as NMAC/HMAC.

Therefore, it is possible to obtain concrete SIM-SO-CCA secure PKE schemes in the QICM or the QROM, by using existing practical (standardized) cryptographic primitives.

Second, we dealt with the quantum security of aggregate MACs (AMACs) and sequential aggregate MACs (SAMACs) for the first time. Regarding AMACs, we formalized the security of AMACs in the security model in which any adversary is allowed to issue quantum queries to tagging oracles. Our security definition is reasonable because it is the extension of the existing one [82] in the classical security model. Moreover, we proved that an existing generic construction [82] starting from any deterministic MAC satisfies our security if the underlying MAC scheme fulfills the quantum security formalized in [25]. Concerning SAMACs, we formalized the quantum security, which is the extension of the existing security definition [44] in the classical security model. And then, we showed that existing SAMACs [44, 128] are broken in our security model. We presented two generic constructions satisfying our security. One is constructed from any (deterministic) quantum-secure pseudorandom function (QPRF). The other is constructed from any randomized pseudorandom generator (randomized PRG) resistant to quantum computing.

Concrete constructions of our schemes are obtained as follows:

- The concrete SAMAC schemes from QPRFs are obtained by applying existing QPRFs [136, 124]. In particular, we can apply standardized MACs such as NMAC/HMAC because it was proven that these meet the quantum security of PRFs [124].

- The SAMAC schemes from randomized PRGs are concretely realized by applying PRGs of [135, 12]. Namely, it is possible to obtain SAMACs with the quantum security based on learning parity with noise (LPN) which is a well-known computationally hard problem even for quantum computers.

Third, we proposed signcryption schemes in the QROM or the standard model which is a model without random oracles and ideal ciphers. We presented two constructions satisfying both multi-user indistinguishability against insider chosen chiphertext attacks (MU-IND-iCCA security) and multi-user strong unforgeability against insider chosen message attacks (MU-sUF-iCMA security). One is a hybrid encryption from our lattice-based signcryption and a DEM scheme with indistinguishability against one-time attacks (IND-OT security) and *one-to-one* property. The other is a generic construction starting from any PKE scheme satisfying indistinguishability against chosen plaintext attacks (IND-CPA security) and any lossy identification scheme meeting several properties. We showed that the key-size and ciphertext-size of our schemes are shorter than those of existing ones which is constructed by applying concrete lattice-based primitives to existing generic constructions [34, 104].

Besides, concerning our schemes, the key-size and ciphertext-size of signcryption scheme in the QROM are shorter than those of the scheme in the standard model.

Concrete constructions of our signcryption schemes are obtained as follows:

- The concrete constructions of the lattice-based hybrid signcryption can be realized by combining our signcryption scheme in Section 5.3.1 and AES (Advanced Encryption Standard), which is a standardized symmetric key encryption, as an IND-OT secure DEM with *one-to-one* property.

- The concrete constructions of the generic construction secure in the QROM can be obtained by applying concrete IND-CPA secure PKE schemes and lossy identification schemes. We can apply lattice-based IND-CPA secure PKE constructions [113, 90] and lattice-based lossy identification schemes [95, 58, 84].

The scheme in the standard model is important in terms of security while the one in the QROM is also significant in terms of practicality. Focusing on security, we presented the lattice-based scheme in the standard model[1] because we do not assume that there exist ideal quantum random functions (quantum random oracles), and the standard model is stronger than the QROM. Although the security of cryptosystems in the (quantum) random oracle model is guaranteed under the strong assumption, cryptographic systems secure in this model are generally more efficient than those in the standard model in terms of key-size, ciphertext-size, and time-complexity. Actually, standardized public-key cryptosystems meet security in the random oracle model, and most ones submitted to the PQC standardization project also satisfy security in the QROM.

At present, researchers have presented designs of fundamental cryptographic primitives secure against quantum computing in classical security models. However, in the future, these ones do not necessarily guarantee security in a situation where quantum computers are widespread, and many users can use quantum computing. Hence, we dealt with security against quantum computing in such a situation, which is called quantum security in this thesis, and gave how to construct cryptosystems with quantum security of fundamental properties such as confidentiality and integrity. In the substantially distant future, the quantum security may not be sufficient since we have considered only quantum security of classical data. In a quantum world, it is natural to use not only classical data but also quantum one. Hence, it is necessary to consider the security of quantum data. Regarding the existing works of this security, there are security definitions of encryption [30, 6, 49]

[1] Although we proved that our lattice-based scheme satisfies both MU-IND-iCCA security and MU-sUF-iCMA security in the classical security model, this scheme also fulfills both security in the quantum security model. The security proofs are the same as the proofs in [25].

and MACs [51] in the quantum world. More concretely, Broadbent and Jeffery considered the security of encryption schemes with quantum algorithms, and formalized the indistinguishability of PKE and symmetric key encryption (SKE) [30]. Alagic et al. gave the simulation-based security of PKE/SKE with quantum algorithms [6]. On the other hand, Gagliardoni et al. considered the security model in which a quantum adversary gets the quantum superposition of ciphertexts of encryption with *classical* computations, and defined the indistinguishability/simulation-based security of encryption schemes with classical algorithms [49]. As for authentication, Garg et al. formalized the one-time unforgeability of MACs with quantum algorithms. Cryptosystems with the security have been proposed in [30, 6, 49, 43, 7, 29, 99, 51]. However, formalizing standard security such as IND-CCA security and unforgeability against (multi-time) chosen message attacks (UF-CMA security) on quantum data is an open problem because of the quantum no-cloning theorem. Therefore, the development of encryption/authentication schemes with the standard security in the quantum world is necessary and challenging as the future work of quantum-secure cryptography.

Bibliography

[1] M. Abdalla, P. Fouque, V. Lyubashevsky, and M. Tibouchi. Tightly secure signatures from lossy identification schemes. *J. Cryptology*, 29(3):597–631, 2016.

[2] M. Abdalla, C. Namprempre, and G. Neven. On the (im)possibility of blind message authentication codes. In *CT-RSA*, volume 3860 of *Lecture Notes in Computer Science*, pages 262–279. Springer, 2006.

[3] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, 2010.

[4] J. H. Ahn, M. Green, and S. Hohenberger. Synchronized aggregate signatures: new definitions, constructions and applications. In *ACM Conference on Computer and Communications Security*, pages 473–484. ACM, 2010.

[5] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108. ACM, 1996.

[6] G. Alagic, A. Broadbent, B. Fefferman, T. Gagliardoni, C. Schaffner, and M. S. Jules. Computational security of quantum encryption. In *ICITS*, volume 10015 of *Lecture Notes in Computer Science*, pages 47–71, 2016.

[7] G. Alagic, Y. Dulek, C. Schaffner, and F. Speelman. Quantum fully homomorphic encryption with verification. In *ASIACRYPT (1)*, volume 10624 of *Lecture Notes in Computer Science*, pages 438–467. Springer, 2017.

[8] G. Alagic and A. Russell. Quantum-secure symmetric-key cryptography based on hidden shifts. In *EUROCRYPT (3)*, volume 10212 of *Lecture Notes in Computer Science*, pages 65–93, 2017.

[9] A. Ambainis, M. Hamburg, and D. Unruh. Quantum security proofs using semi-classical oracles. In *CRYPTO (2)*, volume 11693 of *Lecture Notes in Computer Science*, pages 269–295. Springer, 2019.

[10] A. Ambainis, A. Rosmanis, and D. Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *FOCS*, pages 474–483. IEEE Computer Society, 2014.

[11] J. H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer, 2002.

[12] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009.

[13] J. Baek, R. Steinfeld, and Y. Zheng. Formal proofs for the security of signcryption. *J. Cryptology*, 20(2):203–235, 2007.

[14] M. Bellare, R. Dowsley, B. Waters, and S. Yilek. Standard security does not imply security against selective-opening. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 645–662. Springer, 2012.

[15] M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 1–35. Springer, 2009.

[16] M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining message authentication code. *J. Comput. Syst. Sci.*, 61(3):362–399, 2000.

[17] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *J. Cryptology*, 21(4):469–491, 2008.

[18] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.

[19] M. Bellare, B. Waters, and S. Yilek. Identity-based encryption secure against selective opening attack. In *TCC*, volume 6597 of *Lecture Notes in Computer Science*, pages 235–252. Springer, 2011.

[20] M. Bellare and S. Yilek. Encryption schemes secure under selective opening attack. *IACR Cryptology ePrint Archive*, 2009:101, 2009.

[21] F. Böhl, D. Hofheinz, T. Jager, J. Koch, and C. Striecks. Confined guessing: New signatures from standard assumptions. *J. Cryptology*, 28(1):176–208, 2015.

[22] D. Boneh, R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. *SIAM J. Comput.*, 36(5):1301–1328, 2007.

[23] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011.

[24] D. Boneh and M. Zhandry. Quantum-secure message authentication codes. In *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 592–608. Springer, 2013.

[25] D. Boneh and M. Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In *CRYPTO (2)*, volume 8043 of *Lecture Notes in Computer Science*, pages 361–379. Springer, 2013.

[26] X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 499–517. Springer, 2010.

[27] X. Boyen and Q. Li. Towards tightly secure lattice short signature and id-based encryption. In *ASIACRYPT 2016*, volume 10032 of *Lecture Notes in Computer Science*, pages 404–434, 2016.

[28] X. Boyen and Q. Li. All-but-many lossy trapdoor functions from lattices and applications. In *CRYPTO (3)*, volume 10403 of *Lecture Notes in Computer Science*, pages 298–331. Springer, 2017.

[29] Z. Brakerski. Quantum FHE (almost) as secure as classical. In *CRYPTO (3)*, volume 10993 of *Lecture Notes in Computer Science*, pages 67–95. Springer, 2018.

[30] A. Broadbent and S. Jeffery. Quantum homomorphic encryption for circuits of low t-gate complexity. In *CRYPTO (2)*, volume 9216 of *Lecture Notes in Computer Science*, pages 609–629. Springer, 2015.

[31] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. *J. Cryptology*, 25(4):601–639, 2012.

[32] D. Catalano and D. Fiore. Practical homomorphic macs for arithmetic circuits. In *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 336–352. Springer, 2013.

[33] L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone. Report on post-quantum cryptography. Technical report, NIST: National institute of standards and technology, 2016. `https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf`.

[34] D. Chiba, T. Matsuda, J. C. N. Schuldt, and K. Matsuura. Efficient generic constructions of signcryption with insider security in the multi-user setting. In *ACNS*, volume 6715 of *Lecture Notes in Computer Science*, pages 220–237, 2011.

[35] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.*, 33(1):167–226, 2003.

[36] B. David, R. Dowsley, and A. C. A. Nascimento. Universally composable oblivious transfer based on a variant of LPN. In *CANS*, volume 8813 of *Lecture Notes in Computer Science*, pages 143–158. Springer, 2014.

[37] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.

[38] Y. Dodis, E. Kiltz, K. Pietrzak, and D. Wichs. Message authentication, revisited. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 355–374. Springer, 2012.

[39] J. Don, S. Fehr, C. Majenz, and C. Schaffner. Security of the fiat-shamir transformation in the quantum random-oracle model. In *CRYPTO (2)*, volume 11693 of *Lecture Notes in Computer Science*, pages 356–383. Springer, 2019.

[40] N. Döttling, J. Müller-Quade, and A. C. A. Nascimento. IND-CCA secure cryptography based on a variant of the LPN problem. In *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 485–503. Springer, 2012.

[41] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal gaussians. In *CRYPTO 2013*, volume 8042 of *Lecture Notes in Computer Science*, pages 40–56. Springer, 2013.

[42] L. Ducas and D. Micciancio. Improved short lattice signatures in the standard model. In *CRYPTO 2014*, volume 8616 of *Lecture Notes in Computer Science*, pages 335–352. Springer, 2014.

[43] Y. Dulek, C. Schaffner, and F. Speelman. Quantum homomorphic encryption for polynomial-sized circuits. In *CRYPTO (3)*, volume 9816 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2016.

[44] O. Eikemeier, M. Fischlin, J. Götzmann, A. Lehmann, D. Schröder, P. Schröder, and D. Wagner. History-free aggregate message authentication codes. In *SCN*, volume 6280 of *Lecture Notes in Computer Science*, pages 309–328. Springer, 2010.

[45] S. Fehr, D. Hofheinz, E. Kiltz, and H. Wee. Encryption schemes secure against chosen-ciphertext selective opening attacks. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 381–402. Springer, 2010.

[46] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[47] D. Fiore, A. Mitrokotsa, L. Nizzardo, and E. Pagnin. Multi-key homomorphic authenticators. In *ASIACRYPT (2)*, volume 10032 of *Lecture Notes in Computer Science*, pages 499–530, 2016.

[48] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *J. Cryptology*, 26(1):80–101, 2013.

[49] T. Gagliardoni, A. Hülsing, and C. Schaffner. Semantic security and indistinguishability in the quantum world. In *CRYPTO (3)*, volume 9816 of *Lecture Notes in Computer Science*, pages 60–89. Springer, 2016.

[50] T. E. Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 1984.

[51] S. Garg, H. Yuen, and M. Zhandry. New security notions and feasibility results for authentication of quantum data. In *CRYPTO (2)*, volume 10402 of *Lecture Notes in Computer Science*, pages 342–371. Springer, 2017.

[52] R. Gennaro and D. Wichs. Fully homomorphic message authenticators. In *ASIACRYPT (2)*, volume 8270 of *Lecture Notes in Computer Science*, pages 301–320. Springer, 2013.

[53] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC 2008*, pages 197–206. ACM, 2008.

[54] O. Goldreich, S. Goldwasser, and S. Halevi. Collision-free hashing from lattice problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(42), 1996.

[55] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions (extended abstract). In *FOCS*, pages 464–479. IEEE Computer Society, 1984.

[56] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[57] L. K. Grover. A fast quantum mechanical algorithm for database search. In *STOC*, pages 212–219. ACM, 1996.

[58] T. Güneysu, V. Lyubashevsky, and T. Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES*, volume 7428 of *Lecture Notes in Computer Science*, pages 530–547. Springer, 2012.

[59] B. Hemenway, B. Libert, R. Ostrovsky, and D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In *ASIACRYPT*, volume 7073 of *Lecture Notes in Computer Science*, pages 70–88. Springer, 2011.

[60] F. Heuer, T. Jager, E. Kiltz, and S. Schäge. On the selective opening security of practical public-key encryption schemes. In *Public Key Cryptography*, volume 9020 of *Lecture Notes in Computer Science*, pages 27–51. Springer, 2015.

[61] F. Heuer and B. Poettering. Selective opening security from simulatable data encapsulation. In *ASIACRYPT (2)*, volume 10032 of *Lecture Notes in Computer Science*, pages 248–277, 2016.

[62] S. Hirose and H. Kuwakado. Forward-secure sequential aggregate message authentication revisited. In *ProvSec*, volume 8782 of *Lecture Notes in Computer Science*, pages 87–102. Springer, 2014.

[63] D. Hofheinz. All-but-many lossy trapdoor functions. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 209–227. Springer, 2012.

[64] D. Hofheinz, K. Hövelmanns, and E. Kiltz. A modular analysis of the fujisaki-okamoto transformation. In *TCC (1)*, volume 10677 of *Lecture Notes in Computer Science*, pages 341–371. Springer, 2017.

[65] D. Hofheinz, T. Jager, and A. Rupp. Public-key encryption with simulation-based selective-opening security and compact ciphertexts. In *TCC (B2)*, volume 9986 of *Lecture Notes in Computer Science*, pages 146–168, 2016.

[66] D. Hofheinz, V. Rao, and D. Wichs. Standard security does not imply indistinguishability under selective opening. In *TCC (B2)*, volume 9986 of *Lecture Notes in Computer Science*, pages 121–145, 2016.

[67] D. Hofheinz and A. Rupp. Standard versus selective opening security: Separation and equivalence results. In *TCC*, volume 8349 of *Lecture Notes in Computer Science*, pages 591–615. Springer, 2014.

[68] S. Hohenberger and B. Waters. Synchronized aggregate signatures from the RSA assumption. In *EUROCRYPT (2)*, volume 10821 of *Lecture Notes in Computer Science*, pages 197–229. Springer, 2018.

[69] A. Hosoyamada and K. Aoki. On quantum related-key attacks on iterated even-mansour ciphers. *IEICE Transactions*, 102-A(1):27–34, 2019.

[70] A. Hosoyamada and Y. Sasaki. Cryptanalysis against symmetric-key schemes with online classical queries and offline quantum computations. In *CT-RSA*, volume 10808 of *Lecture Notes in Computer Science*, pages 198–218. Springer, 2018.

[71] A. Hosoyamada and Y. Sasaki. Quantum demiric-selçuk meet-in-the-middle attacks: Applications to 6-round generic feistel constructions. In *SCN*, volume 11035 of *Lecture Notes in Computer Science*, pages 386–403. Springer, 2018.

[72] A. Hosoyamada and K. Yasuda. Building quantum-one-way functions from block ciphers: Davies-meyer and merkle-damgård constructions. In *ASIACRYPT (1)*, volume 11272 of *Lecture Notes in Computer Science*, pages 275–304. Springer, 2018.

[73] K. Hövelmanns, E. Kiltz, S. Schäge, and D. Unruh. Generic authenticated key exchange in the quantum random oracle model. *IACR Cryptology ePrint Archive*, 2018:928, 2018.

[74] A. Hülsing, J. Rijneveld, and F. Song. Mitigating multi-target attacks in hash-based signatures. In *Public Key Cryptography (1)*, volume 9614 of *Lecture Notes in Computer Science*, pages 387–416. Springer, 2016.

[75] G. Ito, A. Hosoyamada, R. Matsumoto, Y. Sasaki, and T. Iwata. Quantum chosen-ciphertext attacks against feistel ciphers. In *CT-RSA*, volume 11405 of *Lecture Notes in Computer Science*, pages 391–411. Springer, 2019.

[76] H. Jiang, Z. Zhang, L. Chen, H. Wang, and Z. Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In *CRYPTO (3)*, volume 10993 of *Lecture Notes in Computer Science*, pages 96–125. Springer, 2018.

[77] H. Jiang, Z. Zhang, and Z. Ma. Key encapsulation mechanism with explicit rejection in the quantum random oracle model. In *Public Key Cryptography (2)*, volume 11443 of *Lecture Notes in Computer Science*, pages 618–645. Springer, 2019.

[78] H. Jiang, Z. Zhang, and Z. Ma. Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model. In

PQCrypto, volume 11505 of *Lecture Notes in Computer Science*, pages 227–248. Springer, 2019.

[79] S. Jiang, K. A. Britt, A. J. McCaskey, T. S. Humble, and S. Kais. Quantum annealing for prime factorization. *Scientific Reports*, 8(17667), 2018.

[80] M. Kaplan, G. Leurent, A. Leverrier, and M. Naya-Plasencia. Breaking symmetric cryptosystems using quantum period finding. In *CRYPTO (2)*, volume 9815 of *Lecture Notes in Computer Science*, pages 207–237. Springer, 2016.

[81] S. Katsumata, S. Yamada, and T. Yamakawa. Tighter security proofs for GPV-IBE in the quantum random oracle model. In *ASIACRYPT (2)*, volume 11273 of *Lecture Notes in Computer Science*, pages 253–282. Springer, 2018.

[82] J. Katz and A. Y. Lindell. Aggregate message authentication codes. In *CT-RSA*, volume 4964 of *Lecture Notes in Computer Science*, pages 155–169. Springer, 2008.

[83] A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT 2008*, volume 5350 of *Lecture Notes in Computer Science*, pages 372–389. Springer, 2008.

[84] E. Kiltz, V. Lyubashevsky, and C. Schaffner. A concrete treatment of fiat-shamir signatures in the quantum random-oracle model. In *EUROCRYPT (3)*, volume 10822 of *Lecture Notes in Computer Science*, pages 552–586. Springer, 2018.

[85] E. Kiltz, D. Masny, and K. Pietrzak. Simple chosen-ciphertext security from low-noise LPN. In *Public Key Cryptography*, volume 8383 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2014.

[86] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.

[87] J. Lai, R. H. Deng, S. Liu, J. Weng, and Y. Zhao. Identity-based encryption secure against selective opening chosen-ciphertext attack. In *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 77–92. Springer, 2014.

[88] B. Libert and J. Quisquater. Efficient signcryption with key privacy from Gap Diffie-Hellman groups. In *PKC 2004*, volume 2947 of *Lecture Notes in Computer Science*, pages 187–200. Springer, 2004.

[89] B. Libert, A. Sakzad, D. Stehlé, and R. Steinfeld. All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In *CRYPTO (3)*, volume 10403 of *Lecture Notes in Computer Science*, pages 332–364. Springer, 2017.

[90] R. Lindner and C. Peikert. Better key sizes (and attacks) for lwe-based encryption. In *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 319–339. Springer, 2011.

[91] Q. Liu and M. Zhandry. Revisiting post-quantum fiat-shamir. In *CRYPTO (2)*, volume 11693 of *Lecture Notes in Computer Science*, pages 326–355. Springer, 2019.

[92] S. Liu and K. G. Paterson. Simulation-based selective opening CCA security for PKE from key encapsulation mechanisms. In *Public Key Cryptography*, volume 9020 of *Lecture Notes in Computer Science*, pages 3–26. Springer, 2015.

[93] L. Lyu, S. Liu, S. Han, and D. Gu. Tightly SIM-SO-CCA secure public key encryption from standard assumptions. In *Public Key Cryptography (1)*, volume 10769 of *Lecture Notes in Computer Science*, pages 62–92. Springer, 2018.

[94] V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Public Key Cryptography*, volume 4939 of *Lecture Notes in Computer Science*, pages 162–179. Springer, 2008.

[95] V. Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616. Springer, 2009.

[96] V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer, 2012.

[97] V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 37–54. Springer, 2008.

[98] V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. *J. Cryptology*, 31(3):774–797, 2018.

[99] U. Mahadev. Classical homomorphic encryption for quantum circuits. In *FOCS*, pages 332–338. IEEE Computer Society, 2018.

[100] T. Matsuda, K. Matsuura, and J. C. N. Schuldt. Efficient constructions of signcryption schemes and signcryption composability. In *INDOCRYPT*, volume 5922 of *Lecture Notes in Computer Science*, pages 321–342. Springer, 2009.

[101] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, 2012.

[102] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007.

[103] V. S. Miller. Use of elliptic curves in cryptography. In *CRYPTO*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, 1985.

[104] R. Nakano and J. Shikata. Constructions of signcryption in the multi-user setting from identity-based encryption. In *IMACC 2013*, volume 8308 of *Lecture Notes in Computer Science*, pages 324–343. Springer, 2013.

[105] C. Namprempre, G. Neven, and M. Abdalla. A study of blind message authentication codes. *IEICE Transactions*, 90-A(1):75–82, 2007.

[106] NIST: National institute of standards and technology. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf`.

[107] NIST: National institute of standards and technology. Post-quantum cryptography standardzation, 2017. available at: `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization`.

[108] T. Okamoto, K. Tanaka, and S. Uchiyama. Quantum public-key cryptosystems. In *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, pages 147–165. Springer, 2000.

[109] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.

[110] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, pages 333–342. ACM, 2009.

[111] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, 2008.

[112] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *STOC 2008*, pages 187–196. ACM, 2008.

[113] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.

[114] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[115] M. Rückert. Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles. In *PQCrypto 2010*, volume 6061 of *Lecture Notes in Computer Science*, pages 182–200. Springer, 2010.

[116] T. Saito, K. Xagawa, and T. Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *EUROCRYPT (3)*, volume 10822 of *Lecture Notes in Computer Science*, pages 520–551. Springer, 2018.

[117] S. Sato and J. Shikata. Lattice-based signcryption without random oracles. In *PQCrypto*, volume 10786 of *Lecture Notes in Computer Science*, pages 331–351. Springer, 2018.

[118] S. Sato and J. Shikata. Signcryption with quantum random oracles. In *ProvSec*, volume 11192 of *Lecture Notes in Computer Science*, pages 406–414. Springer, 2018.

[119] S. Sato and J. Shikata. Quantum-secure (non-)sequential aggregate message authentication codes. In *IMACC*, volume 11929 of *Lecture Notes in Computer Science*, pages 295–316. Springer, 2019.

[120] S. Sato and J. Shikata. SO-CCA secure PKE in the quantum random oracle model or the quantum ideal cipher model. In *IMACC*, volume 11929 of *Lecture Notes in Computer Science*, pages 317–341. Springer, 2019.

[121] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of IEEE Symposium on Foundations of Computer Science*, pages 124–134. IEEE, 1994.

[122] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[123] D. R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, 1997.

[124] F. Song and A. Yun. Quantum security of NMAC and related constructions - PRF domain extension against quantum attacks. In *CRYPTO (2)*, volume 10402 of *Lecture Notes in Computer Science*, pages 283–309. Springer, 2017.

[125] D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 617–635. Springer, 2009.

[126] C. H. Tan. Signcryption scheme in multi-user setting without random oracles. In *IWSEC 2008*, volume 5312 of *Lecture Notes in Computer Science*, pages 64–82. Springer, 2008.

[127] E. E. Targhi and D. Unruh. Post-quantum security of the fujisaki-okamoto and OAEP transforms. In *TCC (B2)*, volume 9986 of *Lecture Notes in Computer Science*, pages 192–216, 2016.

[128] S. Tomita, Y. Watanabe, and J. Shikata. Sequential aggregate authentication codes with information theoretic security. In *CISS*, pages 192–197. IEEE, 2016.

[129] D. Unruh. Revocable quantum timed-release encryption. *J. ACM*, 62(6):49:1–49:76, 2015.

[130] D. Unruh. Computationally binding quantum commitments. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 497–527. Springer, 2016.

[131] D. Unruh. Post-quantum security of fiat-shamir. In *ASIACRYPT (1)*, volume 10624 of *Lecture Notes in Computer Science*, pages 65–95. Springer, 2017.

[132] M. N. Wegman and L. Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.

[133] S. Yamada. Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In *EUROCRYPT 2016*, volume 9666 of *Lecture Notes in Computer Science*, pages 32–62. Springer, 2016.

[134] S. Yamada. Asymptotically compact adaptively secure lattice ibes and verifiable random functions via generalized partitioning techniques. In *CRYPTO (3)*, volume 10403 of *Lecture Notes in Computer Science*, pages 161–193. Springer, 2017.

[135] Y. Yu and J. P. Steinberger. Pseudorandom functions in almost constant depth from low-noise LPN. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 154–183. Springer, 2016.

[136] M. Zhandry. How to construct quantum random functions. In *FOCS*, pages 679–687. IEEE Computer Society, 2012.

[137] M. Zhandry. Secure identity-based encryption in the quantum random oracle model. In *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 758–775. Springer, 2012.

[138] M. Zhandry. A note on the quantum collision and set equality problems. *Quantum Information and Computation*, 15(7& 8), 2015.

[139] J. Zhang, Y. Chen, and Z. Zhang. Programmable hash functions from lattices: Short signatures and ibes with small key sizes. In *CRYPTO 2016*, volume 9816 of *Lecture Notes in Computer Science*, pages 303–332. Springer, 2016.

[140] Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In *CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer, 1997.

List of Publications

Peer-Reviewed Journal Article and Conference Papers

Related to The Thesis:

1. S. Sato and J. Shikata, "SO-CCA Secure PKE in the Quantum Random Oracle Model or the Quantum Ideal Cipher Model," IMA International Conference on Cryptography and Coding, LNCS 11929, pp.317–341, Springer, 2019 (see also [120]).

2. S. Sato and J. Shikata, "Quantum-Secure (Non-)Sequential Aggregate Message Authentication Codes," IMA International Conference on Cryptography and Coding, LNCS 11929, pp.295–316, Springer, 2019 (see also [119]).

3. S. Sato and J. Shikata, "Lattice-based Signcryption without Random Oracles," PQCrypto, LNCS 10786, pp.331–351, Springer, 2018 (see also [117]).

4. S. Sato and J. Shikata, "Signcryption with Quantum Random Oracles," Provable Security, LNCS 11192, pp.406–414, Springer, 2018 (see also [118]).

Other Publications:

5. S. Sato, S. Hirose, and J. Shikata, "Sequential Aggregate MACs from Any MACs Revisited," Network and System Security, LNCS 11928, pp.387–407, Springer, 2019.

6. S. Sato and J. Shikata, "Interactive Aggregate Message Authentication Scheme with Detecting Functionality, " Advanced Information Networking and Applications, vol.926, pp.1316–1328, Springer, 2019.

7. S. Sato, S. Hirose, and J. Shikata, "Sequential Aggregate MACs from Any MACs: Aggregation and Detecting Functionality," Journal of Internet Services and Information Security, vol.9, No.1, pp.2–23, 2019.

8. S. Sato, S. Hirose, and J. Shikata, "Generic Construction of Sequential Aggregate MACs from Any MACs.," Provable Security 2018, LNCS 10786, pp.295–312, Springer, 2018.

Non Peer-Reviewed Papers

9. L. Cao, S. Sato, and J. Shikata, "A Remark on Improving (Sequential) Aggregate Message Authentication Codes with Detecting Functionality," Technical Committee on Information Security (ISEC) IEICE Tech. Rep., vol.119, No.474, ISEC2019-107, pp.143–150, 2020 (in Japanese).

10. T. Miyazawa, S. Sato, and J. Shikata, "Semi-Adaptively Secure Inner-Product Encryption from Lattices," Tech. Rep. of Computer Security Group (CSEC), 2019-CSEC-87(1), pp.1–8, 2019 (in Japanese).

11. S. Sato and J. Shikata, "Post-Quantum Sequential Aggregate Message Authentication Codes," Proc. of Symposium on Cryptography and Information Security (SCIS 2019), 3B1-5, 2019 (in Japanese).

12. S. Sato and J. Shikata, "Interactive Aggregate Message Authentication Scheme with Detecting Functionality," Proc. of Symposium on Cryptography and Information Security (SCIS 2019), 3A3-1, 2019 (in Japanese).

13. J. Shikata, T. Uchikoshi, M. Ebina, S. Sato, Y. Masuda, Y. Unagami, and T. Takazoe, "Security Proof of a Device Authentication Protocol for HEMS," Tech. Rep. of Computer Security Group (CSEC), 2018-CSEC-80(7), pp.1–8, 2018 (in Japanese).

14. S. Sato, S. Hirose, and J. Shikata, "A Generic Construction of Sequential Aggregate Message Authentication Codes," Proc. of Symposium on Cryptography and Information Security (SCIS 2018), 2C3-1, 2018 (in Japanese).

15. T. Mikasa, S. Sato, and J. Shikata, "Interactive Aggregate Message Authentication Codes with Tracing: Model and Construction," Proc. of Symposium on Cryptography and Information Security (SCIS 2018), 2C3-2, 2018 (in Japanese).

16. J. Shikata, T. Uchikoshi, M. Ebina, S. Sato, Y. Masuda, Y. Unagami, and T. Takazoe, "Security Proof of a Device Authentication Protocol for HEMS," Proc. of Symposium on Cryptography and Information Security (SCIS 2018), 3E1-4, 2018 (in Japanese).

17. K. Onuma, S. Sato, and J. Shikata, "Lattice-based Multiple Encryption," Proc. of Symposium on Cryptography and Information Security (SCIS 2018), 1A2-3, 2018 (in Japanese).

18. S. Sato and J. Shikata, “Construction of Signcryption in the Quantum Random Oracle Model,” Proc. of Symposium on Cryptography and Information Security (SCIS 2017), 2F4-2, 2017 (in Jpanese).

19. S. Sato and J. Shikata, “Lattice-based Signcryption, Revisited,” Proc. of Computer Security Symposium 2016 (CSS 2016), 1C3-3, 2016 (in Japanese).

20. S. Sato and J. Shikata, “Lattice-based Signcryption without Random Oracles,” Proc. of Symposium on Cryptography and Information Security (SCIS 2016), 1D1-1, 2016 (in Japanese).