Quantum Public-Key Encryption with Tamper-Resilient Public Keys from One-Way Functions

Fuyuki Kitagawa[1], Tomoyuki Morimae[2], Ryo Nishimaki[1],Takashi Yamakawa[1,2]

[1]NTT Social Informatics Laboratories, Tokyo, Japan
{fuyuki.kitagawa,ryo.nishimaki,takashi.yamakawa}@ntt.com
[2]Yukawa Institute for Theoretical Physics, Kyoto University, Kyoto, Japan
tomoyuki.morimae@yukawa.kyoto-u.ac.jp

Abstract

We construct quantum public-key encryption from one-way functions. In our construction, public keys are quantum, but ciphertexts are classical. Quantum public-key encryption from one-way functions (or weaker primitives such as pseudorandom function-like states) are also proposed in some recent works [Morimae-Yamakawa, eprint:2022/1336; Coladangelo, eprint:2023/282; Barooti-Grilo-Malavolta-Sattath-Vu-Walter, eprint:2023/877]. However, they have a huge drawback: they are secure only when quantum public keys can be transmitted to the sender (who runs the encryption algorithm) without being tampered with by the adversary, which seems to require unsatisfactory physical setup assumptions such as secure quantum channels. Our construction is free from such a drawback: it guarantees the secrecy of the encrypted messages even if we assume only unauthenticated quantum channels. Thus, the encryption is done with adversarially tampered quantum public keys. Our construction is the first quantum public-key encryption that achieves the goal of classical public-key encryption, namely, to establish secure communication over insecure channels, based only on one-way functions. Moreover, we show a generic compiler to upgrade security against chosen plaintext attacks (CPA security) into security against chosen ciphertext attacks (CCA security) only using one-way functions. As a result, we obtain CCA secure quantum public-key encryption based only on one-way functions.

1 Introduction

1.1 Background

Quantum physics provides several advantages in cryptography. For instance, statistically-secure key exchange, which is impossible in classical cryptography, becomes possible if quantum states are transmitted [BB84]. Additionally, oblivious transfers and multiparty computations are possible only from one-way functions (OWFs) in the quantum world [BCKM21, GLSV21]. Those cryptographic primitives are believed to require stronger structured assumptions in classical cryptography [IR89, GKM$^+$00]. Furthermore, it has been shown that several cryptographic tasks, such as (non-interactive) commitments, digital signatures, secret-key encryption, quantum money, and multiparty computations, are possible based on new primitives such as pseudorandom states generators, pseudorandom function-like states generators, one-way states generators, and EFI, which seem to be weaker than OWFs [JLS18, Kre21, MY22b, AQY22, BCQ23, AGQY22, CX22, MY22a, KQST22].

Quantum public key encryption from OWFs. Despite these developments, it is still an open problem whether public-key encryption (PKE) is possible with only OWFs (or the above weaker primitives) in the quantum world. PKE from OWFs is impossible (in a black-box way) in the classical cryptography [IR90]. However, it could be possible if quantum states are transmitted or local operations are quantum. In fact, some recent works [MY22a, Col23, BGH$^+$23] independently constructed quantum PKE (QPKE) with quantum public keys based on OWFs or pseudorandom function-like states generators. However, the constructions proposed in those works have a huge drawback as explained below, and thus we still do not have a satisfactory solution to the problem of "QPKE from OWFs".

How to certify quantum public keys? When we study public key cryptographic primitives, we have to care about how to certify the public keys, that is, how a sender (who encrypts messages) can check if a given public key is a valid public key under which the secrecy of the encrypted messages is guaranteed. When the public keys are classical strings, we can easily certify them using digital signature schemes. However, in the case where the public keys are quantum states, we cannot use digital signature schemes to achieve this goal in general[1], and it is unclear how to certify them.

As stated above, some recent works [MY22a, Col23, BGH$^+$23] realized QPKE with quantum public keys from OWFs or even weaker assumptions. However, those works did not tackle this quantum public key certification problem very much. In fact, as far as we understand, to use the primitives proposed in those works meaningfully, we need to use secure quantum channels to transmit the quantum public keys so that a sender can use an intact quantum public key. This is a huge drawback since the goal of PKE is to transmit a message *without assuming secure channels*. If the sender can establish a secure channel to obtain the quantum public key, the sender could use it to transmit the message in the first place, and there is no advantage to using the PKE scheme.

QPKE with tamper-resilient quantum public keys. One of our goals in this work is to solve this issue and develop a more reasonable notion of QPKE with quantum public keys. Especially, we consider the setting with the following two natural conditions. First, we assume that every quantum state (that is, quantum public keys in this work) is sent via an unauthenticated channel, and thus it can be tampered with by an adversary. If we do not assume secure quantum channels, we have to take such a tampering attack into account since authentication generally requires secrecy for quantum channels [BCG$^+$02]. Second, we assume that

[1] There is a general impossibility result for signing quantum states [AGM21].

every classical string is sent via an authenticated channel. This is the same assumption in classical PKE and can be achieved using digital signatures. Note that the security of the constructions proposed in the above works [MY22a, Col23, BGH+23] is broken in this natural setting. In this work, we tackle whether we can realize QPKE with quantum public keys that provides a security guarantee in this natural setting, especially from OWFs.

1.2 Our Results

We affirmatively answer the above question. We realize the first QPKE scheme based only on OWFs that achieves the goal of classical PKE, which is to establish secure communication over insecure channels. We define the notions of QPKE that can be used in the above setting where unauthenticated quantum channels and classical authenticated channels are available. Then, we propose constructions satisfying the definitions from OWFs. Below, we state each result in detail.

Definitional work. We redefine the syntax of QPKE. The difference from the previous definitions is that the key generation algorithm outputs a classical verification key together with the secret key. Also, this verification key is given to the encryption algorithm with a quantum public key and a message so that the encryption algorithm can check the validity of the given quantum public key. We require ciphertexts to be classical.[2] We require a QPKE scheme to satisfy the following two basic security notions.

- **Indistinguishability against public key tempering chosen plaintext attacks (IND-pkT-CPA security).** Roughly speaking, it guarantees that indistinguishability holds even if messages are encrypted by a public key tampered with by an adversary. More specifically, it guarantees that no efficient adversary can guess the challenge bit b with a probability significantly better than random guessing given $\mathsf{Enc}(\mathsf{vk}, \mathsf{pk}', \mathsf{msg}_b)$, where vk is the correct verification key and $(\mathsf{pk}', \mathsf{msg}_0, \mathsf{msg}_1)$ are generated by the adversary who is given the verification key vk and multiple copies of the correctly generated quantum public keys.[3] IND-pkT-CPA security captures the setting where the classical verification key is sent via a classical authenticated channel. Thus, everyone can obtain the correct verification key. However, a quantum public key is sent via an unauthenticated quantum channel and thus can be tampered with by an adversary.

- **Decryption error detectability.** In our setting, an adversary may try to cause a decryption error by tampering with the quantum public key. To address this issue, we introduce a security notion that we call *decryption error detectability*. It roughly guarantees that a legitimate receiver of a ciphertext can notice if the decrypted message is different from the message intended by the sender.

IND-pkT-CPA security considers adversaries that may tamper with quantum public keys but only passively observe ciphertext. For classical PKE, the golden standard security notion is indistinguishability against chosen ciphertext attacks (IND-CCA security) that considers active adversaries that may see decryption results of any (possibly malformed) ciphertexts. Thus, we also define its analog for QPKE. In Section 1.3, we discuss its importance in a natural application scenario.

- **Indistinguishability against public key tempering chosen ciphertext attacks (IND-pkT-CCA security).** This is similar to IND-pkT-CPA security except that the adversary is given access to the

[2] We could also consider QPKE schemes with quantum ciphertexts if we only consider IND-pkT-CPA security. However, it is unclear how we should define IND-pkT-CCA security for such schemes because the decryption oracle cannot check if a given ciphertext is equivalent to the challenge ciphertext. Thus, we focus on schemes with classical ciphertexts in this paper.

[3] The tampered quantum public key pk' can be entangled with the adversary's internal state.

decryption oracle that returns a decryption result on any ciphertext other than the challenge ciphertext.[4] Moreover, we allow the adversary to learn one-bit information indicating if the challenge ciphertext is decrypted to $\perp$ or not. We note that it is redundant for classical PKE since the challenge ciphertext is always decrypted to the challenge message, which is not $\perp$, by decryption correctness. On the other hand, it may give more power to the adversary for QPKE since if the adversary tempers with the public key that is used to generate the challenge ciphertext, decryption correctness may no longer hold.

IND-pkT-CPA secure construction from OWFs. We propose a QPKE scheme satisfying IND-pkT-CPA security from a digital signature scheme that can be constructed from OWFs. Our construction is inspired by the duality between distinguishing and swapping shown by Aaronson, Atia, and Susskind [AAS20] and its cryptographic applications by Hhan, Morimae, and Yamakawa [HMY22]. Our construction has quantum public keys and classical ciphertexts. We also propose a general transformation that adds decryption error detectability. The transformation uses only a digital signature scheme.

Upgrading to IND-pkT-CCA security. We show a generic compiler that upgrades IND-pkT-CPA security into IND-pkT-CCA security while preserving decryption error detectability only using OWFs. It is worth mentioning that constructing such a generic CPA-to-CCA compiler is a long-standing open problem for classical PKE, and thus we make crucial use of the fact that public keys are quantum for constructing our compiler. By plugging our IND-pkT-CPA secure construction into the compiler, we obtain a QPKE scheme that satisfies IND-pkT-CCA security and decryption error detectability only based on OWFs.

Recyclable variant. Our above definitions for QPKE assume each quantum public key is used to encrypt only a single message and might be consumed. We also introduce a notion of *recyclable QPKE* where the encryption algorithm given a quantum public key outputs a ciphertext together with a classical state that can be used to encrypt a message many times. Then, we show that any standard IND-pkT-CPA (resp. IND-pkT-CCA) secure QPKE scheme with classical ciphertexts can be transformed into a recyclable one with IND-pkT-CPA (resp. IND-pkT-CCA) security while preserving decryption error detectability. The transformation uses only a CPA (resp. CCA) secure classical symmetric key encryption scheme that is implied by OWFs. Thus, by combining the transformation with the above results, we obtain a recyclable IND-pkT-CCA QPKE scheme with decryption error detectability from OWFs.

1.3 Discussion

Pure State Public Keys vs. Mixed State Public Keys. The quantum public keys of our QPKE schemes are mixed states. Some recent works [Col23, BGH$^+$23] that studied QPKE explicitly require that a quantum public key of QPKE be a pure quantum state. The reason is related to the quantum public key certification problem, which is this work's main focus. Barooti et al. [BGH$^+$23] claimed that a sender can check the validity of given quantum public keys by using SWAP test if they are pure states, but not if they are mixed states. However, as far as we understand, this claim implicitly requires that at least one intact quantum public key be transmitted via secure quantum channels where an adversary cannot touch it at all[5], which is an unsatisfactory assumption that makes QPKE less valuable. It is unclear how a sender can check the validity

[4] Recall that ciphertexts are classical in our definition, and thus this is well-defined.

[5] More precisely, their model seems to require a physical setup assumption that enables a sender to obtain at least one intact quantum public key, such as secure quantum channels or tamper-proof quantum hardware.

of a given quantum public key in the constructions proposed in [BGH+23] without assuming such secure transmission of intact quantum public keys.

We believe that it is not important whether the quantum public keys are pure states or mixed states, and what is really important is whether a sender can check the validity of given quantum public keys without assuming unsatisfactory setups such as quantum secure channels. Although our QPKE schemes have mixed state quantum public keys, they provide such a validity checking of quantum public keys by a sender without assuming any unsatisfactory setups. In addition, we can easily extend our construction into one with pure state quantum public keys. We provide the variant in Appendix A.

Necessity of IND-pkT-CCA security. Here, we argue that IND-pkT-CCA security is crucial in a natural application scenario of QPKE. Suppose that Bob generates many copies of his quantum public key along with the corresponding verification key and publishes them on his website on a quantum internet. Alice downloads the public and verification keys, first checks the validity of the public key by using the verification key and then encrypts a message to Bob. When Bob receives a ciphertext from Alice, he decrypts it, and if the decryption result is $\bot$, he judges that a decryption error was caused by a tampering attack on the quantum public key and asks Alice to encrypt the message again by using another copy of the quantum public key. We observe that IND-pkT-CPA security does not imply security in this setting. The reason is that an adversary could send any ciphertext to Bob to know if it is decrypted to be $\bot$ or not, which is not captured by IND-pkT-CPA security. On the other hand, IND-pkT-CCA security implies security in the above setting since it allows the adversary to learn if a ciphertext (including the challenge ciphertext) is decrypted to be $\bot$.

2 Technical Overview

We provide a technical overview of our work.

2.1 Definition of QPKE

Syntax. We define QPKE that can be used in the setting where quantum unauthenticated channels and classical authenticated channels are available. To this end, we introduce the following two modifications to the previous definitions.

- The secret key generation algorithm outputs a classical verification key together with the secret key.
- The verification key is given to the encryption algorithm together with a quantum public key and a message so that the encryption algorithm can check the validity of the given quantum public key.

Concretely, in our definition, a QPKE scheme consists of four algorithms $(\mathsf{SKGen}, \mathsf{PKGen}, \mathsf{Enc}, \mathsf{Dec})$. SKGen is a classical secret key generation algorithm that is given the security parameter and outputs a classical secret key sk and a classical verification key vk. PKGen is a quantum public key generation algorithm that takes as input the classical secret key sk and outputs a quantum public key pk. Enc is a quantum encryption algorithm that takes as inputs the classical verification key vk, a quantum public key pk, and a plaintext msg, and outputs a classical ciphertext ct. Finally, Dec is a classical decryption algorithm that takes as input the classical secret key and a ciphertext, and outputs the decryption result.

The above definitions for QPKE assume each quantum public key is used to encrypt only a single message and might be consumed. We also introduce a notion of recyclable QPKE where the encryption algorithm given a quantum public key outputs a ciphertext together with a classical state that can be used to encrypt a message many times. In this overview, we mainly focus on non-recyclable QPKE for simplicity.

IND-pkT-CPA security. IND-pkT-CPA security roughly guarantees that indistinguishability holds even if messages are encrypted by a public key pk' tampered with by an adversary as long as the encryption is done with the correct verification key vk. Formally, IND-pkT-CPA security is defined using the following security experiment played by an adversary $\mathcal{A}$. The experiment first generates classical secret key and verification key pair $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SKGen}(1^\lambda)$ and m copies of the quantum public key $\mathsf{pk}_1, \ldots, \mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})$. Then, $\mathcal{A}$ is given the classical verification key vk and m quantum public keys $\mathsf{pk}_1, \ldots, \mathsf{pk}_m$, and outputs a tampered quantum public key pk' and a pair of challenge plaintexts $(\mathsf{msg}_0, \mathsf{msg}_1)$. The experiment generates the challenge ciphertext using the adversarially generated quantum public key, that is, $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{vk}, \mathsf{pk}', \mathsf{msg}_b)$, where $b \leftarrow \{0,1\}$. Finally, $\mathcal{A}$ is given ct^* and outputs the guess for b. IND-pkT-CPA security guarantees that any efficient quantum adversary cannot guess b significantly better than random guessing in this experiment.

IND-pkT-CPA security captures the setting where the classical verification key is sent via a classical authenticated channel and thus everyone can obtain correct verification key, but a quantum public key is sent via an unauthenticated quantum channel and thus can be tampered with by an adversary. Especially, it captures an adversary $\mathcal{A}$ who steals a quantum public key pk sent to a user, replace it with a tampered one pk', and try to break the secrecy of a message encrypted by pk'.

To capture wide range of usage scenarios, we give multiple copies of the quantum public keys pk_1, ..., pk_m to $\mathcal{A}$. We also consider a relaxed security notion where an adversary is given a single quantum public key and denote it as IND-pkT-CPA$^{(1)}$.

Decryption error detectability. We also define a security notion related to the correctness notion that we call decryption error detectability. It roughly guarantees that a legitimate receiver of a ciphertext can notice if the decrypted message is different from the message intended by the sender. Such a decryption error could occur frequently in our setting as a result of the tampering attacks on the quantum public key sent via an unauthenticated quantum channel. Note that our definition of QPKE requires a ciphertext of QPKE be a classical string and we assume every classical information is sent though a classical authenticated channel. Thus, similarly to the verification key, we can assume that ciphertexts can be sent without being tampered.

2.2 IND-pkT-CPA Secure Construction

We provide the technical overview for IND-pkT-CPA secure construction.

Duality between distinguishing and swapping. Our construction is inspired by the duality between distinguishing and swapping shown by Aaronson, Atia, and Susskind [AAS20] and its cryptographic applications by Hhan, Morimae, and Yamakawa [HMY22].[6] We first review their idea. Let $|\psi\rangle$ and $|\phi\rangle$ be orthogonal states. [AAS20] showed that $|\psi\rangle + |\phi\rangle$ and $|\psi\rangle - |\phi\rangle$ are computationally indistinguishable[7] if and only if one cannot efficiently "swap" $|\psi\rangle$ and $|\phi\rangle$ with a non-negligible advantage, i.e., for any efficiently computable unitary U, $|\langle\phi| U |\psi\rangle + \langle\psi| U |\phi\rangle|$ is negligible. Based on the above result, [HMY22] suggested to use $|\psi\rangle + (-1)^b |\phi\rangle$ as an encryption of a plaintext $b \in \{0,1\}$. By the result of [AAS20], its security is reduced to the hardness of swapping $|\psi\rangle$ and $|\phi\rangle$.

Basic one-time SKE. We can construct one-time SKE scheme with quantum ciphertext using the above duality between distinguishing and swapping as follows. A secret decryption key is (x_0, x_1) for uniformly

[6] In the main body, we do not explicitly use any result of [AAS20, HMY22] though our analysis is similar to theirs.

[7] We often omit normalization factors.

random bit strings $x_0, x_1 \in \{0,1\}^\lambda$, and the corresponding secret encryption key is

$$|0\rangle |x_0\rangle + |1\rangle |x_1\rangle . \tag{1}$$

Then, when encrypting a plaintext $b \in \{0,1\}$, it transforms the secret encryption key into the ciphertext

$$|0\rangle |x_0\rangle + (-1)^b |1\rangle |x_1\rangle . \tag{2}$$

One-time indistinguishability of this scheme is somewhat obvious because the adversary has no information of x_0 or x_1 besides the ciphertext, but let us analyze it using the idea of [AAS20] to get more insights. Suppose that the above scheme is insecure, i.e., $|0\rangle |x_0\rangle + |1\rangle |x_1\rangle$ and $|0\rangle |x_0\rangle - |1\rangle |x_1\rangle$ are computationally distinguishable with a non-negligible advantage. Then, by the result of [AAS20], there is an efficient unitary U that swaps $|0\rangle |x_0\rangle$ and $|1\rangle |x_1\rangle$ with a non-negligible advantage. By using this unitary, let us consider the following procedure:

1. Given a state $|0\rangle |x_0\rangle \pm |1\rangle |x_1\rangle$, measure it in the computational basis to get $|\alpha\rangle |x_\alpha\rangle$ for random $\alpha \in \{0,1\}$.

2. Apply the unitary U to $|\alpha\rangle |x_\alpha\rangle$ and measure it in the computational basis.

Since U swaps $|0\rangle |x_0\rangle$ and $|1\rangle |x_1\rangle$ with a non-negligible advantage, the probability that the outcome of the second measurement is $|\alpha \oplus 1\rangle |x_{\alpha\oplus 1}\rangle$ is non-negligible. This yields the following observation: If one can efficiently distinguish $|0\rangle |x_0\rangle + |1\rangle |x_1\rangle$ and $|0\rangle |x_0\rangle - |1\rangle |x_1\rangle$, then one can efficiently compute both x_0 and x_1 from $|0\rangle |x_0\rangle \pm |1\rangle |x_1\rangle$. On the other hand, it is easy to show that one cannot compute both x_0 and x_1 from $|0\rangle |x_0\rangle \pm |1\rangle |x_1\rangle$ with a non-negligible probability by a simple information theoretical argument. Thus, the above argument implies one-time indistinguishability of the above construction.

Extension to IND-pkT-CPA$^{(1)}$ secure QPKE with quantum ciphertext. We show how to extend the above SKE scheme into an IND-pkT-CPA$^{(1)}$ secure QPKE scheme with quantum ciphertext. One natural approach is to use the secret encryption key $|0\rangle |x_0\rangle + |1\rangle |x_1\rangle$ as a quantum public key. However, it does not work since the adversary for IND-pkT-CPA$^{(1)}$ who is given $|0\rangle |x_0\rangle + |1\rangle |x_1\rangle$ as the public key can replace it with $|0\rangle |x'_0\rangle + |1\rangle |x'_1\rangle$ for x'_0, x'_1 of its choice. To fix this issue, we partially authenticate a quantum public key by using classical digital signatures. Concretely, the secret key generation algorithm SKGen generates a signing key and verification key pair $(\mathsf{sk}, \mathsf{vk})$ of a digital signature scheme, and use them as the secret key and verification key of the QPKE scheme. Then, public key generation algorithm PKGen takes as input sk and outputs a quantum public key

$$|0\rangle |\sigma(0)\rangle + |1\rangle |\sigma(1)\rangle, \tag{3}$$

where $\sigma(\alpha)$ is a signature for $\alpha \in \{0,1\}$ by the signing key sk. Here, we assume that the signature scheme has a deterministic signing algorithm. The encryption algorithm Enc that is given vk, a quantum public key, and a plaintext $b \in \{0,1\}$ first coherently verifies using vk the validity of the signatures in the second register of the public key and aborts if the verification rejects. Otherwise, Enc generates the ciphertext by encoding the plaintext b into the phase of the quantum public key as before.

The IND-pkT-CPA$^{(1)}$ security of the construction is analyzed as follows. We assume that the digital signature scheme satisfies strong unforgeability, i.e., given message-signature pairs $(\mathsf{msg}_1, \sigma_1), ..., (\mathsf{msg}_n, \sigma_n)$, no efficient adversary can output $(\mathsf{msg}^*, \sigma^*)$ such that $(\mathsf{msg}^*, \sigma^*) \neq (\mathsf{msg}_i, \sigma_i)$ for all $i \in [n]$.[8] Then, no

[8] At this point, two-time security (where $n = 2$) suffices but we finally need to allow n to be an arbitrary polynomial.

matter how the adversary who is given a single correctly generated quantum public key tampers with it, if it passes the verification in Enc, the state after passing the verification is negligibly close to a state in the form of

$$c_0 \left|0\right\rangle \left|\sigma(0)\right\rangle \left|\Psi_0\right\rangle + c_1 \left|1\right\rangle \left|\sigma(1)\right\rangle \left|\Psi_1\right\rangle \tag{4}$$

with some complex coefficients c_0 and c_1, and some states $\left|\Psi_0\right\rangle$ and $\left|\Psi_1\right\rangle$ over the adversary's register (except for a negligible probability). The encryption of a plaintext $b \in \{0,1\}$ is to apply Z^b on the first qubit of Equation (4). The cipertext generated under the tampered public key is therefore

$$c_0 \left|0\right\rangle \left|\sigma(0)\right\rangle \left|\Psi_0\right\rangle + (-1)^b c_1 \left|1\right\rangle \left|\sigma(1)\right\rangle \left|\Psi_1\right\rangle . \tag{5}$$

By a slight extension of the analysis of the above SKE scheme, we show that if one can efficiently distinguish $c_0 \left|0\right\rangle \left|\sigma(0)\right\rangle \left|\Psi_0\right\rangle + c_1 \left|1\right\rangle \left|\sigma(1)\right\rangle \left|\Psi_1\right\rangle$ and $c_0 \left|0\right\rangle \left|\sigma(0)\right\rangle \left|\Psi_0\right\rangle - c_1 \left|1\right\rangle \left|\sigma(1)\right\rangle \left|\Psi_1\right\rangle$, then one can efficiently compute both $\sigma(0)$ and $\sigma(1)$. On the other hand, recall that the adversary is only given one copy of the public key $\left|0\right\rangle \left|\sigma(0)\right\rangle + \left|1\right\rangle \left|\sigma(1)\right\rangle$. We can show that it is impossible to compute both $\sigma(0)$ and $\sigma(1)$ from this state by the strong unforgeability as follows. By [BZ13, Lemma 2.1], the probability to output both $\sigma(0)$ and $\sigma(1)$ is only halved even if $\left|0\right\rangle \left|\sigma(0)\right\rangle + \left|1\right\rangle \left|\sigma(1)\right\rangle$ is measured in the computational basis before given to the adversary. After the measurement, the adversary's input collapses to a classical state $\left|\alpha\right\rangle \left|\sigma(\alpha)\right\rangle$ for random $\alpha \in \{0,1\}$, in which case the adversary can output $\sigma(\alpha \oplus 1)$ only with a negligible probability by the strong unforgeability. Combining the above, security of the above scheme under tampered public keys is proven.

Achiving IND-pkT-CPA security. The above QPKE scheme satisfies IND-pkT-CPA$^{(1)}$ security, but does not satisfy IND-pkT-CPA security where the adversary is given multiple copies of quantum public keys. If the adversary is given two copies of the quantum public key, by measuring each public key in the computational basis, the adversary can learn both $\sigma(0)$ and $\sigma(1)$ with probability 1/2. In order to extend the scheme into IND-pkT-CPA security, we introduce a classical randomness for each public key generation. Specifically, a public key is

$$(r, \left|0\right\rangle \left|\sigma(0,r)\right\rangle + \left|1\right\rangle \left|\sigma(1,r)\right\rangle) \tag{6}$$

where $r \in \{0,1\}^\lambda$ is chosen uniformly at random for every execution of the public key generation algorithm, and $\sigma(\alpha, r)$ is a signature for $\alpha \| r$.[9] An encryption of a plaintext $b \in \{0,1\}$ is

$$(r, \left|0\right\rangle \left|\sigma(0,r)\right\rangle + (-1)^b \left|1\right\rangle \left|\sigma(1,r)\right\rangle). \tag{7}$$

Since each quantum public key uses different r, security of this scheme holds even if the adversary obtains arbitrarily many public keys.

Making ciphertext classical. The above constructions has quantum ciphertext, but our definition explicitly requires that a QPKE scheme have a classical cipheretxt. We observe that the ciphertext of the above schemes can be made classical easily. In the IND-pkT-CPA secure construction, the ciphertext contains a quantum state $\left|0\right\rangle \left|\sigma(r,0)\right\rangle + (-1)^b \left|1\right\rangle \left|\sigma(r,1)\right\rangle$. Suppose that we measure this state in Hadamard basis and let d be the measurement outcome. Then an easy calculation shows that we have

$$b = d \cdot (0\|\sigma(0,r) \oplus 1\|\sigma(1,r)). \tag{8}$$

Thus, sending (r, d) as a ciphertext is sufficient for the receiver who has the decryption key to recover the plaintext b. Moreover, this variant is at least as secure as the original one with quantum ciphertexts since the Hadamard-basis measurement only loses information of the ciphertext.

[9] $\alpha \| r$ is the concatenation of two bit strings α and r.

Achieving recyclability. Given that we achieve classical ciphertext property, it is rather straightforward to transform the construction into recyclable one where the encryption algorithm outputs a classical state that can be used to encrypt many plaintexts. The transformation uses standard hybrid encryption technique. Concretely, the encryption algorithm first generates a key K of a SKE scheme, encrypt each bit of K by the above non-recyclable scheme in a bit-by-bit manner, and encrypt the plaintext msg by the symmetric key encryption scheme under the key K. The final ciphertext is $(\mathsf{ct}, \mathsf{ct}_{\mathsf{ske}})$, where ct is the ciphertext of K by the non-recyclable scheme and $\mathsf{ct}_{\mathsf{ske}}$ is the ciphertext of msg by the SKE scheme. The encryption algorithm outputs a classical state (ct, K) together with the ciphertext. The encryptor can reuse the state when it encrypts another message later.[10]

Adding decryption error detectability. So far, we are only concerned with IND-pkTA security. On the other hand, the schemes presented in the previous paragraphs do not satisfy decryption error detectability. (See Definition 5.2 for formal definition.) Fortunately, there is a simple generic conversion that adds decryption error detectability while preserving IND-pkTA security by using digital signatures. The idea is that the encryption algorithm first generates a signature for the message under a signing key generated by itself, encrypts both the original message and signature under the building block scheme, and outputs the ciphertexts along with the verification key for the signature scheme in the clear. Then, the decryption algorithm can verify that the decryption result is correct as long as it is a valid message-signature pair (except for a negligible probability).

2.3 CPA-to-CCA Transformation

We now explain how to transform IND-pkT-CPA secure QPKE scheme into IND-pkT-CCA secure one using OWFs.

Definition of IND-pkT-CCA security. IND-pkT-CCA security is defined by adding the following two modifications to the security experiment for IND-pkT-CPA security.

- Throughout the experiment, the adversary can get access to the decryption oracle that is given a ciphertext ct and returns $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ if $\mathsf{ct} \neq \mathsf{ct}^*$ and $\perp$ otherwise.
- The adversary is given the 1-bit leakage information that the challenge ciphertext is decrypted to $\perp$ or not.

As discussed in Section 1.3, the second modification is needed to support a natural usage scenario of QPKE. For simplicity, we will ignore this second modification for now and proceed the overview as if IND-pkT-CCA security is defined by just adding the decryption oracle to the security experiment for IND-pkT-CPA security.

We also define a weaker variant of IND-pkT-CCA security where the adversary is allowed to make only a single query to the decryption oracle. We denote it as IND-pkT-1CCA security. We consider a relaxed variant of IND-pkT-CCA security and IND-pkT-1CCA security where the adversary is given only a single copy of quantum public key. We denote them as IND-pkT-CCA$^{(1)}$ security and IND-pkT-1CCA$^{(1)}$ security respectively, similarly to IND-pkT-CPA$^{(1)}$ security.

[10] The idea to achieve the recyclability by the hybrid encryption technique was also used in one of the constructions in [BGH$^+$23].

IND-pkT-1CCA from IND-pkT-CPA. In classical cryptography, the CCA security where the number of decryption query is a-priori bounded to q is called q-bounded-CCA security. It is known that any CPA secure classical PKE scheme can be transformed into q-bounded-CCA secure one using only a digital signature scheme [CHH+07]. We show that by using a similar technique, we can transform an IND-pkT-CPA secure QPKE scheme into an IND-pkT-1CCA secure one using only a digital signature scheme.

Boosting 1-bounded-CCA into full-fledged CCA. Classically, it is not known how to boost bounded-CCA security into CCA security without using additional assumption, and as a result, "general transformation from CPA to CCA" is a major open question in classical public key cryptography. Surprisingly, we show that 1-bounded-CCA security can be boosted into CCA security for QPKE assuming only OWFs. More specifically, we show that IND-pkT-1CCA$^{(1)}$ secure QPKE can be transformed into IND-pkT-CCA$^{(1)}$ secure one assuming only OWFs.

The key component in the transformation is tokenized message authentication code (MAC) [BSS21]. Tokenized MAC is a special MAC scheme where we can generate a quantum MAC token using the secret MAC key. The quantum MAC token can be used to generate a valid signature only once. In other words, an adversary who is given a single quantum MAC token cannot generate valid signatures for two different messages. Tokenized MAC can be realized using only OWFs [BSS21].

The high level idea is to design CCA secure scheme so that a public key contains quantum MAC token and an adversary can generate a valid ciphertext only when it consumes the MAC token, which ensures that the adversary can make only one meaningful decryption query and CCA security is reduced to 1-bounded-CCA security. Consider the following construction of a QPKE scheme CCA based on IND-pkT-1CCA$^{(1)}$ secure QPKE scheme $\mathsf{1CCA}$ and tokenized MAC scheme TMAC. The secret key of CCA consists of the secret keys of $\mathsf{1CCA}$ and TMAC, and the verification key of CCA is that of $\mathsf{1CCA}$. A quantum public key of CCA consists of that of $\mathsf{1CCA}$ and a MAC token of TMAC. The encryption algorithm of CCA first generates a ciphertext $\mathsf{1cca.ct}$ of $\mathsf{1CCA}$ and then generates a signature $\mathsf{tmac}.\sigma$ for the message $\mathsf{1cca.ct}$ by consuming the MAC token contained in the public key. The resulting ciphertext is $(\mathsf{1cca.ct}, \mathsf{tmac}.\sigma)$. The decryption algorithm of CCA that is given the ciphertext $(\mathsf{1cca.ct}, \mathsf{tmac}.\sigma)$ first checks validity of $\mathsf{tmac}.\sigma$ by using the secret MAC key included in the secret key. If it passes, the decryption algorithm decrypts $\mathsf{1cca.ct}$ by using the secret key of $\mathsf{1CCA}$.

In the experiment of IND-pkT-CCA$^{(1)}$ security for CCA, we can ensure that an adversary can make at most one decryption query whose result is not $\perp$ by the power of TMAC, as we want. However, the adversary in fact can make one critical query $(\mathsf{1cca.ct}^*, \mathsf{tmac}.\sigma')$, where $\mathsf{1cca.ct}^*$ is the first component of the challenge ciphertext, which allows the adversary to obtain the challenge bit. This attack is possible due to the fact that the adversary is allowed to tamper the quantum public key.[11] Fortunately, this attack can be prevented by using a digital signature scheme and tying the two components $\mathsf{1cca.ct}^*$ and $\mathsf{tmac}.\sigma'$ together. Once this issue is fixed, we can successfully reduce the IND-pkT-CCA$^{(1)}$ security of the construction to the IND-pkT-1CCA$^{(1)}$ security of $\mathsf{1CCA}$, since now the adversary can make only one non-critical decryption query.

[11] More specifically, the attack is done as follows. The adversary is given a quantum public key $(\mathsf{1cca.pk}, \mathsf{token})$ where $\mathsf{1cca.pk}$ is a public key of $\mathsf{1CCA}$ and token is a MAC token of TMAC. The adversary generates another token token' of TMAC by itself and sends $(\mathsf{1cca.pk}, \mathsf{token}')$ to the challenger as the tempered public key. Since there is no validity check on the MAC token in the encryption algorithm, this tampered public key is not rejected and the challenge ciphertext $(\mathsf{1cca.ct}^*, \mathsf{tmac}.\sigma^*)$ is generated. Given the challenge ciphertext, the adversary generates a signature $\mathsf{tmac}.\sigma'$ for $\mathsf{1cca.ct}^*$ using token contained in the given un-tampered public key and queries $(\mathsf{1cca.ct}^*, \mathsf{tmac}.\sigma')$ to the decryption oracle. Since $\mathsf{tmac}.\sigma'$ is a valid signature generated using the correct token, this query is successful and the adversary obtains the challenge bit.

Upgrading IND-pkT-CCA$^{(1)}$ to IND-pkT-CCA. We can easily transform an IND-pkT-CCA$^{(1)}$ secure QPKE scheme into an IND-pkT-CCA secure one. The transformation is somewhat similar to the one from IND-pkT-CPA$^{(1)}$ secure scheme to IND-pkT-CPA secure one. We bundle multiple instances of IND-pkT-CCA$^{(1)}$ secure scheme each of which is labeled by a classical random string. The transformation uses pseudorandom functions and digital signatures both of which are implied by OWFs.

How to deal with 1-bit leakage "the challenge is decrypted to $\bot$ or not". So far, we ignore the fact that our definition of IND-pkT-CCA security allows the adversary to obtain 1-bit leakage information whether the challenge is decrypted to $\bot$ or not. We introduce an intermediate notion between IND-pkT-CPA security and IND-pkT-CCA security that we call IND-pkT-CVA security where the adversary is given the 1-bit leakage information but is not allowed to get access to the decryption oracle. We then show that an IND-pkT-CPA secure QPKE scheme can be transformed into IND-pkT-CVA secure one using the cut-and-choose technique. Moreover, we show that the above construction strategy towards IND-pkT-CCA secure construction works even if the adversaries are given the 1-bit leakage information, if we start with IND-pkT-CVA secure scheme.

Some Remarks. We finally provide some remarks.

Recyclability: Similarly to IND-pkT-CPA secure scheme, we consider recyclable variant for IND-pkT-CCA secure one. We show that a recyclable IND-pkT-CCA secure QPKE scheme can be constructed from non-recyclable one using the hybrid encryption technique similarly to IND-pkT-CPA secure construction.

Strong decryption error detectability: In the proof of the construction from IND-pkT-1CCA$^{(1)}$ secure scheme to IND-pkT-CCA$^{(1)}$ secure one, we use the underlying scheme's decryption error detectability. The proof of CCA security is sensitive to decryption errors, and it turns out that decryption error detectability that only provides security guarantee against computationally bounded adversaries is not sufficient for this part. Thus, we introduce statistical variant of decryption error detectability that we call strong decryption error detectability. We also prove that our IND-pkT-CVA secure construction based on the cut-and-choose technique achieves strong decryption error detectability, and the subsequent transformations preserve it.

3 Related Works and Open Problems

3.1 Related Works

The possibility that QPKE can be achieved from weaker assumptions was first pointed out by Gottesman [Got], though he did not give any concrete construction. The first concrete construction of QPKE was proposed by Kawachi, Koshiba, Nishimura, and Yamakami [KKNY05]. They formally defined the notion of QPKE with quantum public keys, and provided a construction satisfying it from a distinguishing problem of two quantum states. Recently, Morimae and Yamakawa [MY22a] pointed out that QPKE defined by [KKNY05] can be achieved from any classical or quantum symmetric key encryption almost trivially. The constructions proposed in these two works have mixed state quantum public keys. Then, subsequent works [Col23, BGH$^+$23] independently studied the question whether QPKE with pure state quantum public keys can be constructed from OWFs or even weaker assumptions.

The definition of QPKE studied in the above works essentially assume that a sender can obtain intact quantum public keys. As far as we understand, this requires unsatisfactory physical setup assumptions such as secure quantum channels or tamper-proof quantum hardware, regardless of whether the quantum public keys are pure states or mixed states. In our natural setting where an adversary can touch the quantum channel where quantum public keys are sent, the adversary can easily attack the previous constructions by simply replacing the quantum public key on the channel with the one generated by itself that the adversary knows the corresponding secret key. We need to take such adversarial behavior into consideration, unless we assume physical setup assumptions that deliver intact quantum public keys to the sender. Our work is the first one that proposes a QPKE scheme secure in this natural setting assuming only classical authenticated channels that is the same assumption as classical PKE and can be implemented by digital signature schemes. It is unclear if we could solve the problem in the previous constructions by using classical authenticated channels similarly to our work. Below, we review the constructions of QPKE from OWFs proposed in the recent works.

The construction by Morimae and Yamakawa [MY22a] is highly simple. A (mixed state) public key of their construction is of the form $(\mathsf{ct}_0, \mathsf{ct}_1)$, where ct_b is an encryption of b by a symmetric key encryption scheme. The encryption algorithm with input message b simply outputs ct_b.

Coladangelo [Col23] constructed a QPKE scheme with quantum public keys and quantum ciphertexts from pseudorandom functions (PRFs), which are constructed from OWFs. The public key is

$$|\mathsf{pk}\rangle \coloneqq \sum_y (-1)^{\mathsf{PRF}_k(y)} |y\rangle\,, \tag{9}$$

and the secret key is k. The ciphertext for the plaintext m is

$$(Z^x |\mathsf{pk}\rangle = \sum_y (-1)^{x\cdot y+\mathsf{PRF}_k(y)} |y\rangle\,, r, r\cdot x \oplus m), \tag{10}$$

where r is chosen uniformly at random.

Barooti, Grilo, Huguenin-Dumittan, Malavolta, Sattath, Vu, and Walter [BGH$^+$23] constructed three QPKE schemes: (1) CCA secure QPKE with quantum public keys and classical ciphertexts from OWFs (2) CCA1[12] secure QPKE with quantum public keys and ciphertexts from pseudorandom function-like states generators, (3) CPA secure QPKE with quantum public keys and classical ciphertexts from pseudo-random function-like states with proof of destruction. All constructions considers security under the encryption oracle. We review their construction based on OWFs.

Their construction is hybrid encryption of CPA secure QPKE (the KEM part) and CCA secure classical symmetric key encryption (the DEM part). The public key is

$$|\mathsf{pk}\rangle \coloneqq \sum_x |x\rangle\, |\mathsf{PRF}_k(x)\rangle\,, \tag{11}$$

and the secret key is k. The encryption algorithm first measures $|\mathsf{pk}\rangle$ in the computational basis to get $(x, \mathsf{PRF}_k(x))$ and outputs $(x, \mathsf{SKE.Enc}(\mathsf{PRF}_k(x), m))$ as the ciphertext for the plaintext m, where $\mathsf{SKE.Enc}$ is the encryption algorithm of a symmetric key encryption scheme.

We finally compare Quantum Key Distribution (QKD) [BB84] with our notion of QPKE. QKD also enables us to establish secure communication over an untrusted quantum channel assuming that an authenticated classical channel is available similarly to our QPKE. An advantage of QKD is that it is information theoretically secure and does not need any computational assumption. On the other hand, it has disadvantages that it must be interactive and parties must record secret information for each session. Thus, it is incomparable to the notion of QPKE.

[12] Afther the adversary received a challenge ciphertext, they cannot access the decryption oracle.

3.2 Concurrent Work

A concurrent and independent work by Malavolta and Walter [MW23] constructs a two-round quantum key exchange protocol from OWFs. Their underlying idea is similar to our IND-pkT-CPA secure construction. Indeed, the technical core of their work is a construction a QPKE scheme that is secure against adversaries that only see one copy of the quantum public key. A nice feature of their scheme is that it satisfies everlasting security. That is, as long as the adversary is quantum polynomial-time when tampering with the public key, it cannot recover any information of the encrypted message even if it has an unbounded computational power later. They also show how to extend the scheme to satisfy security in the many-copy setting at the cost of sacrificing everlasting security. This gives an alternative construction of IND-pkT-CPA secure QPKE scheme from OWFs using our terminology. On the other hand, they do not consider CCA security, and our CPA-to-CCA compiler is unique to this work.

3.3 Open Problems

In our construction, public keys are quantum states. It is an open problem whether QPKE with classical public keys are possible from OWFs. Another interesting open problem is whether we can construct QPKE defined in this work from an even weaker assumption than OWFs such as pseudorandom states generators.

In our model of QPKE, a decryption error may be caused by tampering attacks on the quantum public key. To address this issue, we introduce the security notion we call decryption error detectability that guarantees that a legitimate receiver of a ciphertext can notice if the decrypted message is different from the message intended by the sender. We could consider even stronger variant of decryption error detectability that requires that a sender can notice if a given quantum public key does not provide decryption correctness. It is an open problem to construct a QPKE scheme satisfying such a stronger decryption error detectability.

4 Preliminaries

4.1 Basic Notations

We use the standard notations of quantum computing and cryptography. We use λ as the security parameter. For any set S, $x \leftarrow S$ means that an element x is sampled uniformly at random from the set S. We write negl to mean a negligible function. PPT stands for (classical) probabilistic polynomial-time and QPT stands for quantum polynomial-time. For an algorithm A, $y \leftarrow A(x)$ means that the algorithm A outputs y on input x. For two bit strings x and y, $x\|y$ means the concatenation of them. For simplicity, we sometimes omit the normalization factor of a quantum state. (For example, we write $\frac{1}{\sqrt{2}}(|x_0\rangle + |x_1\rangle)$ just as $|x_0\rangle + |x_1\rangle$.) $I := |0\rangle\langle 0| + |1\rangle\langle 1|$ is the two-dimensional identity operator. For the notational simplicity, we sometimes write $I^{\otimes n}$ just as I when the dimension is clear from the context.

4.2 Digital Signatures

Definition 4.1 (Digital signatures). *A digital signature scheme is a set of algorithms* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *such that*

- $\mathsf{Gen}(1^\lambda) \to (k, \mathsf{vk})$: *It is a PPT algorithm that, on input the security parameter* λ*, outputs a signing key* k *and a verification key* vk.
- $\mathsf{Sign}(k, \mathsf{msg}) \to \sigma$: *It is a PPT algorithm that, on input the message* msg *and* k*, outputs a signature* σ.

- $\mathsf{Ver}(\mathsf{vk}, \mathsf{msg}, \sigma) \to \top/\bot$: *It is a deterministic classical polynomial-time algorithm that, on input* vk, msg*, and* σ*, outputs* $\top/\bot$.

We require the following correctness and strong EUF-CMA security.

Correctness: *For any* msg,

$$\Pr[\top \leftarrow \mathsf{Ver}(\mathsf{vk}, \mathsf{msg}, \sigma) : (k, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda), \sigma \leftarrow \mathsf{Sign}(k, \mathsf{msg})] \geq 1 - \mathsf{negl}(\lambda). \tag{12}$$

Strong EUF-CMA security: *For any QPT adversary* $\mathcal{A}$ *with classical oracle access to the signing oracle* $\mathsf{Sign}(k, \cdot)$*,*

$$\Pr[(\mathsf{msg}^*, \sigma^*) \notin \mathcal{Q} \wedge \top \leftarrow \mathsf{Ver}(\mathsf{vk}, \mathsf{msg}^*, \sigma^*) : (k, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda), (\mathsf{msg}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(k,\cdot)}(\mathsf{vk})] \leq \mathsf{negl}(\lambda), \tag{13}$$

where $\mathcal{Q}$ *is the set of message-signature pairs returned by the signing oracle.*

Remark 4.2. Without loss of generality, we can assume that Sign is deterministic. (The random seed used for Sign can be generated by applying a PRF on the message signed, and the key of PRF is appended to the signing key.)

Theorem 4.3 ([Gol04, Sec. 6.5.2]). *Strong EUF-CMA secure digital signatures exist if OWFs exist.*

4.3 Pseudorandom Functions

Definition 4.4 (Pseudorandom functions (PRFs)). *A keyed function* $\{\mathsf{PRF}_K : \mathcal{X} \to \mathcal{Y}\}_{K \in \{0,1\}^\lambda}$ *that is computable in classical deterministic polynomial-time is a quantum-query secure pseudorandom function if for any QPT adversary* $\mathcal{A}$ *with quantum access to the evaluation oracle* $\mathsf{PRF}_K(\cdot)$*,*

$$|\Pr[1 \leftarrow \mathcal{A}^{\mathsf{PRF}_K(\cdot)}(1^\lambda)] - \Pr[1 \leftarrow \mathcal{A}^{H(\cdot)}(1^\lambda)]| \leq \mathsf{negl}(\lambda), \tag{14}$$

where $K \leftarrow \{0,1\}^\lambda$ *and* $H : \mathcal{X} \to \mathcal{Y}$ *is a function chosen uniformly at random.*

As we can see, we consider PRFs that is secure even if an adversary can get access to the oracles in superposition, which is called quantum-query secure PRFs. We use the term PRFs to indicate quantum-query secure PRFs in this work.

Theorem 4.5 ([Zha12]). *(Quantum-query secure) PRFs exist if OWFs exist.*

4.4 Symmetric Key Encryption

Definition 4.6 (Symmetric Key Encryption (SKE)). *A (classical) symmetric key encryption (SKE) scheme with message space* $\{0,1\}^\ell$ *is a set of algorithms* $(\mathsf{Enc}, \mathsf{Dec})$ *such that*

- $\mathsf{Enc}(K, \mathsf{msg}) \to \mathsf{ct}$: *It is a PPT algorithm that, on input* $K \in \{0,1\}^\lambda$ *and the message* $\mathsf{msg} \in \{0,1\}^\ell$*, outputs a ciphertext* ct.
- $\mathsf{Dec}(K, \mathsf{ct}) \to \mathsf{msg}'$: *It is a deterministic classical polynomial-time algorithm that, on input* K *and* ct*, outputs* msg'.

We require the following correctness.

Correctness: *For any* $\mathsf{msg} \in \{0,1\}^\ell$,

$$\Pr[\mathsf{msg} \leftarrow \mathsf{Dec}(K, \mathsf{ct}) : K \leftarrow \{0,1\}^\lambda, \mathsf{ct} \leftarrow \mathsf{Enc}(K, \mathsf{msg})] = 1. \tag{15}$$

Definition 4.7 (IND-CPA Security). *For any QPT adversary $\mathcal{A}$ with classical oracle access to the encryption oracle* $\mathsf{Enc}(K, \cdot)$*,*

$$\Pr\left[b \leftarrow \mathcal{A}(\mathsf{ct}^*, \mathsf{st})^{\mathsf{Enc}(K,\cdot)} : \begin{array}{r} K \leftarrow \{0,1\}^\lambda \\ (\mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}^{\mathsf{Enc}(K,\cdot)}(1^\lambda) \\ b \leftarrow \{0,1\} \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(K, \mathsf{msg}_b) \end{array} \right] \leq \frac{1}{2} + \mathsf{negl}(\lambda). \tag{16}$$

Theorem 4.8 ([GGM86, HILL99]). *IND-CPA secure SKE exists if OWFs exist.*

Definition 4.9 (IND-CCA Security). *For any QPT adversary $\mathcal{A}$ with classical oracle access to the encryption oracle* $\mathsf{Enc}(K, \cdot)$*,*

$$\Pr\left[b \leftarrow \mathcal{A}(\mathsf{ct}^*, \mathsf{st})^{\mathsf{Enc}(K,\cdot), O_{\mathsf{Dec},2}(\cdot)} : \begin{array}{r} K \leftarrow \{0,1\}^\lambda \\ (\mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}^{\mathsf{Enc}(K,\cdot), O_{\mathsf{Dec},1}(\cdot)}(1^\lambda) \\ b \leftarrow \{0,1\} \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(K, \mathsf{msg}_b) \end{array} \right] \leq \frac{1}{2} + \mathsf{negl}(\lambda). \tag{17}$$

Here, $O_{\mathsf{Dec},1}(\mathsf{ct})$ *returns* $\mathsf{Dec}(K, \mathsf{ct})$ *for any* ct*.* $O_{\mathsf{Dec},2}$ *behaves identically to* $O_{\mathsf{Dec},1}$ *except that* $O_{\mathsf{Dec},2}$ *returns $\perp$ to the input* ct^**.*

Theorem 4.10 ([BN08]). *IND-CCA secure SKE exists if OWFs exist.*

4.5 Lemma by Boneh and Zhandry

In this paper, we use the following lemma by Boneh and Zhandry [BZ13].

Lemma 4.11 ([BZ13, Lemma 2.1]). *Let A be a quantum algorithm, and let $\Pr[x]$ be the probability that A outputs x. Let A' be another quantum algorithm obtained from A by pausing A at an arbitrary stage of execution, performing a partial measurement that obtains one of k outcomes, and then resuming A. Let $\Pr'[x]$ be the probability that A' outputs x. Then $\Pr'[x] \geq \Pr[x]/k$.*

5 Definition of QPKE

In this section, we define QPKE.

Definition 5.1 (Quantum Public-Key Encryption (QPKE)). *A quantum public-key encryption scheme with message space $\{0,1\}^\ell$ is a set of algorithms* $(\mathsf{SKGen}, \mathsf{PKGen}, \mathsf{Enc}, \mathsf{Dec})$ *such that*

- $\mathsf{SKGen}(1^\lambda) \to (\mathsf{sk}, \mathsf{vk})$: *It is a PPT algorithm that, on input the security parameter λ, outputs a classical secret key* sk *and a classical verification key* vk*.*
- $\mathsf{PKGen}(\mathsf{sk}) \to \mathsf{pk}$: *It is a QPT algorithm that, on input* sk*, outputs a quantum public key* pk*.*
- $\mathsf{Enc}(\mathsf{vk}, \mathsf{pk}, \mathsf{msg}) \to \mathsf{ct}$: *It is a QPT algorithm that, on input* vk*,* pk*, and a plaintext* $\mathsf{msg} \in \{0,1\}^\ell$*, outputs a classical ciphertext* ct*.*
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to \mathsf{msg}'$: *It is a classical deterministic polynomial-time algorithm that, on input* sk *and* ct*, outputs* $\mathsf{msg}' \in \{0,1\}^\ell \cup \{\perp\}$*.*

We require the following correctness and IND-pkTA security.

Correctness: *For any* $\mathsf{msg} \in \{0,1\}^\ell$*,*

$$\Pr[\mathsf{msg} \leftarrow \mathsf{Dec}(\mathsf{sk},\mathsf{ct}) : (\mathsf{sk},\mathsf{vk}) \leftarrow \mathsf{SKGen}(1^\lambda), \mathsf{pk} \leftarrow \mathsf{PKGen}(\mathsf{sk}), \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{vk},\mathsf{pk},\mathsf{msg})] \geq 1-\mathsf{negl}(\lambda). \tag{18}$$

IND-pkT-CPA Security: *For any polynomial m, and any QPT adversary $\mathcal{A}$,*

$$\Pr\left[b \leftarrow \mathcal{A}(\mathsf{ct}^*,\mathsf{st}) : \begin{array}{r} (\mathsf{sk},\mathsf{vk}) \leftarrow \mathsf{SKGen}(1^\lambda) \\ \mathsf{pk}_1,...,\mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m} \\ (\mathsf{pk}',\mathsf{msg}_0,\mathsf{msg}_1,\mathsf{st}) \leftarrow \mathcal{A}(\mathsf{vk},\mathsf{pk}_1,...,\mathsf{pk}_m) \\ b \leftarrow \{0,1\} \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{vk},\mathsf{pk}',\mathsf{msg}_b) \end{array} \right] \leq \frac{1}{2} + \mathsf{negl}(\lambda). \tag{19}$$

Here, $\mathsf{pk}_1,...,\mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m}$ *means that* PKGen *is executed m times and* pk_i *is the output of the ith execution of* PKGen. st *is a quantum internal state of $\mathcal{A}$, which can be entangled with* pk'.

As we discussed in Section 1.3, the above definition does not require the quantum public key pk to be a pure state.

We also define a security notion related to the correctness notion that we call decryption error detectability.

Definition 5.2 (Decryption error detectability). *We say that a QPKE scheme has decryption error detectability if for any polynomial m, and any QPT adversary $\mathcal{A}$,*

$$\Pr\left[\mathsf{msg}' \neq \bot \ \wedge\ \mathsf{msg}' \neq \mathsf{msg} : \begin{array}{r} (\mathsf{sk},\mathsf{vk}) \leftarrow \mathsf{SKGen}(1^\lambda) \\ \mathsf{pk}_1,...,\mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m} \\ (\mathsf{pk}',\mathsf{msg}) \leftarrow \mathcal{A}(\mathsf{vk},\mathsf{pk}_1,...,\mathsf{pk}_m) \\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{vk},\mathsf{pk}',\mathsf{msg}) \\ \mathsf{msg}' \leftarrow \mathsf{Dec}(\mathsf{sk},\mathsf{ct}) \end{array} \right] \leq \mathsf{negl}(\lambda). \tag{20}$$

It is easy to see that we can generically add decryption error detectability by letting the sender generate a signature for the message under a signing key generated by itself, encrypt the concatenation of the message and signature, and send the ciphertext along with the verification key of the signature to the receiver. The receiver can check that there is no decryption error (except for a negligible probability) if the decryption result is a valid message-signature pair. That is, we have the following theorem.

Theorem 5.3. *If there exist OWFs and a QPKE scheme that satisfies correctness and IND-pkT-CPA security, there exists a QPKE scheme that satisfies correctness, IND-pkT-CPA security, and decryption error detectability.*

We omit the proof since it is straightforward by the construction explained above. Since we have this theorem, we focus on constructing QPKE that satisfies correctness and IND-pkT-CPA security.

6 Construction of QPKE

In this section, we construct a QPKE scheme that satisfies correctness and IND-pkT-CPA security (but not decryption error detectability) from strong EUF-CMA secure digital signatures. The message space of our construction is $\{0,1\}$, but it can be extended to be arbitrarily many bits by parallel repetition. Let $(\mathsf{Gen},\mathsf{Sign},\mathsf{Ver})$ be a strong EUF-CMA secure digital signature scheme with a deterministic Sign algorithm and message space $\{0,1\}^u$ for $u = \omega(\log\lambda)$.

Our construction of QPKE is as follows.

- $\mathsf{SKGen}(1^\lambda) \to (\mathsf{sk}, \mathsf{vk})$: Run $(k, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda)$. Output $\mathsf{sk} := k$. Output vk.
- $\mathsf{PKGen}(\mathsf{sk}) \to \mathsf{pk}$: Parse $\mathsf{sk} = k$. Choose $r \leftarrow \{0,1\}^u$. By running Sign coherently, generate the state
$$|\psi_r\rangle := |0\rangle_{\mathbf{A}} \otimes |\mathsf{Sign}(k, 0\|r)\rangle_{\mathbf{B}} + |1\rangle_{\mathbf{A}} \otimes |\mathsf{Sign}(k, 1\|r)\rangle_{\mathbf{B}} \tag{21}$$
over registers $(\mathbf{A}, \mathbf{B})$. Output
$$\mathsf{pk} := (r, |\psi_r\rangle). \tag{22}$$
- $\mathsf{Enc}(\mathsf{vk}, \mathsf{pk}, b) \to \mathsf{ct}$: Parse $\mathsf{pk} = (r, \rho)$, where ρ is a quantum state over registers $(\mathbf{A}, \mathbf{B})$. The Enc algorithm consists of the following three steps.
  1. It coherently checks the signature in ρ. In other words, it applies the unitary
  $$U_{r,\mathsf{vk}} |\alpha\rangle_{\mathbf{A}} |\beta\rangle_{\mathbf{B}} |0...0\rangle_{\mathbf{D}} = |\alpha\rangle_{\mathbf{A}} |\beta\rangle_{\mathbf{B}} |\mathsf{Ver}(\mathsf{vk}, \alpha\|r, \beta)\rangle_{\mathbf{D}} \tag{23}$$
  on $\rho_{\mathbf{A},\mathbf{B}} \otimes |0...0\rangle \langle 0...0|_{\mathbf{D}}$,[13] and measures the register $\mathbf{D}$ in the computational basis. If the result is $\bot$, it outputs $\mathsf{ct} := \bot$ and halts. If the result is $\top$, it goes to the next step.
  2. It applies Z^b on the register $\mathbf{A}$.
  3. It measures all qubits in the registers $(\mathbf{A}, \mathbf{B})$ in the Hadamard basis to get the result d. It outputs
  $$\mathsf{ct} := (r, d). \tag{24}$$
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to b'$: Parse $\mathsf{sk} = k$ and $\mathsf{ct} = (r, d)$. Output
$$b' := d \cdot (0\|\mathsf{Sign}(k, 0\|r) \oplus 1\|\mathsf{Sign}(k, 1\|r)). \tag{25}$$

Theorem 6.1. *If* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *is a strong EUF-CMA secure digital signature scheme, then the QPKE scheme* $(\mathsf{SKGen}, \mathsf{PKGen}, \mathsf{Enc}, \mathsf{Dec})$ *above is correct and IND-pkT-CPA secure.*

The correctness is straightforward. First, the state over the registers $(\mathbf{A}, \mathbf{B})$ is $|\psi_r\rangle$ if pk was not tampered with and the first step of Enc algorithm got $\top$. Second, in that case, the state becomes
$$|0\rangle |\mathsf{Sign}(k, 0\|r)\rangle + (-1)^b |1\rangle |\mathsf{Sign}(k, 1\|r)\rangle \tag{26}$$
after the second step of Enc algorithm. Finally, because in that case d obtained in the third step of Enc algorithm satisfies
$$b = d \cdot (0\|\mathsf{Sign}(k, 0\|r) \oplus 1\|\mathsf{Sign}(k, 1\|r)), \tag{27}$$
we have $b' = b$.

We prove IND-pkT-CPA security in the next section.

[13] $\mathbf{C}$ is skipped, because $\mathbf{C}$ will be used later.

7 Proof of IND-pkT-CPA Security

In this section, we show IND-pkT-CPA security of our construction to complete the proof of Theorem 6.1. The outline of the proof is as follows. The security game for the IND-pkT-CPA security of our QPKE (Hybrid 0) is given in Figure 1. We introduce two more hybrids, Hybrid 1 (Figure 2) and Hybrid 2 (Figure 3). Hybrid 1 is the same as Hybrid 0 except that the challenger does not do the Hadamard-basis measurement in the third step of Enc algorithm, and the challenger sends the adversary r and the state over the registers $(\mathbf{A},\mathbf{B})$. Hybrid 2 is the same as Hybrid 1 except that the adversary outputs two bit strings μ_0,μ_1 and the adversary wins if $\mu_0=\mathsf{Sign}(k,0\|r)$ and $\mu_1=\mathsf{Sign}(k,1\|r)$. The formal proof is as follows.

Assume that the IND-pkT-CPA security of our construction is broken by a QPT adversary $\mathcal{A}$. It means the QPT adversary $\mathcal{A}$ wins Hybrid 0 with a non-negligible advantage. Then, it is clear that there is another QPT adversary $\mathcal{A}'$ that wins Hybrid 1 with a non-negligible advantage. ($\mathcal{A}'$ has only to do the Hadamard-basis measurement by itself.) From the $\mathcal{A}'$, we can construct a QPT adversary $\mathcal{A}''$ that wins Hybrid 2 with a non-negligible probability by using the idea of [HMY22]. (For details, see Section 7.1). Finally, we show in Section 7.2 that no QPT adversary can win Hybrid 2 except for a negligible probability. We thus have the contradiction, and therefore our QPKE is IND-pkT-CPA secure.

Hybrid 0

1. $\mathcal{C}$ runs $(k,\mathsf{vk})\leftarrow\mathsf{Gen}(1^\lambda)$. $\mathcal{C}$ sends vk to $\mathcal{A}$.
2. $\mathcal{C}$ chooses $r_1,...,r_m\leftarrow\{0,1\}^u$.
3. $\mathcal{C}$ sends $\{(r_i,|\psi_{r_i}\rangle)\}_{i=1}^m$ to the adversary $\mathcal{A}$, where
$$|\psi_{r_i}\rangle:=|0\rangle\otimes|\mathsf{Sign}(k,0\|r_i)\rangle+|1\rangle\otimes|\mathsf{Sign}(k,1\|r_i)\rangle\,. \tag{28}$$
4. $\mathcal{A}$ generates a quantum state over registers $(\mathbf{A},\mathbf{B},\mathbf{C})$. ($(\mathbf{A},\mathbf{B})$ corresponds to the quantum part of pk', and $\mathbf{C}$ corresponds to st.) $\mathcal{A}$ sends a bit string r and the registers $(\mathbf{A},\mathbf{B})$ to $\mathcal{C}$. $\mathcal{A}$ keeps the register $\mathbf{C}$.
5. $\mathcal{C}$ coherently checks the signature in the state sent from $\mathcal{A}$. If the result is $\bot$, it sends $\bot$ to $\mathcal{A}$ and halts. If the result is $\top$, it goes to the next step.
6. $\mathcal{C}$ chooses $b\leftarrow\{0,1\}$. $\mathcal{C}$ applies Z^b on the register $\mathbf{A}$.
7. $\mathcal{C}$ measures all qubits in $(\mathbf{A},\mathbf{B})$ in the Hadamard basis to get the result d. $\mathcal{C}$ sends (r,d) to $\mathcal{A}$.
8. $\mathcal{A}$ outputs b'. If $b'=b$, $\mathcal{A}$ wins.

Figure 1: Hybrid 0

Hybrid 1

1.-6. All the same as Figure 1.

7. $\mathcal{C}$ does not do the Hadamard-basis measurement, and $\mathcal{C}$ sends r and registers $(\mathbf{A},\mathbf{B})$ to $\mathcal{A}$.
8. The same as Figure 1.

Figure 2: Hybrid 1

Hybrid 2

1.-7. All the same as Figure 2.

8. $\mathcal{A}$ outputs (μ_0, μ_1). If $\mu_0 = \mathsf{Sign}(k, 0\|r)$ and $\mu_1 = \mathsf{Sign}(k, 1\|r)$, $\mathcal{A}$ wins.

Figure 3: Hybrid 2

7.1 From Distinguishing to Outputting Two Signatures

We present the construction of $\mathcal{A}''$. Assume that there exists a QPT adversary $\mathcal{A}'$ and a polynomial p such that

$$|\Pr[1 \leftarrow \mathcal{A}' \mid b = 0] - \Pr[1 \leftarrow \mathcal{A}' \mid b = 1]| \geq \frac{1}{p(\lambda)} \tag{29}$$

in Hybrid 1 (Figure 2) for all $\lambda \in I$ with an infinite set I. From the $\mathcal{A}'$, we construct a QPT adversary $\mathcal{A}''$ such that

$$\Pr[(\mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r)) \leftarrow \mathcal{A}''] \geq \frac{1}{q(\lambda)} \tag{30}$$

in Hybrid 2 (Figure 3) with a polynomial q for infinitely many λ.

Let $t := (k, \mathsf{vk}, r_1, ..., r_m, r)$, and $\Pr[t]$ be the probability that t is generated in Item 1, Item 2, and Item 4 in the game of Figure 2. Let Good be the event that $\mathcal{C}$ gets $\top$ in Item 5 in the game of Figure 2. Let Bad be the event that Good does not occur. Then, from Equation (29), we have

$$\begin{aligned}\frac{1}{p(\lambda)} \leq \Big| &\sum_t \Pr[t] \Pr[\mathsf{Good} \mid t] \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Good}, b = 0] + \sum_t \Pr[t] \Pr[\mathsf{Bad} \mid t] \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Bad}, b = 0] \\ &- \sum_t \Pr[t] \Pr[\mathsf{Good} \mid t] \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Good}, b = 1] - \sum_t \Pr[t] \Pr[\mathsf{Bad} \mid t] \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Bad}, b = 1]\Big| \end{aligned} \tag{31}$$

$$\begin{aligned} &\leq \sum_t \Pr[t] \Pr[\mathsf{Good} \mid t] \Big| \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Good}, b = 0] - \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Good}, b = 1] \Big| \\ &+ \sum_t \Pr[t] \Pr[\mathsf{Bad} \mid t] \Big| \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Bad}, b = 0] - \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Bad}, b = 1] \Big| \end{aligned} \tag{32}$$

$$= \sum_t \Pr[t] \Pr[\mathsf{Good} \mid t] \Big| \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Good}, b = 0] - \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Good}, b = 1] \Big| \tag{33}$$

for all $\lambda \in I$, because if Bad occurs, $\mathcal{A}'$ gets only $\perp$ which contains no information about b. (Here, we often abuse notation to just write t to mean the event that t is generated in Item 1, Item 2, and Item 4.) Therefore, if we define

$$T_\lambda := \Big\{ t : \Pr[\mathsf{Good} \mid t] \cdot \Big| \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Good}, b = 0] - \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Good}, b = 1] \Big| \geq \frac{1}{2p(\lambda)} \Big\}, \tag{34}$$

we have, for all $\lambda \in I$,

$$\Pr[\mathsf{Good} \mid t] \geq \frac{1}{4p(\lambda)} \tag{35}$$

and

$$\Big| \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Good}, b = 0] - \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Good}, b = 1] \Big| \geq \frac{1}{2p(\lambda)} \tag{36}$$

for any $t \in T_\lambda$ and

$$\sum_{t \in T_\lambda} \Pr[t] \geq \frac{1}{2p(\lambda)}. \tag{37}$$

Let $|\phi_b^{t,good}\rangle$ be the state over the registers $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ immediately before Item 8 of Figure 2 given that t is generated, Good occurred, and b is chosen in Item 6 of Figure 2. We can show the following lemma. (Its proof is given later.)

Lemma 7.1. *If* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *is strong EUF-CMA secure, there exists a subset* $T'_\lambda \subseteq T_\lambda$ *such that the following is satisfied for all* $\lambda \in I'$*, where* $I' := \{\lambda \in I : \lambda \geq \lambda_0\}$ *with a certain* λ_0*.*

- $\sum_{t \in T'_\lambda} \Pr[t] \geq \frac{1}{4p(\lambda)}$.
- *For any* $t \in T'_\lambda$, $|\phi_b^{t,good}\rangle$ *is close to a state*

  $$|\tilde{\phi}_b^{t,good}\rangle := c_0 |0\rangle_{\mathbf{A}} |\mathsf{Sign}(k, 0\|r)\rangle_{\mathbf{B}} |\Psi_0\rangle_{\mathbf{C}} + (-1)^b c_1 |1\rangle_{\mathbf{A}} |\mathsf{Sign}(k, 1\|r)\rangle_{\mathbf{B}} |\Psi_1\rangle_{\mathbf{C}} \tag{38}$$

  within the trace distance $\frac{1}{p^{10}(\lambda)}$*, where* c_0 *and* c_1 *are some complex coefficients such that* $|c_0|^2 + |c_1|^2 = 1$*, and* $|\Psi_0\rangle$ *and* $|\Psi_1\rangle$ *are some normalized states.*

Now let us fix $t \in T'_\lambda$. Also, assume that Good occurred. Because $T'_\lambda \subseteq T_\lambda$, it means that $t \in T_\lambda$. Then, from Equation (36),

$$\Big| \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Good}, b = 0] - \Pr[1 \leftarrow \mathcal{A}' \mid t, \mathsf{Good}, b = 1] \Big| = \Delta \tag{39}$$

for a non-negligible $\Delta \geq \frac{1}{2p(\lambda)}$ for all $\lambda \in I$. Without loss of generality, we can assume that in Item 8 of Figure 2, $\mathcal{A}'$ applies a unitary V on the state $|\phi_b^{t,good}\rangle$, and measures the register $\mathbf{A}$ in the computational basis to get $b' \in \{0, 1\}$. By Equation (39) we have

$$V |\phi_0^{t,good}\rangle = \sqrt{p} |1\rangle_{\mathbf{A}} |\nu_1\rangle_{\mathbf{B},\mathbf{C}} + \sqrt{1-p} |0\rangle_{\mathbf{A}} |\nu_0\rangle_{\mathbf{B},\mathbf{C}} \tag{40}$$

$$V |\phi_1^{t,good}\rangle = \sqrt{1-p+\Delta} |0\rangle_{\mathbf{A}} |\xi_0\rangle_{\mathbf{B},\mathbf{C}} + \sqrt{p-\Delta} |1\rangle_{\mathbf{A}} |\xi_1\rangle_{\mathbf{B},\mathbf{C}} \tag{41}$$

for some real number p and some normalized states $|\nu_0\rangle, |\nu_1\rangle, |\xi_0\rangle, |\xi_1\rangle$. (This is because any state can be written as $p |1\rangle |\nu_1\rangle + \sqrt{1-p} |0\rangle |\nu_0\rangle$ with some p and normalized states $|\nu_0\rangle, |\nu_1\rangle$, and due to Equation (39), the coefficients of $|1\rangle |\xi_1\rangle$ has to be $\sqrt{p-\Delta}$.) If we define W as

$$W := V^\dagger (Z \otimes I) V, \tag{42}$$

we have

$$|\langle \tilde{\phi}_b^{t,good} | W | \tilde{\phi}_b^{t,good} \rangle - \langle \phi_b^{t,good} | W | \phi_b^{t,good} \rangle| \leq \frac{2}{p^{10}(\lambda)} \tag{43}$$

for all $\lambda \in I'$ from Lemma 7.1. Therefore,

$$|c_0^* c_1 \langle 0| \langle \mathsf{Sign}(k, 0\|r)| \langle \Psi_0| W |1\rangle |\mathsf{Sign}(k, 1\|r)\rangle |\Psi_1\rangle \tag{44}$$

$$+ c_0 c_1^* \langle 1| \langle \mathsf{Sign}(k, 1\|r)| \langle \Psi_1| W |0\rangle |\mathsf{Sign}(k, 0\|r)\rangle |\Psi_0\rangle | \tag{45}$$

$$= \frac{1}{4}|((\langle \tilde{\phi}_0^{t,good}| + \langle \tilde{\phi}_1^{t,good}|)W(|\tilde{\phi}_0^{t,good}\rangle - |\tilde{\phi}_1^{t,good}\rangle) \tag{46}$$

$$+ ((\langle \tilde{\phi}_0^{t,good}| - \langle \tilde{\phi}_1^{t,good}|)W(|\tilde{\phi}_0^{t,good}\rangle + |\tilde{\phi}_1^{t,good}\rangle))| \tag{47}$$

$$= \frac{1}{2}| \langle \tilde{\phi}_0^{t,good}| W |\tilde{\phi}_0^{t,good}\rangle - \langle \tilde{\phi}_1^{t,good}| W |\tilde{\phi}_1^{t,good}\rangle | \tag{48}$$

$$\geq \frac{1}{2}| \langle \phi_0^{t,good}| W |\phi_0^{t,good}\rangle - \langle \phi_1^{t,good}| W |\phi_1^{t,good}\rangle | - \frac{2}{p^{10}(\lambda)} \tag{49}$$

$$\begin{aligned} &= \frac{1}{2}\Big|\Big(\sqrt{p}\,\langle 1| \langle \nu_1| + \sqrt{1-p}\,\langle 0| \langle \nu_0|\Big)\Big(-\sqrt{p}\,|1\rangle |\nu_1\rangle + \sqrt{1-p}\,|0\rangle |\nu_0\rangle\Big) \\ &\quad - \Big(\sqrt{1-p+\Delta}\,\langle 0| \langle \xi_0| + \sqrt{p-\Delta}\,\langle 1| \langle \xi_1|\Big)\Big(\sqrt{1-p+\Delta}\,|0\rangle |\xi_0\rangle - \sqrt{p-\Delta}\,|1\rangle |\xi_1\rangle\Big)\Big| - \frac{2}{p^{10}(\lambda)} \end{aligned} \tag{50}$$

$$= \frac{1}{2}\,|-p + (1-p) - (1-p+\Delta) + (p-\Delta)| - \frac{2}{p^{10}(\lambda)} \tag{51}$$

$$= \Delta - \frac{2}{p^{10}(\lambda)} \tag{52}$$

$$\geq \frac{1}{2p(\lambda)} - \frac{2}{p^{10}(\lambda)} \tag{53}$$

$$\geq \frac{1}{p(\lambda)} \tag{54}$$

for all $\lambda \in I'$. Here, Equation (49) follows from Equation (43), and Equation (50) follows from Equations (40) and (41) and the definition of W. From the triangle inequality and the facts that $|c_0| \leq 1$ and $|c_1| \leq 1$,

$$\frac{1}{p(\lambda)} \leq |c_1| \cdot | \langle 0| \langle \mathsf{Sign}(k, 0\|r)| \langle \Psi_0| W |1\rangle |\mathsf{Sign}(k, 1\|r)\rangle |\Psi_1\rangle | \tag{55}$$

$$+ |c_0| \cdot | \langle 1| \langle \mathsf{Sign}(k, 1\|r)| \langle \Psi_1| W |0\rangle |\mathsf{Sign}(k, 0\|r)\rangle |\Psi_0\rangle | \tag{56}$$

for all $\lambda \in I'$. Then,

$$\frac{1}{2p(\lambda)} \leq |c_1| \cdot | \langle 0| \langle \mathsf{Sign}(k, 0\|r)| \langle \Psi_0| W |1\rangle |\mathsf{Sign}(k, 1\|r)\rangle |\Psi_1\rangle | \tag{57}$$

or

$$\frac{1}{2p(\lambda)} \leq |c_0| \cdot | \langle 1| \langle \mathsf{Sign}(k, 1\|r)| \langle \Psi_1| W |0\rangle |\mathsf{Sign}(k, 0\|r)\rangle |\Psi_0\rangle | \tag{58}$$

holds for all $\lambda \in I'$. Assume that the latter holds. (The following proof can be easily modified even if the former holds.) Then

$$\frac{1}{4p^2(\lambda)} \leq |c_0|^2 \cdot | \langle 1| \langle \mathsf{Sign}(k, 1\|r)| \langle \Psi_1| W |0\rangle |\mathsf{Sign}(k, 0\|r)\rangle |\Psi_0\rangle |^2 \tag{59}$$

$$\leq |c_0|^2 \cdot \|(I \otimes \langle \mathsf{Sign}(k, 1\|r)| \otimes I) W |0\rangle |\mathsf{Sign}(k, 0\|r)\rangle |\Psi_0\rangle \|^2 \tag{60}$$

$\mathcal{A}''$

1. Simulate $\mathcal{A}'$ in steps 1.-7. of Figure 3. If $\bot$ is sent from $\mathcal{C}$, output $\bot$ and halt.
2. Measure the register $\mathbf{A}$ in the computational basis. If the result is 1, output $\bot$ and halt. If the result is 0, measure the register $\mathbf{B}$ of the post-measurement state in the computational basis to get the measurement result μ_0.
3. Apply W on the post-measurement state and measure the register $\mathbf{B}$ in the computational basis to get the result μ_1.
4. Output (μ_0, μ_1).

Figure 4: $\mathcal{A}''$

for all $\lambda \in I'$. With this W, we construct the QPT adversary $\mathcal{A}''$ as is shown in Figure 4.

We show that $\mathcal{A}''$ wins the game of Figure 3 with a non-negligible probability for infinitely many λ. The probability that $t \in T'_\lambda$ and Good occur in Item 1 of Figure 4 is at least $\frac{1}{16p^2(\lambda)}$ for all $\lambda \in I'$, because of the following reasons. First, $\sum_{t \in T'_\lambda} \Pr[t] \geq \frac{1}{4p(\lambda)}$ for all $\lambda \in I'$ from Lemma 7.1. Second, because $t \in T'_\lambda$ means $t \in T_\lambda$, $\Pr[\mathsf{Good} \mid t] \geq \frac{1}{4p(\lambda)}$ for all $\lambda \in I$ from Equation (35).

Assume that $t \in T'_\lambda$ and Good occur. If $\mathcal{A}''$ does the operations in Item 2 and Item 3 on $|\tilde{\phi}_b^{t,good}\rangle$, the probability that $(\mu_0, \mu_1) = (\mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r))$ is at least $\frac{1}{4p^2(\lambda)}$ for all $\lambda \in I'$ from Equation (60). From Lemma 7.1, the trace distance between $|\phi_b^{t,good}\rangle$ and $|\tilde{\phi}_b^{t,good}\rangle$ is at most $\frac{1}{p^{10}(\lambda)}$ for all $\lambda \in I'$. Therefore, if $\mathcal{A}''$ does the operations in Item 2 and Item 3 on $|\phi_b^{t,good}\rangle$, the probability that $(\mu_0, \mu_1) = (\mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r))$ is at least $\frac{1}{4p^2(\lambda)} - \frac{1}{p^{10}(\lambda)}$ for all $\lambda \in I'$. Hence, the overall probability that $\mathcal{A}''$ outputs $(\mu_0, \mu_1) = (\mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r))$ is non-negligible for infinitely many λ.

We prove Lemma 7.1 to complete this subsection.

Proof of Lemma 7.1. Fix $t \in T_\lambda$. Immediately before the coherent signature test in Item 5 of Figure 2, the entire state over the registers $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is generally written as

$$\sum_{\alpha,\beta} d_{\alpha,\beta} \, |\alpha\rangle_{\mathbf{A}} \, |\beta\rangle_{\mathbf{B}} \, |\Lambda_{\alpha,\beta}\rangle_{\mathbf{C}} \,, \tag{61}$$

where $d_{\alpha,\beta}$ are some complex coefficients such that $\sum_{\alpha,\beta} |d_{\alpha,\beta}|^2 = 1$, and $|\Lambda_{\alpha,\beta}\rangle$ are some normalized states. Define the set

$$S \coloneqq \{(\alpha, \beta) : \mathsf{Ver}(\mathsf{vk}, \alpha\|r, \beta) = \top \wedge \beta \neq \mathsf{Sign}(k, \alpha\|r)\}. \tag{62}$$

The (unnormalized) state after obtaining $\top$ in the coherent signature test in Item 5 of Figure 2 is

$$\begin{aligned} &d_{0,\mathsf{Sign}(k,0\|r)} \, |0\rangle_{\mathbf{A}} \, |\mathsf{Sign}(k, 0\|r)\rangle_{\mathbf{B}} \, |\Lambda_{0,\mathsf{Sign}(k,0\|r)}\rangle_{\mathbf{C}} \\ &+ d_{1,\mathsf{Sign}(k,1\|r)} \, |1\rangle_{\mathbf{A}} \, |\mathsf{Sign}(k, 1\|r)\rangle_{\mathbf{B}} \, |\Lambda_{1,\mathsf{Sign}(k,1\|r)}\rangle_{\mathbf{C}} \\ &+ \sum_{(\alpha,\beta) \in S} d_{\alpha,\beta} \, |\alpha\rangle_{\mathbf{A}} \, |\beta\rangle_{\mathbf{B}} \, |\Lambda_{\alpha,\beta}\rangle_{\mathbf{C}} \,. \end{aligned} \tag{63}$$

Define

$$T'_\lambda \coloneqq \Big\{ t \in T_\lambda : \sum_{(\alpha,\beta) \in S} |d_{\alpha,\beta}|^2 \leq \frac{1}{4p^{21}(\lambda)} \Big\}. \tag{64}$$

If

$$\sum_{t\in T_\lambda\setminus T'_\lambda} \Pr[t] \geq \frac{1}{4p(\lambda)} \tag{65}$$

for infinitely many $\lambda \in I$, it contradicts the strong EUF-CMA security of the digital signature scheme. Therefore,

$$\sum_{t\in T_\lambda\setminus T'_\lambda} \Pr[t] \leq \frac{1}{4p(\lambda)} \tag{66}$$

for all $\lambda \in I'$, where $I' := \{\lambda \in I : \lambda \geq \lambda_0\}$ with a certain λ_0. This means that

$$\begin{align}\sum_{t\in T'_\lambda} \Pr[t] &\geq \sum_{t\in T_\lambda} \Pr[t] - \frac{1}{4p(\lambda)} \tag{67}\\ &\geq \frac{1}{2p(\lambda)} - \frac{1}{4p(\lambda)} \tag{68}\\ &= \frac{1}{4p(\lambda)} \tag{69}\end{align}$$

for all $\lambda \in I'$.

Moreover, because $t \in T'_\lambda$ means $t \in T_\lambda$, $\Pr[\mathsf{Good} \mid t] \geq \frac{1}{4p(\lambda)}$ for all $\lambda \in I$ from Equation (35). Therefore, for any $t \in T'_\lambda$,

$$|d_{0,\mathsf{Sign}(k,0\|r)}|^2 + |d_{1,\mathsf{Sign}(k,1\|r)}|^2 + \sum_{(\alpha,\beta)\in S} |d_{\alpha,\beta}|^2 \geq \frac{1}{4p(\lambda)} \tag{70}$$

for all $\lambda \in I$. If we renormalize the state of Equation (63) and apply Z^b, we have

$$\begin{align}|\phi_b^{t,good}\rangle = &\frac{d_{0,\mathsf{Sign}(k,0\|r)}}{\sqrt{|d_{0,\mathsf{Sign}(k,0\|r)}|^2 + |d_{1,\mathsf{Sign}(k,1\|r)}|^2 + \sum_{(\alpha,\beta)\in S} |d_{\alpha,\beta}|^2}} |0\rangle_{\mathbf{A}} |\mathsf{Sign}(k,0\|r)\rangle_{\mathbf{B}} |\Lambda_{0,\mathsf{Sign}(k,0\|r)}\rangle_{\mathbf{C}} \tag{71}\\ &+ (-1)^b \frac{d_{1,\mathsf{Sign}(k,1\|r)}}{\sqrt{|d_{0,\mathsf{Sign}(k,0\|r)}|^2 + |d_{1,\mathsf{Sign}(k,1\|r)}|^2 + \sum_{(\alpha,\beta)\in S} |d_{\alpha,\beta}|^2}} |1\rangle_{\mathbf{A}} |\mathsf{Sign}(k,1\|r)\rangle_{\mathbf{B}} |\Lambda_{1,\mathsf{Sign}(k,1\|r)}\rangle_{\mathbf{C}} \tag{72}\\ &+ Z^b \frac{\sum_{(\alpha,\beta)\in S} d_{\alpha,\beta}}{\sqrt{|d_{0,\mathsf{Sign}(k,0\|r)}|^2 + |d_{1,\mathsf{Sign}(k,1\|r)}|^2 + \sum_{(\alpha,\beta)\in S} |d_{\alpha,\beta}|^2}} |\alpha\rangle_{\mathbf{A}} |\beta\rangle_{\mathbf{B}} |\Lambda_{\alpha,\beta}\rangle_{\mathbf{C}}. \tag{73}\end{align}$$

For any $t \in T'_\lambda$, its trace distance to the state

$$\begin{align}&\frac{d_{0,\mathsf{Sign}(k,0\|r)}}{\sqrt{|d_{0,\mathsf{Sign}(k,0\|r)}|^2 + |d_{1,\mathsf{Sign}(k,1\|r)}|^2}} |0\rangle_{\mathbf{A}} |\mathsf{Sign}(k,0\|r)\rangle_{\mathbf{B}} |\Lambda_{0,\mathsf{Sign}(k,0\|r)}\rangle_{\mathbf{C}} \tag{74}\\ &+ (-1)^b \frac{d_{1,\mathsf{Sign}(k,1\|r)}}{\sqrt{|d_{0,\mathsf{Sign}(k,0\|r)}|^2 + |d_{1,\mathsf{Sign}(k,1\|r)}|^2}} |1\rangle_{\mathbf{A}} |\mathsf{Sign}(k,1\|r)\rangle_{\mathbf{B}} |\Lambda_{1,\mathsf{Sign}(k,1\|r)}\rangle_{\mathbf{C}} \tag{75}\end{align}$$

is less than $\frac{1}{p^{10}(\lambda)}$ for all $\lambda \in I$. □

7.2 No QPT Adversary Can Output Two Signatures

Here we show that no QPT adversary can win Hybrid 2 (Figure 3) with a non-negligible probability. We first give an intuitive argument for the proof, and them give a precise proof.

Intuitive argument for the proof is as follows. First, note that the probability that all $\{r_i\}_{i=1}^m$ are distinct in Item 2 in Figure 3 is at least $1-\mathsf{negl}(\lambda)$. Therefore, we can assume that all $\{r_i\}_{i=1}^m$ are distinct with a negligible loss in the adversary's winning probability. If $r\notin\{r_i\}_{i=1}^m$, it is clear that $\mathcal{A}$ cannot win the game of Figure 3 except for a negligible probability. The reason is that $\mathcal{A}$ cannot find $\mathsf{Sign}(k,0\|r)$ or $\mathsf{Sign}(k,1\|r)$ except for a negligible probability due to the security of the digital signature scheme. Therefore, we assume that r is equal to one of the $\{r_i\}_{i=1}^m$.

Assume that, in the game of Figure 3, $\mathcal{C}$ is replaced with $\mathcal{C}'$ who is the same as $\mathcal{C}$ except that it measures the first qubit of $|\psi_r\rangle$ in the computational basis before sending the states in Item 3. Let $s\in\{0,1\}$ be the measurement result. Then, for any QPT adversary $\mathcal{A}$, the probability that $\mathcal{A}$ wins the game of Figure 3 is negligible. The reason is that $\mathcal{A}$ cannot find $\mathsf{Sign}(k,s\oplus 1\|r)$ except for a negligible probability due to the strong EUF-CMA security of the digital signature scheme. From Lemma 4.11, we therefore have

$$\begin{aligned}\Pr[(\mathsf{Sign}(k,0\|r),\mathsf{Sign}(k,1\|r))\leftarrow\mathcal{A}\mid\mathcal{C}]&\le 2\Pr[(\mathsf{Sign}(k,0\|r),\mathsf{Sign}(k,1\|r))\leftarrow\mathcal{A}\mid\mathcal{C}'] &(76)\\ &\le\mathsf{negl}(\lambda), &(77)\end{aligned}$$

where the left-hand-side of Equation (76) is the probability that $\mathcal{A}$ outputs $(\mathsf{Sign}(k,0\|r),\mathsf{Sign}(k,1\|r))$ with the challenger $\mathcal{C}$, and the right-hand-side is that with the challenger $\mathcal{C}'$.

We give a precise proof below. Let Alg be an algorithm that, on input $(r_1,...,r_m)$, simulates $\mathcal{C}$ and $\mathcal{A}$ in

Figure 3 and outputs (r, μ_0, μ_1). The probability that $\mathcal{A}$ wins in the game of Figure 3 is

$$\frac{1}{2^{um}} \sum_{r_1,...,r_m} \sum_{r} \Pr[(r, \mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r)) \leftarrow \mathsf{Alg}(r_1, ..., r_m)] \tag{78}$$

$$= \frac{1}{2^{um}} \sum_{(r_1,...,r_m)\in R} \sum_{r} \Pr[(r, \mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r)) \leftarrow \mathsf{Alg}(r_1, ..., r_m)] \tag{79}$$

$$+ \frac{1}{2^{um}} \sum_{(r_1,...,r_m)\notin R} \sum_{r} \Pr[(r, \mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r)) \leftarrow \mathsf{Alg}(r_1, ..., r_m)] \tag{80}$$

$$\leq \frac{1}{2^{um}} \sum_{(r_1,...,r_m)\in R} \sum_{r} \Pr[(r, \mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r)) \leftarrow \mathsf{Alg}(r_1, ..., r_m)] + \frac{1}{2^{um}} \sum_{(r_1,...,r_m)\notin R} \tag{81}$$

$$\leq \frac{1}{2^{um}} \sum_{(r_1,...,r_m)\in R} \sum_{r} \Pr[(r, \mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r)) \leftarrow \mathsf{Alg}(r_1, ..., r_m)] + \frac{(m-1)m}{2^u} \tag{82}$$

$$= \frac{1}{2^{um}} \sum_{(r_1,...,r_m)\in R} \sum_{r\in\{r_i\}_{i=1}^m} \Pr[(r, \mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r)) \leftarrow \mathsf{Alg}(r_1, ..., r_m)] \tag{83}$$

$$+ \frac{1}{2^{um}} \sum_{(r_1,...,r_m)\in R} \sum_{r\notin\{r_i\}_{i=1}^m} \Pr[(r, \mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r)) \leftarrow \mathsf{Alg}(r_1, ..., r_m)] + \frac{(m-1)m}{2^u} \tag{84}$$

$$\leq \frac{1}{2^{um}} \sum_{(r_1,...,r_m)\in R} \sum_{r\in\{r_i\}_{i=1}^m} \Pr[(r, \mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r)) \leftarrow \mathsf{Alg}(r_1, ..., r_m)] \tag{85}$$

$$+ \frac{1}{2^{um}} \sum_{(r_1,...,r_m)\in R} \mathsf{negl}(\lambda) + \frac{(m-1)m}{2^u} \tag{86}$$

$$\leq \frac{1}{2^{um}} \sum_{(r_1,...,r_m)\in R} \sum_{r\in\{r_i\}_{i=1}^m} \Pr[(r, \mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r)) \leftarrow \mathsf{Alg}(r_1, ..., r_m)] \tag{87}$$

$$+ \mathsf{negl}(\lambda) + \frac{(m-1)m}{2^u} \tag{88}$$

$$\leq \frac{1}{2^{um}} \sum_{(r_1,...,r_m)\in R} \sum_{r\in\{r_i\}_{i=1}^m} 2\Pr'[(r, \mathsf{Sign}(k, 0\|r), \mathsf{Sign}(k, 1\|r)) \leftarrow \mathsf{Alg}(r_1, ..., r_m)] \tag{89}$$

$$+ \mathsf{negl}(\lambda) + \frac{(m-1)m}{2^u} \tag{90}$$

$$\leq \frac{1}{2^{um}} \sum_{(r_1,...,r_m)\in R} \sum_{r\in\{r_i\}_{i=1}^m} \mathsf{negl}(\lambda) + \mathsf{negl}(\lambda) + \frac{(m-1)m}{2^u} \tag{91}$$

$$\leq \mathsf{negl}(\lambda) + \mathsf{negl}(\lambda) + \frac{(m-1)m}{2^u} \tag{92}$$

$$= \mathsf{negl}(\lambda). \tag{93}$$

Here, $R := \{(r_1, ..., r_m) : \text{All of them are distinct}\}$. In Equation (86), we have used the strong EUF-CMA security of the digital signature scheme. $\Pr'$ is the probability that, in Alg, $\mathcal{C}$ is replaced with $\mathcal{C}'$ who is the same as $\mathcal{C}$ except that it measures the first qubit of $|\psi_r\rangle$ in the computational basis before sending the states in Item 3. Equation (89) comes from Lemma 4.11. Equation (91) is from the strong EUF-CMA security of the digital signature scheme.

8 Definition of Chosen Ciphertext Security

In this section, we define CCA security for QPKE and related security notions. We start with an intermediate notion between CPA security and CCA security that we call security against challenge validity attack (CVA).

Definition 8.1 (IND-pkT-CVA security). *For any polynomial m, and any QPT adversary $\mathcal{A}$, we have*

$$\Pr\left[b \leftarrow \mathcal{A}(\mathsf{ct}^*, \mathsf{cv}, \mathsf{st}) : \begin{array}{r} (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SKGen}(1^\lambda) \\ \mathsf{pk}_1, ..., \mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m} \\ (\mathsf{pk}', \mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{vk}, \mathsf{pk}_1, ..., \mathsf{pk}_m) \\ b \leftarrow \{0,1\} \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{vk}, \mathsf{pk}', \mathsf{msg}_b) \\ \mathsf{cv} := 0 \textit{ if } \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}^*) = \bot \textit{ and otherwise } \mathsf{cv} := 1 \end{array}\right] \le \frac{1}{2} + \mathsf{negl}(\lambda). \tag{94}$$

Here, $\mathsf{pk}_1, ..., \mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m}$ *means that* PKGen *is executed* m *times and* pk_i *is the output of the* i*th execution of* PKGen. st *is a quantum internal state of* $\mathcal{A}$*, which can be entangled with* pk'.

We then define CCA security for QPKE.

Definition 8.2 (IND-pkT-CCA security). *For any polynomial m, and any QPT adversary $\mathcal{A}$, we have*

$$\Pr\left[b \leftarrow \mathcal{A}^{O_{\mathsf{Dec},2}(\cdot)}(\mathsf{ct}^*, \mathsf{cv}, \mathsf{st}) : \begin{array}{r} (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SKGen}(1^\lambda) \\ \mathsf{pk}_1, ..., \mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m} \\ (\mathsf{pk}', \mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}^{O_{\mathsf{Dec},1}(\cdot)}(\mathsf{vk}, \mathsf{pk}_1, ..., \mathsf{pk}_m) \\ b \leftarrow \{0,1\} \\ \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{vk}, \mathsf{pk}', \mathsf{msg}_b) \\ \mathsf{cv} := 0 \textit{ if } \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}^*) = \bot \textit{ and otherwise } \mathsf{cv} := 1 \end{array}\right] \le \frac{1}{2} + \mathsf{negl}(\lambda). \tag{95}$$

Here, $\mathsf{pk}_1, ..., \mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m}$ *means that* PKGen *is executed* m *times and* pk_i *is the output of the* i*th execution of* PKGen. st *is a quantum internal state of* $\mathcal{A}$*, which can be entangled with* pk'. *Also,* $O_{\mathsf{Dec},1}(\mathsf{ct})$ *returns* $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ *for any* ct. $O_{\mathsf{Dec},2}$ *behaves identically to* $O_{\mathsf{Dec},1}$ *except that* $O_{\mathsf{Dec},2}$ *returns* $\bot$ *to the input* ct^*.

Definition 8.3 (IND-pkT-1CCA security). *IND-pkT-1CCA security is defined in the same way as IND-pkT-CCA security except that in the security game we require that the total number of* $\mathcal{A}$*'s query to* $O_{\mathsf{Dec},1}$ *and* $O_{\mathsf{Dec},2}$ *is at most one.*

Security under single public key. For $X \in \{$IND-pkT-CPA, IND-pkT-CVA, IND-pkT-CCA, IND-pkT-1CCA$\}$ security, we define $X^{(1)}$ security as its variant where the number of public keys given to the adversary is fixed to one. Note that $X^{(1)}$ security is implied by X security for any $X \in \{$IND-pkT-CPA, IND-pkT-CVA, IND-pkT-CCA, IND-pkT-1CCA$\}$.

We also define statistical variant of decryption error detectability that is useful to achieve CCA security with our transformations.

Definition 8.4 (Strong decryption error detectability). *We say that a QPKE scheme has strong decryption error detectability if for any* $\mathsf{sk}', \mathsf{vk}', \mathsf{pk}'$*, and* msg*, we have*

$$\Pr\left[\mathsf{msg}' \neq \bot \ \wedge \ \mathsf{msg}' \neq \mathsf{msg} : \begin{array}{r} \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{vk}', \mathsf{pk}', \mathsf{msg}) \\ \mathsf{msg}' \leftarrow \mathsf{Dec}(\mathsf{sk}', \mathsf{ct}) \end{array}\right] \le \mathsf{negl}(\lambda). \tag{96}$$

9 Transformations Achieving Chosen Ciphertext Security

In this section, we present the transformation from CPA secure QPKE scheme to CCA secure one. The transformation consists of the following four subroutines.

1. Transformation from IND-pkT-CPA secure one to IND-pkT-CVA secure one presented in Section 9.2.

2. Transformation from IND-pkT-CVA$^{(1)}$ secure one to IND-pkT-1CCA$^{(1)}$ secure one presented in Section 9.3.

3. Transformation from IND-pkT-1CCA$^{(1)}$ secure one to IND-pkT-CCA$^{(1)}$ secure one presented in Section 9.4.

4. Transformation from IND-pkT-CCA$^{(1)}$ secure one to IND-pkT-CCA secure one presented in Section 9.5.

Below, we first introduce the notion of tokenized MAC [BSS21] in Section 9.1 that is used in the third transformation, and then provide each transformations.

9.1 Preparations

Definition 9.1 (Tokenized MAC [BSS21]). *A tokenized MAC scheme with the message space* $\{0,1\}^\ell$ *is a set of algorithms* $(\mathsf{SKGen}, \mathsf{TKGen}, \mathsf{Sign}, \mathsf{Ver})$ *such that*

- $\mathsf{SKGen}(1^\lambda) \to \mathsf{sk}$: *It is a PPT algorithm that, on input the security parameter* λ*, outputs a classical secret key* sk.

- $\mathsf{TKGen}(\mathsf{sk}) \to \mathsf{token}$: *It is a QPT algorithm that, on input* sk*, outputs a quantum signing token* token.

- $\mathsf{Sign}(\mathsf{token}, \mathsf{msg}) \to \sigma$: *It is a QPT algorithm that, on input* token *and a message* $\mathsf{msg} \in \{0,1\}^\ell$*, outputs a classical signature* σ.

- $\mathsf{Ver}(\mathsf{sk}, \mathsf{msg}, \sigma) \to \top/\bot$: *It is a classical deterministic polynomial-time algorithm that, on input* sk, msg*, and* σ*, outputs* $\top$ *or* $\bot$.

We require the following correctness and unforgeability.

Correctness: *For any* msg,

$$\Pr[\top \leftarrow \mathsf{Ver}(\mathsf{sk}, \mathsf{msg}, \sigma) : \mathsf{sk} \leftarrow \mathsf{SKGen}(1^\lambda), \mathsf{token} \leftarrow \mathsf{TKGen}(\mathsf{sk}), \sigma \leftarrow \mathsf{Sign}(\mathsf{token}, \mathsf{msg})] \geq 1 - \mathsf{negl}(\lambda). \tag{97}$$

Unforgeability: *For any QPT adversary* $\mathcal{A}$ *with classical oracle access to the verification oracle* $\mathsf{Ver}(\mathsf{sk}, \cdot, \cdot)$,

$$\Pr\left[\begin{array}{rcr} \mathsf{msg}_1 \neq \mathsf{msg}_2 & & \mathsf{sk} \leftarrow \mathsf{SKGen}(1^\lambda), \\ \wedge \top \leftarrow \mathsf{Ver}(\mathsf{sk}, \mathsf{msg}_1, \sigma_1) & : & \mathsf{token} \leftarrow \mathsf{TKGen}(\mathsf{sk}), \\ \wedge \top \leftarrow \mathsf{Ver}(\mathsf{sk}, \mathsf{msg}_2, \sigma_2) & & (\mathsf{msg}_a, \sigma_a)_{a \in [2]} \leftarrow \mathcal{A}^{\mathsf{Ver}(\mathsf{sk}, \cdot, \cdot)}(\mathsf{token}) \end{array}\right] \leq \mathsf{negl}(\lambda). \tag{98}$$

Theorem 9.2 ([BSS21]). *Tokenized MAC exists if OWFs exist.*

Note that the unforgeability in the above definition is weaker than that in the original definition by [BSS21]. We use this weaker definition that is sufficient for our purpose for ease of exposition.

9.2 IND-pkT-CVA Secure QPKE via Cut-and-Choose

We show a generic construction of IND-pkT-CVA secure QPKE from IND-pkT-CPA secure QPKE using the cut-and-choose technique.

Let $\mathsf{QPKE} = (\mathsf{QPKE.SKGen}, \mathsf{QPKE.PKGen}, \mathsf{QPKE.Enc}, \mathsf{QPKE.Dec})$ be a QPKE scheme with message space $\{0,1\}^\ell$. Then we construct a QPKE scheme $\mathsf{CVA} = (\mathsf{CVA.SKGen}, \mathsf{CVA.PKGen}, \mathsf{CVA.Enc}, \mathsf{CVA.Dec})$ with message space $\{0,1\}^\ell$ as follows:

- $\mathsf{CVA.SKGen}(1^\lambda) \to (\mathsf{sk}, \mathsf{vk})$: Run $(\mathsf{sk}_i, \mathsf{vk}_i) \leftarrow \mathsf{QPKE.SKGen}(1^\lambda)$ for every $i \in [4\lambda]$. Output $\mathsf{sk} := (\mathsf{sk}_i)_{i\in[4\lambda]}$ and $\mathsf{vk} := (\mathsf{vk}_i)_{i\in[4\lambda]}$.
- $\mathsf{CVA.PKGen}(\mathsf{sk}) \to \mathsf{pk}$: Parse $\mathsf{sk} := (\mathsf{sk}_i)_{i\in[4\lambda]}$. Run $\mathsf{pk}_i \leftarrow \mathsf{QPKE.PKGen}(\mathsf{sk}_i)$ for $i \in [4\lambda]$ and outputs $\mathsf{pk} := (\mathsf{pk}_i)_{i\in[4\lambda]}$.
- $\mathsf{CVA.Enc}(\mathsf{vk}, \mathsf{pk}, \mathsf{msg}) \to \mathsf{ct}$: Parse $\mathsf{vk} := (\mathsf{vk}_i)_{i\in[4\lambda]}$ and $\mathsf{pk} := (\mathsf{pk}_i)_{i\in[4\lambda]}$. Generate a random 2λ size subset Test of $[4\lambda]$. Generate $u_i \leftarrow \{0,1\}^\ell$, run $\mathsf{ct}_i \leftarrow \mathsf{QPKE.Enc}(\mathsf{vk}_i, \mathsf{pk}_i, u_i)$ for every $i \in [4\lambda]$. Set $v_i := u_i$ if $i \in \mathsf{Test}$ and $v_i := u_i \oplus \mathsf{msg}$ otherwise. Output $\mathsf{ct} := (\mathsf{Test}, (\mathsf{ct}_i, v_i)_{i\in[4\lambda]})$.
- $\mathsf{CVA.Dec}(\mathsf{sk}, \mathsf{ct}) \to \mathsf{msg}$: Parse $\mathsf{sk} := (\mathsf{sk}_i)_{i\in[4\lambda]}$ and $\mathsf{ct} = (\mathsf{Test}, (\mathsf{ct}_i, v_i)_{i\in[4\lambda]})$. Output $\bot$ if $\mathsf{QPKE.Dec}(\mathsf{sk}_i, \mathsf{ct}_i) \neq v_i$ for some $i \in \mathsf{Test}$. Otherwise, run $u_i \leftarrow \mathsf{QPKE.Dec}(\mathsf{sk}_i, \mathsf{ct}_i)$ and compute $\mathsf{msg}_i := v_i \oplus u_i$ for every $i \in [4\lambda] \setminus \mathsf{Test}$, and output most frequently appeared msg. (If there are multiple such msg, output the lexicographically first one.)

Correctness. Correctness of CVA immediately follows from correctness of QPKE.

Strong decryption error detectability. Let $(\mathsf{sk}', \mathsf{vk}', \mathsf{pk}', \mathsf{msg})$ be any tuple of a secret key, verification key, public key, and message, where $\mathsf{sk}' := (\mathsf{sk}'_i)_{i\in[4\lambda]}$, $\mathsf{vk}' := (\mathsf{vk}'_i)_{i\in[4\lambda]}$, $\mathsf{pk}' := (\mathsf{pk}'_i)_{i\in[4\lambda]}$, and $\mathsf{msg} \in \{0,1\}^\ell$. Suppose we pick $u_i \leftarrow \{0,1\}^\ell$, generate $\mathsf{ct}_i \leftarrow \mathsf{QPKE.Enc}(\mathsf{vk}'_i, \mathsf{pk}'_i, u_i)$, and compute $u'_i \leftarrow \mathsf{QPKE.Dec}(\mathsf{sk}'_i, \mathsf{ct}_i)$ for every $i \in [4\lambda]$. We consider the following two cases.

- The first case is $u_i \neq u'_i$ for more than λ indices. In this case, a randomly chosen 2λ size subset Test includes at least one index i such that $u_i \neq u'_i$ and thus the decryption result of $\mathsf{ct} := (\mathsf{Test}, (\mathsf{ct}_i, v_i)_{i\in[4\lambda]})$ for randomly chosen Test is $\bot$ with overwhelming probability, where $v_i := u_i$ if $i \in \mathsf{Test}$ and otherwise $v_i := u_i \oplus \mathsf{msg}$.
- The second case is $u_i \neq u'_i$ for less than λ indices. In this case, for every choice of Test, msg occupies the majority among $\mathsf{msg}_i := \mathsf{msg} \oplus u_i \oplus u'_i$ for $i \in [4\lambda] \setminus \mathsf{Test}$. Thus, the decryption result of $\mathsf{ct} := (\mathsf{Test}, (\mathsf{ct}_i, v_i)_{i\in[4\lambda]})$ is either $\bot$ or msg, regardless of the choice of Test.

This proves the strong decryption error detectability of CVA.

IND-pkT-CVA security. We prove that if QPKE satisfies IND-pkT-CPA security, then CVA satisfies IND-pkT-CVA security. We consider the following games.

Hyb_0**:** This is the original security experiment for the IND-pkT-CVA security of CVA played between $\mathcal{A}$ and the challenger. The detailed description is as follows.

1. The challenger generates $(\mathsf{sk}_i, \mathsf{vk}_i) \leftarrow \mathsf{QPKE.SKGen}(1^\lambda)$ for every $i \in [4\lambda]$, and set $\mathsf{sk} := (\mathsf{sk}_i)_{i\in[4\lambda]}$ and $\mathsf{vk} := (\mathsf{vk}_i)_{i\in[4\lambda]}$. The challenger generates $\mathsf{pk}_i \leftarrow \mathsf{QPKE.PKGen}(\mathsf{sk}_i)$ for $i \in [4\lambda]$ and set $\mathsf{pk} := (\mathsf{pk}_i)_{i\in[4\lambda]}$.
2. The challenger runs $(\mathsf{pk}', \mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{vk}, \mathsf{pk})$.
3. The challenger parses $\mathsf{pk}' := (\mathsf{pk}'_i)_{i\in[4\lambda]}$ and picks $b \leftarrow \{0, 1\}$. The challenger generates ct^* as follows.
    - Generate a random 2λ size subset Test^* of $[4\lambda]$.
    - Generate $u_i^* \leftarrow \{0, 1\}^\ell$, run $\mathsf{ct}_i^* \leftarrow \mathsf{QPKE.Enc}(\mathsf{vk}_i, \mathsf{pk}'_i, u_i^*)$ for every $i \in [4\lambda]$.
    - Set $v_i^* := u_i^*$ if $i \in \mathsf{Test}^*$ and $v_i^* := u_i^* \oplus \mathsf{msg}_b$ otherwise.
    - Output $\mathsf{ct}^* := (\mathsf{Test}^*, (\mathsf{ct}_i^*, v_i^*)_{i\in[4\lambda]})$.

    The challenger also sets $\mathsf{cv} := 0$ if $\mathsf{CVA.Dec}(\mathsf{sk}, \mathsf{ct}^*) = \bot$ and otherwise sets $\mathsf{cv} := 1$.
4. The challenger runs $b' \leftarrow \mathcal{A}(\mathsf{cv}, \mathsf{ct}^*, \mathsf{st})$. The challenger outputs 1 if $b = b'$ and otherwise outputs 0.

Hyb_1**:** This is the same as Hyb_0 except that the challenger generates $\mathsf{ct}_i^* \leftarrow \mathsf{QPKE.Enc}(\mathsf{vk}_i, \mathsf{pk}'_i, 0^\ell)$ for every $i \in [4\lambda] \setminus \mathsf{Test}^*$.

We can prove $|\Pr[1 \leftarrow \mathsf{Hyb}_0] - \Pr[1 \leftarrow \mathsf{Hyb}_1]| = \mathsf{negl}(\lambda)$ using the IND-pkT-CPA security of QPKE with respect to instances such that the corresponding index i is not included in Test^*. Note that the reduction needs to know whether $\mathsf{CVA.Dec}(\mathsf{sk}, \mathsf{ct}^*) = \bot$ or not. This is possible since it can be computed with only sk_i for $i \in \mathsf{Test}^*$, which is generated by the reduction itself.

In Hyb_1, the challenge bit b is completely hidden from the view of $\mathcal{A}$ since b is masked by u_i for $i \in [4\lambda] \setminus \mathsf{Test}^*$. Thus, we have $\Pr[1 \leftarrow \mathsf{Hyb}_2] = \frac{1}{2}$. From the above discussions, CVA satisfies IND-pkT-CVA security.

9.3 IND-pkT-1CCA Secure QPKE from IND-pkT-CVA Secure One

We show how to construct IND-pkT-1CCA$^{(1)}$ secure QPKE from IND-pkT-CVA$^{(1)}$ secure one.

Let $\mathsf{CVA} = (\mathsf{CVA.SKGen}, \mathsf{CVA.PKGen}, \mathsf{CVA.Enc}, \mathsf{CVA.Dec})$ be a QPKE scheme with message space $\{0, 1\}^\ell$ and $\mathsf{SIG} = (\mathsf{SIG.Gen}, \mathsf{SIG.Sign}, \mathsf{SIG.Ver})$ be a digital signature scheme whose verification key is of length n. Then we construct a QPKE scheme $\mathsf{1CCA} = (\mathsf{1CCA.SKGen}, \mathsf{1CCA.PKGen}, \mathsf{1CCA.Enc}, \mathsf{1CCA.Dec})$ with message space $\{0, 1\}^\ell$ as follows, where for a verification key sigvk of SIG and an integer $i \in [n]$, $\mathsf{sigvk}[i]$ denotes the i-th bit of sigvk:

- $\mathsf{1CCA.SKGen}(1^\lambda) \rightarrow (\mathsf{sk}, \mathsf{vk})$: Run $(\mathsf{cva.sk}_{i,\alpha}, \mathsf{cva.vk}_{i,\alpha}) \leftarrow \mathsf{CVA.SKGen}(1^\lambda)$ for every $i \in [n]$ and $\alpha \in \{0, 1\}$. Output $\mathsf{sk} := (\mathsf{cva.sk}_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$ and $\mathsf{vk} := (\mathsf{cva.vk}_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$.
- $\mathsf{1CCA.PKGen}(\mathsf{sk}) \rightarrow \mathsf{pk}$: Parse $\mathsf{sk} := (\mathsf{cva.sk}_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$. Run $\mathsf{cva.pk}_{i,\alpha} \leftarrow \mathsf{CVA.PKGen}(\mathsf{cva.sk}_{i,\alpha})$ for every $i \in [n]$ and $\alpha \in \{0, 1\}$. Output $\mathsf{pk} := (\mathsf{cva.pk}_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$.
- $\mathsf{1CCA.Enc}(\mathsf{vk}, \mathsf{pk}, \mathsf{msg}) \rightarrow \mathsf{ct}$: Parse $\mathsf{vk} := (\mathsf{cva.vk}_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$ and $\mathsf{pk} := (\mathsf{cva.pk}_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$. Run $(\mathsf{sigvk}, \mathsf{sigk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$. Generate $u_i \leftarrow \{0, 1\}^\ell$ for every $i \in [n - 1]$ and set $u_n := \mathsf{msg} \oplus \bigoplus_{i\in[n-1]} u_i$. Run $\mathsf{cva.ct}_i \leftarrow \mathsf{CVA.Enc}(\mathsf{cva.vk}_{i,\mathsf{sigvk}[i]}, \mathsf{cva.pk}_{i,\mathsf{sigvk}[i]}, u_i)$ for every $i \in [n]$. Run $\sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{sigk}, \mathsf{cva.ct}_1 \| \cdots \| \mathsf{cva.ct}_n)$. Output $\mathsf{ct} := (\mathsf{sigvk}, (\mathsf{cva.ct}_i)_{i\in[n]}, \sigma)$.

- $\mathsf{1CCA.Dec}(\mathsf{sk}, \mathsf{ct}) \rightarrow \mathsf{msg}$: Parse $\mathsf{sk} := (\mathsf{cva.sk}_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$ and $\mathsf{ct} = (\mathsf{sigvk}, (\mathsf{cva.ct}_i)_{i\in[n]}, \sigma)$. Output $\bot$ if $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{cva.ct}_1\|\cdots\|\mathsf{cva.ct}_n, \sigma) = \bot$ and otherwise go to the next step. Run $u_i \leftarrow \mathsf{CVA.Dec}(\mathsf{cva.sk}_{i,\mathsf{sigvk}[i]}, \mathsf{cva.ct}_i)$ for every $i \in [n]$, and output $\bot$ if $u_i = \bot$ for some $i \in [n]$. Otherwise, output $\bigoplus_{i\in[n]} u_i$.

Correctness and strong decryption error detectability. The correctness and the strong decryption error detectability of $\mathsf{1CCA}$ immediately follow from those of CVA and the correctness of SIG.

IND-pkT-1CCA$^{(1)}$ security. We prove that if CVA satisfies IND-pkT-CVA$^{(1)}$ security and SIG satisfies strong unforgeability, then $\mathsf{1CCA}$ satisfies IND-pkT-1CCA$^{(1)}$ security.

Let $\mathcal{A}$ be any QPT adversary attacking the IND-pkT-1CCA$^{(1)}$ security of $\mathsf{1CCA}$. Without loss of generality, we assume that $\mathcal{A}$ makes exactly one decryption query. We proceed the proof using a sequence of experiments.

Hyb_0: This is the original security experiment for the IND-pkT-1CCA$^{(1)}$ security of $\mathsf{1CCA}$ played between $\mathcal{A}$ and the challenger. The detailed description is as follows.

1. The challenger generates $(\mathsf{cva.sk}_{i,\alpha}, \mathsf{cva.vk}_{i,\alpha}) \leftarrow \mathsf{CVA.SKGen}(1^\lambda)$ for every $i \in [n]$ and $\alpha \in \{0,1\}$, and sets $\mathsf{sk} := (\mathsf{cva.sk}_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$ and $\mathsf{vk} := (\mathsf{cva.vk}_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$. The challenger generates $\mathsf{cva.pk}_{i,\alpha} \leftarrow \mathsf{CVA.PKGen}(\mathsf{cva.sk}_{i,\alpha})$ for $i \in [n]$ and $\alpha \in \{0,1\}$, and sets $\mathsf{pk} := (\mathsf{cva.pk}_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$.
2. The challenger runs $(\mathsf{pk}', \mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{vk}, \mathsf{pk})^{O_{\mathsf{Dec},1}(\cdot)}$, where $O_{\mathsf{Dec},1}(\mathsf{ct})$ behaves as follows.
   - Parse $\mathsf{ct} = (\mathsf{sigvk}, (\mathsf{cva.ct}_i)_{i\in[n]}, \sigma)$.
   - Output $\bot$ if $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{cva.ct}_1\|\cdots\|\mathsf{cva.ct}_n, \sigma) = \bot$ and otherwise go to the next step.
   - Run $u_i \leftarrow \mathsf{CVA.Dec}(\mathsf{cva.sk}_{i,\mathsf{sigvk}[i]}, \mathsf{cva.ct}_i)$ for every $i \in [n]$, and output $\bot$ if $u_i = \bot$ for some $i \in [n]$.
   - Otherwise, output $\bigoplus_{i\in[n]} u_i$.
3. The challenger parses $\mathsf{pk}' := (\mathsf{cva.pk}'_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$ and picks $b \leftarrow \{0,1\}$. The challenger generates ct^* as follows.
   - Run $(\mathsf{sigvk}^*, \mathsf{sigk}^*) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$.
   - Generate $u_i^* \leftarrow \{0,1\}^\ell$ for every $i \in [n-1]$ and set $u_n^* := \mathsf{msg}_b \oplus \bigoplus_{i\in[n-1]} u_i^*$.
   - Run $\mathsf{cva.ct}_i^* \leftarrow \mathsf{CVA.Enc}(\mathsf{cva.vk}_{i,\mathsf{sigvk}^*[i]}, \mathsf{cva.pk}'_{i,\mathsf{sigvk}^*[i]}, u_i^*)$ for every $i \in [n]$.
   - Run $\sigma^* \leftarrow \mathsf{SIG.Sign}(\mathsf{sigk}^*, \mathsf{cva.ct}_1^*\|\cdots\|\mathsf{cva.ct}_n^*)$ and set $\mathsf{ct}^* := (\mathsf{sigvk}^*, (\mathsf{cva.ct}_i^*)_{i\in[n]}, \sigma^*)$.

   The challenger also sets $\mathsf{cv} := 0$ if $\mathsf{1CCA.Dec}(\mathsf{sk}, \mathsf{ct}^*) = \bot$ and otherwise sets $\mathsf{cv} := 1$.
4. The challenger runs $b' \leftarrow \mathcal{A}(\mathsf{cv}, \mathsf{ct}^*, \mathsf{st})^{O_{\mathsf{Dec},2}(\cdot)}$, where $O_{\mathsf{Dec},2}$ behaves exactly in the same way as $O_{\mathsf{Dec},1}$ except that $O_{\mathsf{Dec},2}$ given ct returns $\bot$ if $\mathsf{ct} = \mathsf{ct}^*$. The challenger outputs 1 if $b = b'$ and otherwise outputs 0.

Hyb_1: This is the same as Hyb_0 except that the challenger generates the key pair $(\mathsf{sigvk}^*, \mathsf{sigk}^*)$ of SIG that is used to generate the challenge ciphertext at the beginning of the game, and $O_{\mathsf{Dec},1}$ and $O_{\mathsf{Dec},2}$ behave as follows.

- Parse $\mathsf{ct} = (\mathsf{sigvk}, (\mathsf{cva.ct}_i)_{i\in[n]}, \sigma)$.

- Output $\bot$ if $\mathsf{sigvk} = \mathsf{sigvk}^*$ and otherwise go to the next step.
- Output $\bot$ if $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{cva.ct}_1\| \cdots \|\mathsf{cva.ct}_n, \sigma) = \bot$ and otherwise go to the next step.
- Run $u_i \leftarrow \mathsf{CVA.Dec}(\mathsf{cva.sk}_{i,\mathsf{sigvk}[i]}, \mathsf{cva.ct}_i)$ for every $i \in [n]$, and output $\bot$ if $u_i = \bot$ for some $i \in [n]$.
- Otherwise, output $\bigoplus_{i\in[n]} u_i$.

We define the following two events.

$\mathtt{Forge}_{j,1}$**:** In Hyb_j, $\mathcal{A}$ queries $\mathsf{ct} = (\mathsf{sigvk}, (\mathsf{cva.ct}_i)_{i\in[n]}, \sigma)$ to $O_{\mathsf{Dec},1}$ such that $\mathsf{sigvk} = \mathsf{sigvk}^*$ and $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{cva.ct}_1\| \cdots \|\mathsf{cva.ct}_n, \sigma) = \top$.

$\mathtt{Forge}_{j,2}$**:** In Hyb_j, $\mathcal{A}$ queries $\mathsf{ct} = (\mathsf{sigvk}, (\mathsf{cva.ct}_i)_{i\in[n]}, \sigma)$ to $O_{\mathsf{Dec},2}$ such that $\mathsf{sigvk} = \mathsf{sigvk}^*$, $\mathsf{ct} \neq \mathsf{ct}^*$, and $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{cva.ct}_1\| \cdots \|\mathsf{cva.ct}_n, \sigma) = \top$.

We also let $\mathtt{Forge}_j = \mathtt{Forge}_{j,1} \vee \mathtt{Forge}_{j,2}$. Hyb_0 and Hyb_1 are identical games unless the events $\mathtt{Forge}_0$ and $\mathtt{Forge}_1$ happen in Hyb_0 and Hyb_1, respectively. Thus, we have $\Pr[1 \leftarrow \mathsf{Hyb}_0 \wedge \neg\mathtt{Forge}_0] = \Pr[1 \leftarrow \mathsf{Hyb}_1 \wedge \neg\mathtt{Forge}_1]$ and $\Pr[\mathtt{Forge}_0] = \Pr[\mathtt{Forge}_1]$. Then, we have

$$\begin{aligned}
|\Pr[1 \leftarrow \mathsf{Hyb}_0] - \Pr[1 \leftarrow \mathsf{Hyb}_1]| &\leq |\Pr[1 \leftarrow \mathsf{Hyb}_0 \wedge \mathtt{Forge}_0] - \Pr[1 \leftarrow \mathsf{Hyb}_1 \wedge \mathtt{Forge}_1]| && (99)\\
&\quad + |\Pr[1 \leftarrow \mathsf{Hyb}_0 \wedge \neg\mathtt{Forge}_0] - \Pr[1 \leftarrow \mathsf{Hyb}_1 \wedge \neg\mathtt{Forge}_1]| && (100)\\
&= |\Pr[1 \leftarrow \mathsf{Hyb}_0 \wedge \mathtt{Forge}_0] - \Pr[1 \leftarrow \mathsf{Hyb}_1 \wedge \mathtt{Forge}_1]| && (101)\\
&\leq \Pr[\mathtt{Forge}_1] \cdot |\Pr[1 \leftarrow \mathsf{Hyb}_0 | \mathtt{Forge}_0] - \Pr[1 \leftarrow \mathsf{Hyb}_1 | \mathtt{Forge}_1]| && (102)\\
&\leq \Pr[\mathtt{Forge}_1] && (103)\\
&\leq \Pr[\mathtt{Forge}_{1,1}] + \Pr[\mathtt{Forge}_{1,2}]. && (104)
\end{aligned}$$

From the strong unforgeability of SIG, we have $\Pr[\mathtt{Forge}_{1,1}] \leq \mathsf{negl}(\lambda)$ and $\Pr[\mathtt{Forge}_{1,2}] \leq \mathsf{negl}(\lambda)$, and thus obtain $|\Pr[1 \leftarrow \mathsf{Hyb}_0] - \Pr[1 \leftarrow \mathsf{Hyb}_1]| \leq \mathsf{negl}(\lambda)$.

Hyb_2**:** This is the same as Hyb_1 except that the challenger generates

$$\mathsf{cva.ct}^*_{i^*} \leftarrow \mathsf{CVA.Enc}(\mathsf{cva.vk}_{i^*,\mathsf{sigvk}^*[i^*]}, \mathsf{cva.pk}'_{i^*,\mathsf{sigvk}^*[i^*]}, 0^\ell) \qquad (105)$$

for randomly chosen $i^* \leftarrow [n]$.

To estimate $|\Pr[1 \leftarrow \mathsf{Hyb}_1] - \Pr[1 \leftarrow \mathsf{Hyb}_2]|$, we construct the following adversary $\mathcal{B}$ that uses $\mathcal{A}$ and attacks the IND-pkT-CVA$^{(1)}$ security of CVA.

1. Given $(\mathsf{cva.vk}, \mathsf{cva.pk})$, $\mathcal{B}$ generates $(\mathsf{sigvk}^*, \mathsf{sigk}^*) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$, picks $i^* \leftarrow [n]$, and sets $\mathsf{cva.vk}_{i^*,\mathsf{sigvk}^*[i^*]} := \mathsf{cva.vk}$ and $\mathsf{cva.pk}_{i^*,\mathsf{sigvk}^*[i^*]} := \mathsf{cva.pk}$. $\mathcal{B}$ generates $(\mathsf{cva.sk}_{i,\alpha}, \mathsf{cva.vk}_{i,\alpha}) \leftarrow \mathsf{CVA.SKGen}(1^\lambda)$ and $\mathsf{cva.pk}_{i,\alpha} \leftarrow \mathsf{CVA.PKGen}(\mathsf{cva.sk}_{i,\alpha})$ for every $(i, \alpha) \in [n]\times\{0, 1\}\backslash\{(i^*, \mathsf{sigvk}^*[i^*])\}$. $\mathcal{B}$ sets $\mathsf{vk} := (\mathsf{cva.vk}_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$ and $\mathsf{pk} := (\mathsf{cva.pk}_{i,\alpha})_{i\in[n],\alpha\in\{0,1\}}$.
2. $\mathcal{B}$ runs $(\mathsf{pk}', \mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{pk}, \mathsf{vk})^{O_{\mathsf{Dec},1}(\cdot)}$, where $O_{\mathsf{Dec},1}(\mathsf{ct})$ is simulated as follows.
    - Parse $\mathsf{ct} := (\mathsf{sigvk}, (\mathsf{cva.ct}_i)_{i\in[n]}, \sigma)$.

- If $\mathsf{sigvk} = \mathsf{sigvk}^*$ or $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{cva.ct}_1 \| \cdots \| \mathsf{cva.ct}_n, \sigma) = \bot$, return $\bot$ with probability $\frac{1}{n}$ and abort with output $\beta' = 0$ with probability $\frac{n-1}{n}$. Otherwise, go to the next step.
- If i^* is not the smallest index i such that $\mathsf{sigvk}[i] \neq \mathsf{sigvk}^*[i]$, abort with output $\beta' = 0$. Otherwise, go to the next step.
- Return $\mathsf{1CCA.Dec}(\mathsf{sk}, \mathsf{ct})$. Note that in this case, $\mathcal{B}$ can compute $\mathsf{1CCA.Dec}(\mathsf{sk}, \mathsf{ct})$ by using $\mathsf{cva.sk}_{i,\alpha}$ for $(i, \alpha) \in [n] \times \{0,1\} \setminus \{(i^*, \mathsf{sigvk}^*[i^*])\}$.

3. $\mathcal{B}$ parses $\mathsf{pk}' := (\mathsf{cva.pk}'_{i,\alpha})_{i \in [n], \alpha \in \{0,1\}}$ and picks $b \leftarrow \{0,1\}$. $\mathcal{B}$ generates ct^* as follows.

   - Generate $u_i^* \leftarrow \{0,1\}^\ell$ for every $i \in [n] \setminus \{i^*\}$ and set $u_{i^*}^* := \mathsf{msg}_b \oplus \bigoplus_{i \in [n] \setminus \{i^*\}} u_i^*$.
   - Run $\mathsf{cva.ct}_i^* \leftarrow \mathsf{CVA.Enc}(\mathsf{cva.vk}_{i,\mathsf{sigvk}^*[i]}, \mathsf{cva.pk}_{i,\mathsf{sigvk}^*[i]}, u_i^*)$ for every $i \in [n] \setminus \{i^*\}$.
   - Output $(\mathsf{cva.pk}'_{i^*,\mathsf{sigvk}^*[i^*]}, u_{i^*}^*, 0^\ell, \mathsf{st}_\mathcal{B})$, where $\mathsf{st}_\mathcal{B}$ includes all information $\mathcal{B}$ knows at this point.
   - Obtain $(\mathsf{ct}_{i^*}^*, \mathsf{cv}_{i^*}, \mathsf{st}_\mathcal{B})$. Set $\mathsf{cva.ct}_{i^*}^* := \mathsf{ct}_{i^*}^*$.
   - Generate $\sigma^* \leftarrow \mathsf{SIG.Sign}(\mathsf{sigk}^*, \mathsf{cva.ct}_1^* \| \cdots \| \mathsf{cva.ct}_n^*)$ and set $\mathsf{ct}^* := (\mathsf{sigvk}^*, (\mathsf{cva.ct}_i^*)_{i \in [n]}, \sigma^*)$.

   $\mathcal{B}$ also sets $\mathsf{cv} = 0$ if $\mathsf{cv}_{i^*} = 0$. Otherwise, $\mathcal{B}$ sets $\mathsf{cv} = 1$ if and only if $\mathsf{CVA.Dec}(\mathsf{cva.sk}_{i,\mathsf{sigvk}^*[i]}, \mathsf{cva.ct}_i^*) \neq \bot$ holds for every $i \in [n] \setminus \{i^*\}$.

4. $\mathcal{B}$ runs $b' \leftarrow \mathcal{A}(\mathsf{ct}^*, \mathsf{cv}, \mathsf{st})^{O_{\mathsf{Dec},2}(\cdot)}$, where $O_{\mathsf{Dec},2}$ is simulated in exactly the same way as $O_{\mathsf{Dec},1}$. $\mathcal{B}$ outputs $\beta' = 1$ if $b = b'$ and otherwise outputs $\beta' = 0$.

We define $\mathtt{Good}$ as the event that $\mathcal{B}$ does not abort when simulating decryption oracles. We also let the challenge bit in the security experiment played by $\mathcal{B}$ be β. $\mathcal{B}$ aborts with probability $n - 1/n$ regardless of the value of β, that is, $\Pr[\mathtt{Good}|\beta = 0] = \Pr[\mathtt{Good}|\beta = 1] = 1/n$ holds. Then, $\mathcal{B}$'s advantage is calculated as follows.

$$|\Pr[\beta' = 1|\beta = 0] - \Pr[\beta' = 1|\beta = 1]| = |\Pr[b = b' \wedge \mathtt{Good}|\beta = 0] - \Pr[b = b' \wedge \mathtt{Good}|\beta = 1]| \tag{106}$$

$$= \frac{1}{n}|\Pr[b = b'|\beta = 0 \wedge \mathtt{Good}] - \Pr[b = b'|\beta = 1 \wedge \mathtt{Good}]| \tag{107}$$

$$= \frac{1}{n}|\Pr[1 \leftarrow \mathsf{Hyb}_1] - \Pr[1 \leftarrow \mathsf{Hyb}_2]|. \tag{108}$$

The second line follows from the fact that we have $\Pr[\mathtt{Good}|\beta = 0] = \Pr[\mathtt{Good}|\beta = 1] = 1/n$ as stated above. The third line follows since $\mathcal{B}$ perfectly simulates Hyb_1 (resp. Hyb_2) conditioned that $\beta = 0$ (resp. $\beta = 1$) and the event $\mathtt{Good}$ occurs. Thus, from the IND-pkT-CVA$^{(1)}$ security of CVA, we have $|\Pr[1 \leftarrow \mathsf{Hyb}_1] - \Pr[1 \leftarrow \mathsf{Hyb}_2]| \leq \mathsf{negl}(\lambda)$.

Clearly, we have $\Pr[1 \leftarrow \mathsf{Hyb}_2] = \frac{1}{2}$. From the above discussions, 1CCA satisfies IND-pkT-1CCA$^{(1)}$ security.

9.4 Boosting IND-pkT-1CCA$^{(1)}$ Security into IND-pkT-CCA$^{(1)}$ Security

We show how to transform IND-pkT-1CCA$^{(1)}$ secure QPKE into IND-pkT-CCA$^{(1)}$ secure one using tokenized MAC.

We construct a QPKE scheme $\mathsf{CCA} = (\mathsf{CCA.SKGen}, \mathsf{CCA.PKGen}, \mathsf{CCA.Enc}, \mathsf{CCA.Dec})$ using the following building blocks.

- A QPKE scheme $\mathsf{1CCA} = (\mathsf{1CCA.SKGen}, \mathsf{1CCA.PKGen}, \mathsf{1CCA.Enc}, \mathsf{1CCA.Dec})$.
- A tokenized MAC scheme $\mathsf{TMAC} = (\mathsf{TMAC.SKGen}, \mathsf{TMAC.TKGen}, \mathsf{TMAC.Sign}, \mathsf{TMAC.Ver})$.
- A signature scheme $\mathsf{SIG} = (\mathsf{SIG.Gen}, \mathsf{SIG.Sign}, \mathsf{SIG.Ver})$.

The construction of CCA is as follows.

- $\mathsf{CCA.SKGen}(1^\lambda) \to (\mathsf{sk}, \mathsf{vk})$: Run $(\mathsf{1cca.sk}, \mathsf{1cca.vk}) \leftarrow \mathsf{1CCA.SKGen}(1^\lambda)$ and $\mathsf{mk} \leftarrow \mathsf{TMAC.SKGen}(1^\lambda)$. Output $\mathsf{sk} := (\mathsf{1cca.sk}, \mathsf{mk})$ and $\mathsf{vk} := \mathsf{1cca.vk}$.
- $\mathsf{CCA.PKGen}(\mathsf{sk}) \to \mathsf{pk}$: Parse $\mathsf{sk} := (\mathsf{1cca.sk}, \mathsf{mk})$. Run $\mathsf{1cca.pk} \leftarrow \mathsf{1CCA.PKGen}(\mathsf{1cca.sk})$ and $\mathsf{token} \leftarrow \mathsf{TMAC.TKGen}(\mathsf{mk})$ and outputs $\mathsf{pk} := (\mathsf{1cca.pk}, \mathsf{token})$.
- $\mathsf{CCA.Enc}(\mathsf{vk}, \mathsf{pk}, \mathsf{msg}) \to \mathsf{ct}$: Parse $\mathsf{vk} := \mathsf{1cca.vk}$ and $\mathsf{pk} := (\mathsf{1cca.pk}, \mathsf{token})$. Run $(\mathsf{sigvk}, \mathsf{sigk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$. Run $\mathsf{1cca.ct} \leftarrow \mathsf{1CCA.Enc}(\mathsf{1cca.vk}, \mathsf{1cca.pk}, \mathsf{sigvk}\|\mathsf{msg})$. Run $\mathsf{tmac}.\sigma \leftarrow \mathsf{TMAC.Sign}(\mathsf{token}, \mathsf{1cca.ct})$. Run $\mathsf{sig}.\sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{sigk}, \mathsf{1cca.ct}\|\mathsf{tmac}.\sigma)$. Output $\mathsf{ct} := (\mathsf{1cca.ct}, \mathsf{tmac}.\sigma, \mathsf{sig}.\sigma)$.
- $\mathsf{CCA.Dec}(\mathsf{sk}, \mathsf{ct}) \to \mathsf{msg}$: Parse $\mathsf{sk} := (\mathsf{1cca.sk}, \mathsf{mk})$ and $\mathsf{ct} = (\mathsf{1cca.ct}, \mathsf{tmac}.\sigma, \mathsf{sig}.\sigma)$. Output $\bot$ if $\mathsf{TMAC.Ver}(\mathsf{mk}, \mathsf{1cca.ct}, \mathsf{tmac}.\sigma) = \bot$ and otherwise go to the next step. Run $\mathsf{sigvk}\|\mathsf{msg} \leftarrow \mathsf{1CCA.Dec}(\mathsf{1cca.sk}, \mathsf{1cca.ct})$, and output $\bot$ if $\mathsf{sigvk}\|\mathsf{msg} = \bot$ or $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{1cca.ct}\|\mathsf{tmac}.\sigma, \mathsf{sig}.\sigma) = \bot$. Otherwise, output msg.

Correctness and strong decryption error detectability. The correctness and the strong decryption error detectability of CCA immediately follow from those of $\mathsf{1CCA}$ and the correctness of TMAC and SIG.

IND-pkT-CCA$^{(1)}$ security. We prove that if $\mathsf{1CCA}$ satisfies IND-pkT-1CCA$^{(1)}$ security, TMAC satisfies unforgeability, and SIG satisfies strong unforgeability, then CCA satisfies IND-pkT-CCA$^{(1)}$ security.

Let $\mathcal{A}$ be any QPT adversary attacking the IND-pkT-CCA$^{(1)}$ security of CCA. We proceed the proof using a sequence of experiments.

Hyb_0**:** This is the original security experiment for the IND-pkT-CCA$^{(1)}$ security of CCA played between $\mathcal{A}$ and the challenger. The detailed description is as follows.

1. The challenger generates $(\mathsf{1cca.sk}, \mathsf{1cca.vk}) \leftarrow \mathsf{1CCA.SKGen}(1^\lambda)$ and $\mathsf{mk} \leftarrow \mathsf{TMAC.SKGen}(1^\lambda)$, and sets $\mathsf{sk} := (\mathsf{1cca.sk}, \mathsf{mk})$ and $\mathsf{vk} := \mathsf{1cca.vk}$. The challenger also generates $\mathsf{1cca.pk} \leftarrow \mathsf{1CCA.PKGen}(\mathsf{1cca.sk})$ and $\mathsf{token} \leftarrow \mathsf{TMAC.TKGen}(\mathsf{mk})$ and sets $\mathsf{pk} := (\mathsf{1cca.pk}, \mathsf{token})$.
2. The challenger runs $(\mathsf{pk}', \mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{vk}, \mathsf{pk})^{O_{\mathsf{Dec},1}(\cdot)}$, where $O_{\mathsf{Dec},1}(\mathsf{ct})$ behaves as follows.
   - Parse $\mathsf{ct} = (\mathsf{1cca.ct}, \mathsf{tmac}.\sigma, \mathsf{sig}.\sigma)$.
   - Output $\bot$ if $\mathsf{TMAC.Ver}(\mathsf{mk}, \mathsf{1cca.ct}, \mathsf{tmac}.\sigma) = \bot$ and otherwise go to the next step.
   - Run $\mathsf{sigvk}\|\mathsf{msg} \leftarrow \mathsf{1CCA.Dec}(\mathsf{1cca.sk}, \mathsf{1cca.ct})$, and output $\bot$ if $\mathsf{sigvk}\|\mathsf{msg} = \bot$ or $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{1cca.ct}\|\mathsf{tmac}.\sigma, \mathsf{sig}.\sigma) = \bot$. Otherwise, output msg.
3. The challenger parses $\mathsf{pk}' := (\mathsf{1cca.pk}', \mathsf{token}')$ and picks $b \leftarrow \{0,1\}$. The challenger generates ct^* as follows.
   - Run $(\mathsf{sigvk}^*, \mathsf{sigk}^*) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$.

- Run $\mathsf{1cca.ct}^* \leftarrow \mathsf{1CCA.Enc}(\mathsf{1cca.vk}, \mathsf{1cca.pk}', \mathsf{sigvk}^* \| \mathsf{msg}_b)$.
- Run $\mathsf{tmac}.\sigma^* \leftarrow \mathsf{TMAC.Sign}(\mathsf{token}', \mathsf{1cca.ct}^*)$.
- Run $\mathsf{sig}.\sigma^* \leftarrow \mathsf{SIG.Sign}(\mathsf{sigk}^*, \mathsf{1cca.ct}^* \| \mathsf{tmac}.\sigma^*)$.
- Set $\mathsf{ct}^* := (\mathsf{1cca.ct}^*, \mathsf{tmac}.\sigma^*, \mathsf{sig}.\sigma^*)$.

The challenger sets $\mathsf{cv} = 0$ if $\mathsf{CCA.Dec}(\mathsf{sk}, \mathsf{ct}^*) = \bot$ and $\mathsf{cv} = 1$ otherwise.

4. The challenger runs $b' \leftarrow \mathcal{A}(\mathsf{ct}^*, \mathsf{cv}, \mathsf{st})^{O_{\mathsf{Dec},2}(\cdot)}$, where $O_{\mathsf{Dec},2}(\mathsf{ct})$ behaves in the same way as $O_{\mathsf{Dec},1}$ except that $O_{\mathsf{Dec},2}$ returns $\bot$ if $\mathsf{ct} = \mathsf{ct}^*$. The challenger outputs 1 if $b = b'$ and otherwise outputs 0.

Hyb_1: This is the same as Hyb_0 except that $O_{\mathsf{Dec},2}$ given $\mathsf{ct} = (\mathsf{1cca.ct}, \mathsf{tmac}.\sigma, \mathsf{sig}.\sigma)$ returns $\bot$ if $\mathsf{1cca.ct} = \mathsf{1cca.ct}^*$.

We define the following events.

$\mathtt{DecError}_j$: In Hyb_j, It holds that $\mathsf{1CCA.Dec}(\mathsf{1cca.sk}, \mathsf{1cca.ct}^*) \notin \{\mathsf{sigvk}^* \| \mathsf{msg}_b, \bot\}$.

$\mathtt{Forge}_j$: In Hyb_j, $\mathcal{A}$ queries $\mathsf{ct} = (\mathsf{1cca.ct}, \mathsf{tmac}.\sigma, \mathsf{sig}.\sigma)$ to $O_{\mathsf{Dec},2}$ such that $\mathsf{1cca.ct} = \mathsf{1cca.ct}^*$, $\mathsf{ct} \neq \mathsf{ct}^*$, and $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{1cca.ct} \| \mathsf{tmac}.\sigma, \mathsf{sig}.\sigma) = \top$, where $\mathsf{sigvk} \| \mathsf{msg} \leftarrow \mathsf{1CCA.Dec}(\mathsf{1cca.sk}, \mathsf{1cca.ct})$.

$O_{\mathsf{Dec},2}$ returns $\bot$ for a queried ciphertext $\mathsf{ct} = (\mathsf{1cca.ct}, \mathsf{tmac}.\sigma, \mathsf{sig}.\sigma)$ such that $\mathsf{1cca.ct} = \mathsf{1cca.ct}^*$ in Hyb_0, unless the event $\mathtt{DecError}_0$ or $\mathtt{Forge}_0$ occur. Thus, we have $|\Pr[1 \leftarrow \mathsf{Hyb}_0] - \Pr[1 \leftarrow \mathsf{Hyb}_1]| \leq \Pr[\mathtt{DecError}_1 \vee \mathtt{Forge}_1] \leq \Pr[\mathtt{DecError}_1] + \Pr[\mathtt{Forge}_1]$. We have $\Pr[\mathtt{DecError}_1] \leq \mathsf{negl}(\lambda)$ from the strong decryption error detectability of 1CCA and $\Pr[\mathtt{Forge}_1] \leq \mathsf{negl}(\lambda)$ from the strong unforgeability of SIG. From these, we obtain $|\Pr[1 \leftarrow \mathsf{Hyb}_0] - \Pr[1 \leftarrow \mathsf{Hyb}_1]| \leq \mathsf{negl}(\lambda)$.

Hyb_2: This is the same as Hyb_1 except that $O_{\mathsf{Dec},1}$ has a state (s,t) that is initially set to $(0,\bot)$ and behaves as follows.

- Parse $\mathsf{ct} = (\mathsf{1cca.ct}, \mathsf{tmac}.\sigma, \mathsf{sig}.\sigma)$.
- If $s = 1$, do the following
  - Output $\bot$ if $\mathsf{1cca.ct} \neq t$. Otherwise, go to the next step.
  - If $\mathsf{TMAC.Ver}(\mathsf{mk}, \mathsf{1cca.ct}, \mathsf{tmac}.\sigma) = \bot$ output $\bot$. Otherwise, go to the next step.
  - Run $\mathsf{sigvk} \| \mathsf{msg} \leftarrow \mathsf{1CCA.Dec}(\mathsf{1cca.sk}, \mathsf{1cca.ct})$, and output $\bot$ if $\mathsf{sigvk} \| \mathsf{msg} = \bot$ or $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{1cca.ct} \| \mathsf{tmac}.\sigma, \mathsf{sig}.\sigma) = \bot$. Otherwise, output msg.
- If $s = 0$, do the following.
  - If $\mathsf{TMAC.Ver}(\mathsf{mk}, \mathsf{1cca.ct}, \mathsf{tmac}.\sigma) = \bot$ output $\bot$. Otherwise, update the state (s,t) into $(1, \mathsf{1cca.ct})$ and go to the next step.
  - Run $\mathsf{sigvk} \| \mathsf{msg} \leftarrow \mathsf{1CCA.Dec}(\mathsf{1cca.sk}, \mathsf{1cca.ct})$, and output $\bot$ if $\mathsf{sigvk} \| \mathsf{msg} = \bot$ or $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{1cca.ct} \| \mathsf{tmac}.\sigma, \mathsf{sig}.\sigma) = \bot$. Otherwise, output msg.

Also, the state (s,t) is passed to $O_{\mathsf{Dec},2}$ at the end of the execution of $O_{\mathsf{Dec},1}$ and $O_{\mathsf{Dec},2}$ behaves in the same way as $O_{\mathsf{Dec},1}$ except that $O_{\mathsf{Dec},2}$ given $\mathsf{ct} = (\mathsf{1cca.ct}, \mathsf{tmac}.\sigma, \mathsf{sig}.\sigma)$ returns $\bot$ if $\mathsf{1cca.ct} = \mathsf{1cca.ct}^*$.

From the unforgeability of TMAC, we have $|\Pr[1 \leftarrow \mathsf{Hyb}_1] - \Pr[1 \leftarrow \mathsf{Hyb}_2]| \leq \mathsf{negl}(\lambda)$.

In Hyb_2, the decryption oracle decrypts at most one queried ciphertext $\mathsf{ct} = (\mathsf{1cca.ct}, \mathsf{tmac}.\sigma, \mathsf{sig}.\sigma)$ such that $\mathsf{1cca.ct} \neq \mathsf{1cca.ct}^*$. Then, from the IND-pkT-1CCA$^{(1)}$ security of 1CCA, we have $\Pr[1 \leftarrow \mathsf{Hyb}_2] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$. From the above discussions, CCA satisfies IND-pkT-CCA$^{(1)}$ security.

9.5 IND-pkT-CCA$^{(1)}$ Security to IND-pkT-CCA Security

We show how to construct IND-pkT-CCA secure QPKE from IND-pkT-CCA$^{(1)}$ secure one.

We construct a QPKE scheme $\mathsf{MKey} = (\mathsf{MKey.SKGen}, \mathsf{MKey.PKGen}, \mathsf{MKey.Enc}, \mathsf{MKey.Dec})$ using the following building blocks.

- A QPKE scheme $\mathsf{1Key} = (\mathsf{1Key.SKGen}, \mathsf{1Key.PKGen}, \mathsf{1Key.Enc}, \mathsf{1Key.Dec})$.
- PRFs $\{\mathsf{PRF}_K\}_{K\in\{0,1\}^\lambda}$.
- A signature scheme $\mathsf{SIG} = (\mathsf{SIG.Gen}, \mathsf{SIG.Sign}, \mathsf{SIG.Ver})$.

The construction of MKey is as follows.

- $\mathsf{MKey.SKGen}(1^\lambda) \to (\mathsf{sk}, \mathsf{vk})$: Generate $K \leftarrow \{0,1\}^\lambda$ and $(\mathsf{sigvk}, \mathsf{sigk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$. Output $\mathsf{sk} := (K, \mathsf{sigk})$ and $\mathsf{vk} := \mathsf{sigvk}$.
- $\mathsf{MKey.PKGen}(\mathsf{sk}) \to \mathsf{pk}$: Parse $\mathsf{sk} := (K, \mathsf{sigk})$ and generate $\mathsf{snum} \leftarrow \{0,1\}^\lambda$. Run $r \leftarrow \mathsf{PRF}_K(\mathsf{snum})$, $(\mathsf{1key.sk}, \mathsf{1key.vk}) \leftarrow \mathsf{1Key.SKGen}(1^\lambda; r)$, and $\mathsf{1key.pk} \leftarrow \mathsf{1Key.PKGen}(\mathsf{1key.sk})$. Run $\sigma \leftarrow \mathsf{SIG.Sign}(\mathsf{sigk}, \mathsf{snum}\|\mathsf{1key.vk})$. Outputs $\mathsf{pk} := (\mathsf{snum}, \mathsf{1key.vk}, \mathsf{1key.pk}, \sigma)$.
- $\mathsf{MKey.Enc}(\mathsf{vk}, \mathsf{pk}, \mathsf{msg}) \to \mathsf{ct}$: Parse $\mathsf{vk} := \mathsf{sigvk}$ and $\mathsf{pk} := (\mathsf{snum}, \mathsf{1key.vk}, \mathsf{1key.pk}, \sigma)$. Output $\bot$ if $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{snum}\|\mathsf{1key}, \sigma) = \bot$ and otherwise go to the next step. Run $\mathsf{1key.ct} \leftarrow \mathsf{1Key.Enc}(\mathsf{1key.vk}, \mathsf{1key.pk}, \mathsf{msg})$. Output $\mathsf{ct} := (\mathsf{snum}, \mathsf{1key.ct})$.
- $\mathsf{MKey.Dec}(\mathsf{sk}, \mathsf{ct}) \to \mathsf{msg}$: Parse $\mathsf{sk} := (K, \mathsf{sigk})$ and $\mathsf{ct} = (\mathsf{snum}, \mathsf{1key.ct})$. Run $r \leftarrow \mathsf{PRF}_K(\mathsf{snum})$ and $(\mathsf{1key.sk}, \mathsf{1key.vk}) \leftarrow \mathsf{1Key.SKGen}(1^\lambda; r)$. Output $\mathsf{msg} \leftarrow \mathsf{1Key.Dec}(\mathsf{1key.sk}, \mathsf{1key.ct})$.

Correctness and strong decryption error detectability. The correctness and the strong decryption error detectability of MKey immediately follow from those of $\mathsf{1Key}$ and the correctness of SIG.

IND-pkT-CCA security. We prove that if $\mathsf{1Key}$ satisfies IND-pkT-CCA$^{(1)}$ security, $\{\mathsf{PRF}_K\}_{K\in\{0,1\}^\lambda}$ is a secure PRF, and SIG satisfies strong unforgeability, then MKey satisfies IND-pkT-CCA security.

Let $\mathcal{A}$ be any QPT adversary attacking the IND-pkT-CCA security of MKey. Let m be a polynomial of λ. We proceed the proof using a sequence of experiments.

Hyb_0**:** This is the original security experiment for the IND-pkT-CCA security of MKey played between $\mathcal{A}$ and the challenger. The detailed description is as follows.

1. The challenger generates $K \leftarrow \{0,1\}^\lambda$ and $(\mathsf{sigvk}, \mathsf{sigk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$, and sets $\mathsf{sk} := (K, \mathsf{sigk})$ and $\mathsf{vk} := \mathsf{sigvk}$. The challenger generates pk_i for every $i \in [m]$ as follows.
   - Generate $\mathsf{snum}_i \leftarrow \{0,1\}^\lambda$.
   - Run $r_i \leftarrow \mathsf{PRF}_K(\mathsf{snum}_i)$, $(\mathsf{1key.sk}_i, \mathsf{1key.vk}_i) \leftarrow \mathsf{1Key.SKGen}(1^\lambda; r_i)$, and $\mathsf{1key.pk}_i \leftarrow \mathsf{1Key.PKGen}(\mathsf{1key.sk}_i)$.
   - Run $\sigma_i \leftarrow \mathsf{SIG.Sign}(\mathsf{sigk}, \mathsf{snum}_i\|\mathsf{1key.vk}_i)$.
   - Set $\mathsf{pk}_i := (\mathsf{snum}_i, \mathsf{1key.vk}_i, \mathsf{1key.pk}_i, \sigma_i)$.
2. The challenger runs $(\mathsf{pk}', \mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{vk}, \mathsf{pk}_1, \cdots, \mathsf{pk}_m)^{O_{\mathsf{Dec},1}(\cdot)}$, where $O_{\mathsf{Dec},1}(\mathsf{ct})$ behaves as follows.

- Parse $\mathsf{ct} = (\mathsf{snum}, \mathsf{1key.ct})$.
- Run $r \leftarrow \mathsf{PRF}_K(\mathsf{snum})$ and $(\mathsf{1key.sk}, \mathsf{1key.vk}) \leftarrow \mathsf{1Key.SKGen}(1^\lambda; r)$.
- Return $\mathsf{msg} \leftarrow \mathsf{1Key.Dec}(\mathsf{1key.sk}, \mathsf{1key.ct})$.

3. The challenger parses $\mathsf{pk}' := (\mathsf{snum}', \mathsf{1key.vk}', \mathsf{1key.pk}', \sigma')$ and picks $b \leftarrow \{0,1\}$. The challenger generates ct^* as follows.
   - Set $\mathsf{ct}^* := \bot$ if $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{snum}' \| \mathsf{1key.vk}', \sigma') = \bot$ and otherwise go to the next step.
   - Run $\mathsf{1key.ct}^* \leftarrow \mathsf{1Key.Enc}(\mathsf{1key.vk}', \mathsf{1key.pk}', \mathsf{msg}_b)$.
   - Set $\mathsf{ct}^* := (\mathsf{snum}', \mathsf{1key.ct}^*)$.

   The challenger also sets $\mathsf{cv} := 0$ if $\mathsf{MKey.Dec}(\mathsf{sk}, \mathsf{ct}^*) = \bot$ and otherwise sets $\mathsf{cv} := 1$.
4. The challenger runs $b' \leftarrow \mathcal{A}(\mathsf{cv}, \mathsf{ct}^*, \mathsf{st})^{O_{\mathsf{Dec},2}(\cdot)}$, where $O_{\mathsf{Dec},2}$ behaves exactly in the same way as $O_{\mathsf{Dec},1}$ except that $O_{\mathsf{Dec},2}$ given ct returns $\bot$ if $\mathsf{ct} = \mathsf{ct}^*$. The challenger outputs 1 if $b = b'$ and otherwise outputs 0.

Hyb_1**:** This is the same as Hyb_1 except that $\mathsf{PRF}_K(\cdot)$ is replaced with a truly random function.

From the security of PRF, we have $|\Pr[1 \leftarrow \mathsf{Hyb}_0] - \Pr[1 \leftarrow \mathsf{Hyb}_1]| \leq \mathsf{negl}(\lambda)$.

Hyb_2**:** This is the same as Hyb_1 except that if $\mathsf{snum}' \| \mathsf{1key.vk}' \neq \mathsf{snum}_i \| \mathsf{1key.vk}_i$ for every $i \in [m]$, the challenger sets $\mathsf{ct}^* := \bot$.

From the strong unforgeability of SIG, we have $|\Pr[1 \leftarrow \mathsf{Hyb}_1] - \Pr[1 \leftarrow \mathsf{Hyb}_2]| \leq \mathsf{negl}(\lambda)$.

Hyb_3**:** This is the same as Hyb_2 except that if $\mathsf{snum}' \| \mathsf{1key.vk}' = \mathsf{snum}_i \| \mathsf{1key.vk}_i$ for some $i \in [m]$, the challenger generates $\mathsf{1key.ct}^* \leftarrow \mathsf{1Key.Enc}(\mathsf{1key.vk}_i, \mathsf{1key.pk}', 0^\ell)$.

To estimate $|\Pr[1 \leftarrow \mathsf{Hyb}_2] - \Pr[1 \leftarrow \mathsf{Hyb}_3]|$, we construct the following adversary $\mathcal{B}$ that uses $\mathcal{A}$ and attacks the IND-pkT-CCA$^{(1)}$ security of 1Key.

1. Given $(\mathsf{1key.vk}, \mathsf{1key.pk})$, $\mathcal{B}$ picks $i^* \leftarrow [m]$, generates $(\mathsf{sigvk}, \mathsf{sigk}) \leftarrow \mathsf{SIG.Gen}(1^\lambda)$, and sets $\mathsf{vk} := \mathsf{sigvk}$. $\mathcal{B}$ then generates $\mathsf{snum}_{i^*} \leftarrow \{0,1\}^\lambda$, sets $\mathsf{1key.vk}_{i^*} := \mathsf{1key.vk}$ and $\mathsf{1key.pk}_{i^*} := \mathsf{1key.pk}$, generates $\sigma_{i^*} \leftarrow \mathsf{SIG.Sign}(\mathsf{sigk}, \mathsf{snum}_{i^*} \| \mathsf{1key.vk}_{i^*})$, and sets $\mathsf{pk}_{i^*} := (\mathsf{snum}_{i^*}, \mathsf{1key.vk}_{i^*}, \mathsf{1key.pk}_{i^*}, \sigma_{i^*})$. $\mathcal{B}$ prepares an empty list KL. $\mathcal{B}$ does the following for every $i \in [m] \setminus \{i^*\}$.
   - Generate $\mathsf{snum}_i \leftarrow \{0,1\}^\lambda$ and $r_i \leftarrow \{0,1\}^\lambda$.
   - Run $(\mathsf{1key.sk}_i, \mathsf{1key.vk}_i) \leftarrow \mathsf{1Key.SKGen}(1^\lambda; r_i)$ and $\mathsf{1key.pk}_i \leftarrow \mathsf{1Key.PKGen}(\mathsf{1key.sk}_i)$.
   - Run $\sigma_i \leftarrow \mathsf{SIG.Sign}(\mathsf{sigk}, \mathsf{snum}_i \| \mathsf{1key.vk}_i)$.
   - Set $\mathsf{pk}_i := (\mathsf{snum}_i, \mathsf{1key.vk}_i, \mathsf{1key.pk}_i, \sigma_i)$ and adds $(\mathsf{snum}_i, \mathsf{1key.sk}_i)$ to KL.
2. $\mathcal{B}$ runs $(\mathsf{pk}', \mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{vk}, \mathsf{pk}_1, \cdots, \mathsf{pk}_m)^{O_{\mathsf{Dec},1}(\cdot)}$, where $O_{\mathsf{Dec},1}(\mathsf{ct})$ is simulated as follows.
   - Parse $\mathsf{ct} := (\mathsf{snum}, \mathsf{1key.ct})$.
   - If $\mathsf{snum} = \mathsf{snum}_{i^*}$, queries $\mathsf{1key.ct}$ to its decryption oracle and forwards the answer to $\mathcal{A}$. Otherwise, go to the next step.
   - If there exists an entry of the form $(\mathsf{snum}, \mathsf{1key.sk})$ in KL, return $\mathsf{1Key.Dec}(\mathsf{1key.sk}, \mathsf{1key.ct})$ to $\mathcal{A}$. Otherwise, go to the next step.

- Generate $r \leftarrow \{0,1\}^\lambda$ and $(\mathsf{1key.vk}, \mathsf{1key.sk}) \leftarrow \mathsf{1Key.SKGen}(1^\lambda; r)$, and add $(\mathsf{snum}, \mathsf{1key.sk})$ to KL.
- Return $\mathsf{1Key.Dec}(\mathsf{1key.sk}, \mathsf{1key.ct})$ to $\mathcal{A}$.

3. $\mathcal{B}$ parses $\mathsf{pk}' := (\mathsf{snum}', \mathsf{1key.vk}', \mathsf{1key.pk}', \sigma')$ and picks $b \leftarrow \{0,1\}$. $\mathcal{B}$ does the following.
   - Set $\mathsf{ct}^* := \bot$ if $\mathsf{SIG.Ver}(\mathsf{sigvk}, \mathsf{snum}' \| \mathsf{1key.vk}', \sigma') = \bot$ and otherwise go to the next step.
   - Set $\mathsf{ct}^* := \bot$ if $\mathsf{snum}' \| \mathsf{1key.vk}' \neq \mathsf{snum}_i \| \mathsf{1key.vk}_i$ for every $i \in [m]$. Otherwise, go to the next step.
   - Abort with $\beta' := 0$ if $\mathsf{snum}' \| \mathsf{1key.vk}' \neq \mathsf{snum}_{i^*} \| \mathsf{1key.vk}_{i^*}$. Otherwise, go to the next step.
   - Output $(\mathsf{1key.pk}', \mathsf{msg}_b, 0^\ell, \mathsf{st}_{\mathcal{B}})$, where $\mathsf{st}_{\mathcal{B}}$ is all information that $\mathcal{B}$ knows at this point.
   - Obtain $(\mathsf{1key.ct}^*, \mathsf{cv}, \mathsf{st}_{\mathcal{B}})$.
   - Set $\mathsf{ct}^* := (\mathsf{snum}', \mathsf{1key.ct}^*)$.
4. $\mathcal{B}$ runs $b' \leftarrow \mathcal{A}(\mathsf{cv}, \mathsf{ct}^*, \mathsf{st})^{O_{\mathsf{Dec},2}(\cdot)}$, where $O_{\mathsf{Dec},2}$ is simulated exactly in the same way as $O_{\mathsf{Dec},1}$ except that $O_{\mathsf{Dec},2}$ given ct returns $\bot$ if $\mathsf{ct} = \mathsf{ct}^*$. $\mathcal{B}$ outputs $\beta' := 1$ if $b = b'$ and otherwise outputs $\beta' := 0$.

We define $\mathtt{Good}$ as the event that $\mathcal{B}$ does not abort when generating the challenge ciphertext. Then, letting the challenge bit in the security experiment played by $\mathcal{B}$ be β, $\mathcal{B}$'s advantage is calculated as follows.

$$
\begin{aligned}
|\Pr[\beta' = 1 | \beta = 0] - \Pr[\beta' = 1 | \beta = 1]| &= |\Pr[b = b' \wedge \mathtt{Good} | \beta = 0] - \Pr[\beta' = 1 \wedge \mathtt{Good} | \beta = 1]| \quad (109)\\
&\geq \frac{1}{m} |\Pr[b = b' | \beta = 0 \wedge \mathtt{Good}] - \Pr[\beta' = 1 | \beta = 1 \wedge \mathtt{Good}]| \quad (110)\\
&= \frac{1}{m} |\Pr[1 \leftarrow \mathsf{Hyb}_2] - \Pr[1 \leftarrow \mathsf{Hyb}_3]|. \quad (111)
\end{aligned}
$$

The second line follows from the fact that we have $\Pr[\mathtt{Good} | \beta = 0] = \Pr[\mathtt{Good} | \beta = 1] \geq \frac{1}{m}$. The third line follows since $\mathcal{B}$ perfectly simulates Hyb_2 (resp. Hyb_3) conditioned that $\beta = 0$ (resp. $\beta = 1$) and the event $\mathtt{Good}$ occurs. Thus, from the IND-pkT-CCA$^{(1)}$ security of CCA, we have $|\Pr[1 \leftarrow \mathsf{Hyb}_2] - \Pr[1 \leftarrow \mathsf{Hyb}_3]| \leq \mathsf{negl}(\lambda)$.

Clearly, we have $\Pr[1 \leftarrow \mathsf{Hyb}_3] = \frac{1}{2}$. From the above discussions, MKey satisfies IND-pkT-CCA security.

10 Recyclable Variants

In the construction given in Sections 6 and 9, a quantum public key can be used to encrypt only one message and a sender needs to obtain a new quantum public key whenever it encrypts a message. This is not desirable from practical perspective. In this section, we define recyclable QPKE where a sender only needs to receive one quantum public key to send arbitrarily many messages, and then show how to achieve it.

10.1 Definitions

The definition is similar to QPKE as defined in Definition 5.1 except that the encryption algorithm outputs a classical *recycled* key that can be reused to encrypt messages many times.

Definition 10.1 (Recyclable QPKE). *A recyclable QPKE scheme with message space $\{0,1\}^\ell$ is a set of algorithms* $(\mathsf{SKGen}, \mathsf{PKGen}, \mathsf{Enc}, \mathsf{rEnc}, \mathsf{Dec})$ *such that*

- $\mathsf{SKGen}(1^\lambda) \to (\mathsf{sk}, \mathsf{vk})$: *It is a PPT algorithm that, on input the security parameter λ, outputs a classical secret key* sk *and a classical verification key* vk.
- $\mathsf{PKGen}(\mathsf{sk}) \to \mathsf{pk}$: *It is a QPT algorithm that, on input* sk, *outputs a quantum public key* pk.
- $\mathsf{Enc}(\mathsf{vk}, \mathsf{pk}, \mathsf{msg}) \to (\mathsf{ct}, \mathsf{rk})$: *It is a QPT algorithm that, on input* vk, pk, *and a plaintext* $\mathsf{msg} \in \{0,1\}^\ell$, *outputs a classical ciphertext* ct *and classical recycled key* rk.
- $\mathsf{rEnc}(\mathsf{rk}, \mathsf{msg}) \to \mathsf{ct}$: *It is a PPT algorithm that, on input* rk *and a plaintext* $\mathsf{msg} \in \{0,1\}^\ell$, *outputs a classical ciphertext* ct.
- $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \to \mathsf{msg}'$: *It is a classical deterministic polynomial-time algorithm that, on input* sk *and* ct, *outputs* $\mathsf{msg}' \in \{0,1\}^\ell \cup \{\bot\}$.

We require the following correctness.

Correctness: *For any* $\mathsf{msg}, \mathsf{msg}' \in \{0,1\}^\ell$,

$$\Pr\left[\mathsf{msg} \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) \wedge \mathsf{msg}' \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}') : \begin{array}{r} (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SKGen}(1^\lambda) \\ \mathsf{pk} \leftarrow \mathsf{PKGen}(\mathsf{sk}) \\ (\mathsf{ct}, \mathsf{rk}) \leftarrow \mathsf{Enc}(\mathsf{vk}, \mathsf{pk}, \mathsf{msg}) \\ \mathsf{ct}' \leftarrow \mathsf{rEnc}(\mathsf{rk}, \mathsf{msg}') \end{array}\right] \geq 1 - \mathsf{negl}(\lambda). \quad (112)$$

Definition 10.2 (IND-pkT-CPA Security for Recyclable QPKE). *We require the followings.*

Security under quantum public keys: *For any polynomial m, and any QPT adversary $\mathcal{A}$,*

$$\Pr\left[b \leftarrow \mathcal{A}^{\mathsf{rEnc}(\mathsf{rk},\cdot)}(\mathsf{ct}^*, \mathsf{st}) : \begin{array}{r} (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SKGen}(1^\lambda) \\ \mathsf{pk}_1, ..., \mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m} \\ (\mathsf{pk}', \mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{vk}, \mathsf{pk}_1, ..., \mathsf{pk}_m) \\ b \leftarrow \{0,1\} \\ (\mathsf{ct}^*, \mathsf{rk}) \leftarrow \mathsf{Enc}(\mathsf{vk}, \mathsf{pk}', \mathsf{msg}_b) \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda). \quad (113)$$

Security under recycled keys: *For any polynomial m, and any QPT adversary $\mathcal{A}$,*

$$\Pr\left[b \leftarrow \mathcal{A}^{\mathsf{rEnc}(\mathsf{rk},\cdot)}(\mathsf{ct}^*, \mathsf{st}') : \begin{array}{r} (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SKGen}(1^\lambda) \\ \mathsf{pk}_1, ..., \mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m} \\ (\mathsf{pk}', \mathsf{msg}, \mathsf{st}) \leftarrow \mathcal{A}(\mathsf{vk}, \mathsf{pk}_1, ..., \mathsf{pk}_m) \\ (\mathsf{ct}', \mathsf{rk}) \leftarrow \mathsf{Enc}(\mathsf{vk}, \mathsf{pk}', \mathsf{msg}) \\ (\mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}') \leftarrow \mathcal{A}^{\mathsf{rEnc}(\mathsf{rk},\cdot)}(\mathsf{ct}', \mathsf{st}) \\ b \leftarrow \{0,1\} \\ \mathsf{ct}^* \leftarrow \mathsf{rEnc}(\mathsf{rk}, \mathsf{msg}_b) \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda). \quad (114)$$

Here, $\mathsf{pk}_1, ..., \mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m}$ *means that* PKGen *is executed* m *times and* pk_i *is the output of the* i*th execution of* PKGen*,* $\mathsf{rEnc}(\mathsf{rk}, \cdot)$ *means a classically-accessible encryption oracle, and* st *and* st' *are quantum internal states of* $\mathcal{A}$*, which can be entangled with* pk'*.*

Definition 10.3 (IND-pkT-CCA Security for Recyclable QPKE). *We require the followings.*

Security under quantum public keys: *For any polynomial* m*, and any QPT adversary* $\mathcal{A}$*,*

$$\Pr\left[b \leftarrow \mathcal{A}^{\mathsf{rEnc}(\mathsf{rk},\cdot),O_{\mathsf{Dec},2}(\cdot)}(\mathsf{ct}^*, \mathsf{cv}, \mathsf{st}) : \begin{array}{r} (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SKGen}(1^\lambda) \\ \mathsf{pk}_1, ..., \mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m} \\ (\mathsf{pk}', \mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}) \leftarrow \mathcal{A}^{O_{\mathsf{Dec},1}(\cdot)}(\mathsf{vk}, \mathsf{pk}_1, ..., \mathsf{pk}_m) \\ b \leftarrow \{0,1\} \\ (\mathsf{ct}^*, \mathsf{rk}) \leftarrow \mathsf{Enc}(\mathsf{vk}, \mathsf{pk}', \mathsf{msg}_b) \\ \mathsf{cv} := 0 \textit{ if } \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}^*) = \bot \textit{ and otherwise } \mathsf{cv} := 1 \end{array} \right] \leq \frac{1}{2} + \mathsf{negl}(\lambda). \tag{115}$$

Security under recycled keys: *For any polynomial* m*, and any QPT adversary* $\mathcal{A}$*,*

$$\Pr\left[b \leftarrow \mathcal{A}^{\mathsf{rEnc}(\mathsf{rk},\cdot),O_{\mathsf{Dec},2}(\cdot)}(\mathsf{ct}^*, \mathsf{cv}, \mathsf{st}') : \begin{array}{r} (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{SKGen}(1^\lambda) \\ \mathsf{pk}_1, ..., \mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m} \\ (\mathsf{pk}', \mathsf{msg}, \mathsf{st}) \leftarrow \mathcal{A}^{O_{\mathsf{Dec},1}(\cdot)}(\mathsf{vk}, \mathsf{pk}_1, ..., \mathsf{pk}_m) \\ (\mathsf{ct}', \mathsf{rk}) \leftarrow \mathsf{Enc}(\mathsf{vk}, \mathsf{pk}', \mathsf{msg}) \\ (\mathsf{msg}_0, \mathsf{msg}_1, \mathsf{st}') \leftarrow \mathcal{A}^{\mathsf{rEnc}(\mathsf{rk},\cdot),O_{\mathsf{Dec},1}(\cdot)}(\mathsf{ct}', \mathsf{st}) \\ b \leftarrow \{0,1\} \\ \mathsf{ct}^* \leftarrow \mathsf{rEnc}(\mathsf{rk}, \mathsf{msg}_b) \\ \mathsf{cv} := 0 \textit{ if } \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}^*) = \bot \textit{ and otherwise } \mathsf{cv} := 1 \end{array} \right] \leq \frac{1}{2} + \mathsf{negl}(\lambda). \tag{116}$$

Here, $\mathsf{pk}_1, ..., \mathsf{pk}_m \leftarrow \mathsf{PKGen}(\mathsf{sk})^{\otimes m}$ *means that* PKGen *is executed* m *times and* pk_i *is the output of the* i*th execution of* PKGen*,* $\mathsf{rEnc}(\mathsf{rk}, \cdot)$ *means a classically-accessible encryption oracle, and* st *and* st' *are quantum internal states of* $\mathcal{A}$*, which can be entangled with* pk'*. Also,* $O_{\mathsf{Dec},1}(\mathsf{ct})$ *returns* $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ *for any* ct*.* $O_{\mathsf{Dec},2}$ *behaves identically to* $O_{\mathsf{Dec},1}$ *except that* $O_{\mathsf{Dec},2}$ *returns* $\bot$ *to the input* ct^**.*

10.2 Construction

We show a generic construction of recyclable QPKE from (non-recyclable) QPKE with classical ciphertexts and SKE via standard hybrid encryption.

Let $\mathsf{QPKE} = (\mathsf{QPKE.SKGen}, \mathsf{QPKE.PKGen}, \mathsf{QPKE.Enc}, \mathsf{QPKE.Dec})$ be a (non-recyclable) QPKE scheme with message space $\{0,1\}^\lambda$ and $\mathsf{SKE} = (\mathsf{SKE.Enc}, \mathsf{SKE.Dec})$ be an SKE scheme with message space $\{0,1\}^\ell$. Then we construct a recyclable QPKE scheme $\mathsf{QPKE}' = (\mathsf{QPKE}'.\mathsf{SKGen}, \mathsf{QPKE}'.\mathsf{PKGen}, \mathsf{QPKE}'.\mathsf{Enc}, \mathsf{QPKE}'.\mathsf{rEnc}, \mathsf{QPKE}'.\mathsf{Dec})$ with message space $\{0,1\}^\ell$ as follows:

- $\mathsf{QPKE}'.\mathsf{SKGen}(1^\lambda) \to (\mathsf{sk}', \mathsf{vk}')$: Run $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{QPKE.SKGen}(1^\lambda)$ and output a secret key $\mathsf{sk}' := \mathsf{sk}$ and verification key $\mathsf{vk}' := \mathsf{vk}$.
- $\mathsf{QPKE}'.\mathsf{PKGen}(\mathsf{sk}') \to \mathsf{pk}'$: Run $\mathsf{pk} \leftarrow \mathsf{QPKE.PKGen}(\mathsf{sk})$ and outputs $\mathsf{pk}' := \mathsf{pk}$.
- $\mathsf{QPKE}'.\mathsf{Enc}(\mathsf{vk}', \mathsf{pk}', \mathsf{msg}) \to (\mathsf{ct}', \mathsf{rk}')$: Parse $\mathsf{pk}' = \mathsf{pk}$ and $\mathsf{vk}' = \mathsf{vk}$, sample $K \leftarrow \{0,1\}^\lambda$, run $\mathsf{ct} \leftarrow \mathsf{QPKE.Enc}(\mathsf{vk}, \mathsf{pk}, K)$ and $\mathsf{ct}_{\mathsf{ske}} \leftarrow \mathsf{SKE.Enc}(K, \mathsf{msg})$, and output a ciphertext $\mathsf{ct}' := (\mathsf{ct}, \mathsf{ct}_{\mathsf{ske}})$ and recycled key $\mathsf{rk}' := (K, \mathsf{ct})$.

- $\mathsf{QPKE}'.\mathsf{rEnc}(\mathsf{rk}', \mathsf{msg}) \to \mathsf{ct}'$: Parse $\mathsf{rk}' = (K, \mathsf{ct})$, run $\mathsf{ct}_{\mathsf{ske}} \leftarrow \mathsf{SKE}.\mathsf{Enc}(K, \mathsf{msg})$, and output a ciphertext $\mathsf{ct}' := (\mathsf{ct}, \mathsf{ct}_{\mathsf{ske}})$.
- $\mathsf{QPKE}'.\mathsf{Dec}(\mathsf{sk}', \mathsf{ct}') \to \mathsf{msg}'$: Parse $\mathsf{ct}' = (\mathsf{ct}, \mathsf{ct}_{\mathsf{ske}})$ and $\mathsf{sk}' = \mathsf{sk}$, run $K' \leftarrow \mathsf{QPKE}.\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ and $\mathsf{msg}' \leftarrow \mathsf{SKE}.\mathsf{Dec}(K', \mathsf{ct}_{\mathsf{ske}})$, and output msg'.

Correctness and decryption error detectability. Correctness of QPKE' immediately follows from correctness of QPKE and SKE. Also, the decryption error detectability of QPKE' directly follows from that of QPKE.

IND-pkT-CPA security and IND-pkT-CCA security. If QPKE satisfies IND-pkT-CPA (resp. IND-pkT-CCA) security and SKE satisfies IND-CPA (resp. IND-CCA) security, then QPKE' satisfies IND-pkT-CPA (resp. IND-pkT-CCA) security. The proofs can be done by standard hybrid arguments, thus omitted.

Acknowledgments. TM is supported by JST Moonshot R&D JPMJMS2061-5-1-1, JST FOREST, MEXT QLEAP, the Grant-in-Aid for Scientific Research (B) No.JP19H04066, the Grant-in Aid for Transformative Research Areas (A) 21H05183, and the Grant-in-Aid for Scientific Research (A) No.22H00522.

References

[AAS20] Scott Aaronson, Yosi Atia, and Leonard Susskind. On the hardness of detecting macroscopic superpositions. *Electron. Colloquium Comput. Complex.*, page 146, 2020. (Cited on page 3, 5, 6.)

[AGM21] Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. Can you sign a quantum state? *Quantum*, 2021. (Cited on page 1.)

[AGQY22] Prabhanjan Ananth, Aditya Gulati, Luowen Qian, and Henry Yuen. Pseudorandom (function-like) quantum state generators: New definitions and applications. TCC, 2022. (Cited on page 1.)

[AQY22] Prabhanjan Ananth, Luowen Qian, and Henry Yuen. Cryptography from pseudorandom quantum states. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 208–236. Springer, Heidelberg, August 2022. (Cited on page 1.)

[BB84] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *IEEE International Conference on Computers Systems and Signal Processing*, pages 175–179. IEEE, 1984. (Cited on page 1, 11.)

[BCG$^+$02] Howard Barnum, Claude Crépeau, Daniel Gottesman, Adam Smith, and Alain Tapp. Authentication of quantum messages. In *43rd FOCS*, pages 449–458. IEEE Computer Society Press, November 2002. (Cited on page 1.)

[BCKM21] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. One-way functions imply secure computation in a quantum world. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 467–496, Virtual Event, August 2021. Springer, Heidelberg. (Cited on page 1.)

[BCQ23] Zvika Brakerski, Ran Canetti, and Luowen Qian. On the computational hardness needed for quantum cryptography. ITCS 2023: 14th Innovations in Theoretical Computer Science, 2023. (Cited on page 1.)

[BGH+23] Khashayar Barooti, Alex B. Grilo, Loïs Huguenin-Dumittan, Giulio Malavolta, Or Sattath, Quoc-Huy Vu, and Michael Walter. Public-key encryption with quantum keys. *IACR Cryptol. ePrint Arch.*, page 877, 2023. (Cited on page 1, 2, 3, 4, 8, 10, 11.)

[BN08] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Journal of Cryptology*, 21(4):469–491, October 2008. (Cited on page 14.)

[BSS21] Amit Behera, Or Sattath, and Uriel Shinar. Noise-tolerant quantum tokens for MAC. Cryptology ePrint Archive, Report 2021/1353, 2021. https://eprint.iacr.org/2021/1353. (Cited on page 9, 26.)

[BZ13] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 361–379. Springer, Heidelberg, August 2013. (Cited on page 7, 14.)

[CHH+07] Ronald Cramer, Goichiro Hanaoka, Dennis Hofheinz, Hideki Imai, Eike Kiltz, Rafael Pass, abhi shelat, and Vinod Vaikuntanathan. Bounded CCA2-secure encryption. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 502–518. Springer, Heidelberg, December 2007. (Cited on page 9.)

[Col23] Andrea Coladangelo. Quantum trapdoor functions from classical one-way functions. Cryptology ePrint Archive, Paper 2023/282, 2023. https://eprint.iacr.org/2023/282. (Cited on page 1, 2, 3, 10, 11.)

[CX22] Shujiao Cao and Rui Xue. On constructing one-way quantum state generators, and more. Cryptology ePrint Archive, Report 2022/1323, 2022. https://eprint.iacr.org/2022/1323. (Cited on page 1.)

[DS15] Nico Döttling and Dominique Schröder. Efficient pseudorandom functions via on-the-fly adaptation. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 329–350. Springer, Heidelberg, August 2015. (Cited on page 45.)

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986. (Cited on page 14.)

[GKM+00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st FOCS*, pages 325–335. IEEE Computer Society Press, November 2000. (Cited on page 1.)

[GLSV21] Alex B. Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in MiniQCrypt. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 531–561. Springer, Heidelberg, October 2021. (Cited on page 1.)

[Gol04] Oded Goldreich. Foundations of cryptography: Volume 2, basic applications. 2004. (Cited on page 13, 42.)

[Got] Daniel Gottesman. Quantum public-key cryptography with information-theoretic security. *https://www2.perimeterinstitute.ca/personal/dgottesman/Public-key.ppt*. (Cited on page 10.)

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. (Cited on page 14.)

[HMY22] Minki Hhan, Tomoyuki Morimae, and Takashi Yamakawa. From the hardness of detecting superpositions to cryptography: Quantum public key encryption and commitments. *arXiv:2210.05978*, 2022. (Cited on page 3, 5, 17.)

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM STOC*, pages 44–61. ACM Press, May 1989. (Cited on page 1.)

[IR90] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 8–26. Springer, Heidelberg, August 1990. (Cited on page 1.)

[JLS18] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 126–152. Springer, Heidelberg, August 2018. (Cited on page 1.)

[KKNY05] Akinori Kawachi, Takeshi Koshiba, Harumichi Nishimura, and Tomoyuki Yamakami. Computational indistinguishability between quantum states and its cryptographic application. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 268–284. Springer, Heidelberg, May 2005. (Cited on page 10.)

[KQST22] William Kretschmer, Luowen Qian, Makrand Sinha, and Avishay Tal. Quantum cryptography in algorithmica. *arXiv:2212.00879*, 2022. (Cited on page 1.)

[Kre21] W. Kretschmer. Quantum pseudorandomness and classical complexity. *TQC 2021*, 2021. (Cited on page 1.)

[MW23] Giulio Malavolta and Michael Walter. Robust quantum public-key encryption with applications to quantum key distribution. 2023. (Cited on page 12.)

[MY22a] Tomoyuki Morimae and Takashi Yamakawa. One-wayness in quantum cryptography. Cryptology ePrint Archive, Report 2022/1336, 2022. `https://eprint.iacr.org/2022/1336`. (Cited on page 1, 2, 10, 11.)

[MY22b] Tomoyuki Morimae and Takashi Yamakawa. Quantum commitments and signatures without one-way functions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 269–295. Springer, Heidelberg, August 2022. (Cited on page 1.)

[YZ21] Takashi Yamakawa and Mark Zhandry. Classical vs quantum random oracles. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 568–597. Springer, Heidelberg, October 2021. (Cited on page 42, 46.)

[Zha12] Mark Zhandry. How to construct quantum random functions. In *53rd FOCS*, pages 679–687. IEEE Computer Society Press, October 2012. (Cited on page 13.)

A Pure State Public Key Variant

As discussed in Section 1.3, we believe that the distinction between pure state and mixed state public keys is not important from a practical point of view. Nonetheless, it is a mathematically valid question if we can construct an IND-pkT-CPA secure QPKE scheme with pure state public keys. We give such a scheme based on the existence of quantum-secure OWFs by extending the construction given in Section 6. For the ease of exposition, we first show a construction based on *slightly superpolynomially secure* OWFs in Appendix A.1. Then, we explain how to modify the scheme to base its security on standard polynomially secure OWFs in Appendix A.2.

Preparation. We define a fine-grained version of strong EUF-CMA security for digital signature schemes.

Definition A.1 (**T-strong EUF-CMA security**)**.** *A digital signature scheme* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *is T-strong EUF-CMA secure if the following holds: For any quantum adversary $\mathcal{A}$ that runs in time T and makes at most T classical queries to the signing oracle* $\mathsf{Sign}(k, \cdot)$*,*

$$\Pr[(\mathsf{msg}^*, \sigma^*) \notin \mathcal{Q} \ \wedge\ \top \leftarrow \mathsf{Ver}(\mathsf{vk}, \mathsf{msg}^*, \sigma^*) : (k, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda), (\mathsf{msg}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(k,\cdot)}(\mathsf{vk})] \leq T^{-1}, \tag{117}$$

where $\mathcal{Q}$ is the set of message-signature pairs returned by the signing oracle.

Remark A.2. Strong EUF-CMA security defined in Definition 4.1 holds if and only if T-strong EUF-CMA security holds for all polynomials T.

Remark A.3. We can show that there exists a T-strong EUF-CMA secure digital signature scheme for some $T = \lambda^{\omega(1)}$ if slightly superpolynomially secure OWFs exist similarly to the proof of Theorem 4.3 in [Gol04, Sec. 6.5.2]. Here, a superpolynomially secure OWF is a function f for which there exists $T = \lambda^{\omega(1)}$ such that any adversary with running time T can invert f with probability at most T^{-1}.

We also need the following lemma in the security proof.

Lemma A.4. *For a function $H : \{0,1\}^{u+1} \to \{0,1\}^v$, let $|\psi_H\rangle := \sum_{R \in \{0,1\}^{u+1}} |R\rangle |H(R)\rangle$. For any integer m and (unbounded-time) quantum algorithm $\mathcal{A}$,*

$$\Pr_H[y_0 = H(0\|r) \ \wedge\ y_1 = H(1\|r) : (r, y_0, y_1) \leftarrow \mathcal{A}(|\psi_H\rangle^{\otimes m})] \leq (2m+1)^4 (2^{-u} + 2^{-v}) \tag{118}$$

where H is a uniformly random function from $\{0,1\}^{u+1}$ to $\{0,1\}^v$.

We prove it using the result of [YZ21]. We defer the proof to Appendix A.3.

A.1 Construction from Slightly Superpolynomially Secure OWFs

In this section, we construct a QPKE scheme that satisfies correctness and IND-pkT-CPA security (but not decryption error detectability) and has pure state public keys from T-strong EUF-CMA secure digital signatures for a superpolynomial T and quantum-query secure PRFs. Note that they exist assuming the existence of slightly superpolynomially secure OWFs as noted in Remark A.3 and Theorem 4.5. The message space of our construction is $\{0,1\}$, but it can be extended to be arbitrarily many bits by parallel repetition. Let $(\mathsf{Gen},\mathsf{Sign},\mathsf{Ver})$ be a T-strong EUF-CMA secure digital signature scheme with a deterministic Sign algorithm and message space $\{0,1\}^{u+v+1}$ and $\{\mathsf{PRF}_K:\{0,1\}^{u+1}\to\{0,1\}^v\}_{K\in\{0,1\}^\lambda}$ be a quantum-query secure PRF where $T=\lambda^{\omega(1)}$, $u:=\lfloor(\log T)/2\rfloor$, and $v=\omega(\log\lambda)$.

Then we construct a QPKE scheme $(\mathsf{SKGen},\mathsf{PKGen},\mathsf{Enc},\mathsf{Dec})$ as follows.

- $\mathsf{SKGen}(1^\lambda)\to(\mathsf{sk},\mathsf{vk})$: Run $(k,\mathsf{vk})\leftarrow\mathsf{Gen}(1^\lambda)$ and sample $K\leftarrow\{0,1\}^\lambda$. Output $\mathsf{sk}:=(k,K)$ and vk.
- $\mathsf{PKGen}(\mathsf{sk})\to\mathsf{pk}$: Parse $\mathsf{sk}=(k,K)$. Choose $r\leftarrow\{0,1\}^u$. By running Sign and PRF coherently, generate the state

$$|\psi_{\mathsf{sk}}\rangle:=\sum_{r\in\{0,1\}^u}|r\rangle_{\mathbf{R}}\otimes\left(\begin{array}{c}|0\rangle_{\mathbf{A}}\otimes|y(0,r)\rangle_{\mathbf{B}}\otimes|\sigma(0,r)\rangle_{\mathbf{C}}\\+|1\rangle_{\mathbf{A}}\otimes|y(1,r)\rangle_{\mathbf{B}}\otimes|\sigma(1,r)\rangle_{\mathbf{C}}\end{array}\right)\tag{119}$$

  over registers $(\mathbf{R},\mathbf{A},\mathbf{B},\mathbf{C})$ where $y(b,r):=\mathsf{PRF}_K(b\|r)$ and $\sigma(b,r):=\mathsf{Sign}(k,b\|r\|y(b,r))$ for $b\in\{0,1\}$ and $r\in\{0,1\}^u$. (We omit K and k from the notations for simplicity.) Output

$$\mathsf{pk}:=|\psi_{\mathsf{sk}}\rangle\,.\tag{120}$$

- $\mathsf{Enc}(\mathsf{vk},\mathsf{pk},b)\to\mathsf{ct}$: Parse $\mathsf{pk}=\rho$, where ρ is a quantum state over registers $(\mathbf{R},\mathbf{A},\mathbf{B},\mathbf{C})$. The Enc algorithm consists of the following three steps.

  1. It coherently checks the signature in ρ. In other words, it applies the unitary

$$U_{\mathsf{vk}}|r\rangle_{\mathbf{R}}|\alpha\rangle_{\mathbf{A}}|\beta\rangle_{\mathbf{B}}|\gamma\rangle_{\mathbf{C}}|0...0\rangle_{\mathbf{E}}=|r\rangle_{\mathbf{R}}|\alpha\rangle_{\mathbf{A}}|\beta\rangle_{\mathbf{B}}|\gamma\rangle_{\mathbf{C}}|\mathsf{Ver}(\mathsf{vk},\alpha\|r\|\beta,\gamma)\rangle_{\mathbf{E}}\tag{121}$$

     on $\rho_{\mathbf{R},\mathbf{A},\mathbf{B},\mathbf{C}}\otimes|0...0\rangle\langle0...0|_{\mathbf{E}}$, and measures the register $\mathbf{E}$ in the computational basis. If the result is $\bot$, it outputs $\mathsf{ct}:=\bot$ and halts. If the result is $\top$, it goes to the next step.
  2. It applies Z^b on the register $\mathbf{A}$.
  3. It measures $\mathbf{R}$ in the computational basis to get r and all qubits in the registers $(\mathbf{A},\mathbf{B},\mathbf{C})$ in the Hadamard basis to get the result d. It outputs

$$\mathsf{ct}:=(r,d).\tag{122}$$

- $\mathsf{Dec}(\mathsf{sk},\mathsf{ct})\to b'$: Parse $\mathsf{sk}=(k,K)$ and $\mathsf{ct}=(r,d)$. Output

$$b':=d\cdot(0\|y(0,r)\|\sigma(0,r)\oplus1\|y(1,r)\|\sigma(1,r)).\tag{123}$$

Theorem A.5. *If* $(\mathsf{Gen},\mathsf{Sign},\mathsf{Ver})$ *is a T-strong EUF-CMA secure digital signature scheme and* $\{\mathsf{PRF}_K:\{0,1\}^{u+1}\to\{0,1\}^v\}_{K\in\{0,1\}^\lambda}$ *is a quantum-query secure PRF, then the QPKE scheme* $(\mathsf{SKGen},\mathsf{PKGen},\mathsf{Enc},\mathsf{Dec})$ *above is correct and satisfies IND-pkT-CPA security.*

Proof (sketch). The correctness is easily seen similarly to the proof of Theorem 6.1.

For IND-pkT-CPA security, we only explain the differences from the proof of Theorem 6.1 since it is very similar. We define Hybrid 0, 1, and 2 similarly to those in the proof of Theorem 6.1. For clarity, we describe them in Figures 5 to 7.

Assume that the IND-pkT-CPA security of our construction is broken by a QPT adversary $\mathcal{A}$. It means the QPT adversary $\mathcal{A}$ wins Hybrid 0 with a non-negligible advantage. Then, it is clear that there is another QPT adversary $\mathcal{A}'$ that wins Hybrid 1 with a non-negligible advantage. ($\mathcal{A}'$ has only to do the Hadamard-basis measurement by itself.)

From the $\mathcal{A}'$, we construct a QPT adversary $\mathcal{A}''$ that wins Hybrid 2 with a non-negligible probability based on a similar proof to that in Section 7.1. Indeed, the proof is almost identical once we show that any QPT adversary given polynomially many copies of the public key can output a valid signature for a message that is not of the form $b\|r\|y(b,r)$ only with a negligible probability. To prove this, we consider a reduction algorithm that queries signatures on *all* messages of the form $b\|r\|y(b,r)$. Thus, the reduction algorithm makes 2^{u+1} queries and runs in time $2^u \cdot \mathrm{poly}(\lambda)$. Since we have $2^{u+1} < T$ and $2^u \cdot \mathrm{poly}(\lambda) < T$ for sufficiently large λ by $u = \lfloor (\log T)/2 \rfloor$, which in particular implies $2^u \leq T^{1/2}$, and $T = \lambda^{\omega(1)}$, the reduction enables us to prove the above property assuming the T-strong EUF-CMA security of the digital signature scheme.[14]

Thus, we are left to prove that no QPT adversary can win Hybrid 2 with a non-negligible probability. Let Hybrid 2' be a hybrid that works similarly to Hybrid 2 except that $y(b,r)$ is defined as $y(b,r) := H(b\|r)$ for a uniformly random function H instead of PRF. By the quantum-query security of PRF, the winning probabilities in Hybrid 2' and Hybrid 2 are negligibly close. Thus, it suffices to prove the winning probability in Hybrid 2' is negligible. This is proven by a straightforward reduction to Lemma A.4 noting that $|\psi_{\mathsf{sk}}\rangle$ with the modification of $y(b,r)$ as above can be generated from $|\psi_H\rangle = \sum_{R\in\{0,1\}^{u+1}} |R\rangle\, |H(R)\rangle$ by coherently running Sign. This completes the proof of IND-pkT-CPA security.

Hybrid 0

1. $\mathcal{C}$ runs $(k, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^\lambda)$. $\mathcal{C}$ sends vk to $\mathcal{A}$.
2. $\mathcal{C}$ sends $|\psi_{\mathsf{sk}}\rangle^{\otimes m}$ to the adversary $\mathcal{A}$, where

$$|\psi_{\mathsf{sk}}\rangle := \sum_{r\in\{0,1\}^u} |r\rangle \otimes \begin{pmatrix} |0\rangle \otimes |y(0,r)\rangle \otimes |\sigma(0,r)\rangle \\ + |1\rangle \otimes |y(1,r)\rangle \otimes |\sigma(1,r)\rangle \end{pmatrix} \tag{124}$$

3. $\mathcal{A}$ generates a quantum state over registers $(\mathbf{R}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$. ($(\mathbf{R}, \mathbf{A}, \mathbf{B}, \mathbf{C})$ corresponds to the quantum part of pk', and $\mathbf{D}$ corresponds to st.) $\mathcal{A}$ sends the registers $(\mathbf{R}, \mathbf{A}, \mathbf{B}, \mathbf{C})$ to $\mathcal{C}$. $\mathcal{A}$ keeps the register $\mathbf{D}$.
4. $\mathcal{C}$ coherently checks the signature in the state sent from $\mathcal{A}$. If the result is $\bot$, it sends $\bot$ to $\mathcal{A}$ and halts. If the result is $\top$, it goes to the next step.
5. $\mathcal{C}$ chooses $b \leftarrow \{0,1\}$. $\mathcal{C}$ applies Z^b on the register $\mathbf{A}$.
6. $\mathcal{C}$ measures $\mathbf{R}$ in the computational basis to get r.
7. $\mathcal{C}$ measures all qubits in $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ in the Hadamard basis to get the result d. $\mathcal{C}$ sends (r, d) to $\mathcal{A}$.
8. $\mathcal{A}$ outputs b'. If $b' = b$, $\mathcal{A}$ wins.

Figure 5: Hybrid 0

□

Remark A.6. We can add decryption error detectability by Theorem 5.3 and extend it to recyclable QPKE by

[14] In the proof for the mixed state public key version in Section 7.1, the reduction algorithm only needs to query signatures on $b\|r$ for r's used in one of the public keys given to the adversary. On the other hand, in the pure state public key case, each public key involves all r's and thus the reduction algorithm needs to query signatures on superpolynomially many messages. This is why we need superpolynomial security for the digital signature scheme.

Hybrid 1

1.-6. All the same as Figure 5.

7. $\mathcal{C}$ does not do the Hadamard-basis measurement, and $\mathcal{C}$ sends r and registers $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ to $\mathcal{A}$.

8. The same as Figure 5.

Figure 6: Hybrid 1

Hybrid 2

1.-7. All the same as Figure 6.

8. $\mathcal{A}$ outputs (μ_0, μ_1). If $\mu_0 = y(0, r) \| \sigma(0, r)$ and $\mu_1 = y(1, r) \| \sigma(1, r)$, $\mathcal{A}$ wins.

Figure 7: Hybrid 2

the construction of Section 10. These extensions preserve the property that public keys are pure states.

A.2 Construction from Polynomially Secure OWFs

We explain how to extend the construction in Appendix A.1 to base security on standard polynomial hardness of OWFs. We rely on a similar idea to the "on-the-fly-adaptation" technique introduced in [DS15]. The reason why we need superpolynomial security in Appendix A.1 is that the reduction algorithm for the transition from Hybrid 1 to 2 has to make $2^{u+1} \approx 2T^{1/2}$ signing queries for a superpolynomial T. Suppose that we set T to be a polynomial. i.e., $T = \lambda^c$ for some constant c. Then, the reduction algorithm for the transition from Hybrid 1 to 2 works under polynomial security of the digital signature scheme. The problem, however, is that we cannot show that the winning probability in Hybrid 2 is negligible: It can be only bounded by $(2m+1)^4(2^{-u} + 2^{-v})$, which is not negligible since $2^u \approx T^{1/2} = \lambda^{c/2}$. On the other hand, we can make it arbitrarily small inverse-polynomial by making c larger. Based on this observation, we can show the following: Let $(\mathsf{SKGen}_c, \mathsf{PKGen}_c, \mathsf{Enc}_c, \mathsf{Dec}_c)$ be the QPKE scheme given in Appendix A.1 where $T := \lambda^c$. Then, for any polynomials p and m, there exists a constant c such that any QPT adversary given m copies of the quantum public key has an advantage to break IND-pkT-CPA security of $(\mathsf{SKGen}_c, \mathsf{PKGen}_c, \mathsf{Enc}_c, \mathsf{Dec}_c)$ at most $1/p(\lambda)$ for all sufficiently large λ.

Then, our idea is to parallelly run $(\mathsf{SKGen}_c, \mathsf{PKGen}_c, \mathsf{Enc}_c, \mathsf{Dec}_c)$ for $c = 1, 2, ..., \lambda$ where the encryption algorithm generates a λ-out-of-λ secret sharing of the message and encrypts c-th share under Enc_c.[15] Suppose that this scheme is not IND-pkT-CPA secure. Then, there is a polynomial q and QPT adversary $\mathcal{A}$ given $m = \mathrm{poly}(\lambda)$ copies of the quantum public key that has an advantage to break the IND-pkT-CPA security at least $1/q(\lambda)$ for infinitely many λ. For each c, it is easy to construct a QPT adversary $\mathcal{A}_c$ that breaks IND-pkT-CPA security of $(\mathsf{SKGen}_c, \mathsf{PKGen}_c, \mathsf{Enc}_c, \mathsf{Dec}_c)$ with the same advantage as $\mathcal{A}$'s advantage. On the other hand, by the observation explained above, we can take a constant c (depending on q and m) such that any QPT adversary given m copies of the public key has an advantage to break IND-pkT-CPA security of $(\mathsf{SKGen}_c, \mathsf{PKGen}_c, \mathsf{Enc}_c, \mathsf{Dec}_c)$ at most $1/2q(\lambda)$ for all sufficiently large λ. This is a contradiction. Thus, the above scheme is IND-pkT-CPA secure.

[15] In fact, it suffices to parallelly run $(\mathsf{SKGen}_c, \mathsf{PKGen}_c, \mathsf{Enc}_c, \mathsf{Dec}_c)$ for $c = 1, 2, ..., \eta(\lambda)$ for any super-constant function η.

A.3 Proof of Lemma A.4

For proving Lemma A.4, we rely on the following lemma shown by [YZ21].

Lemma A.7 ([YZ21, Theorem 4.2]). *Let $H : \mathcal{X} \rightarrow \mathcal{Y}$ be a uniformly random function. Let $\mathcal{A}$ be an (unbounded-time) randomized algorithm that makes q quantum queries to H and outputs a classical string z. Let $\mathcal{C}$ be an (unbounded-time) randomized algorithm that takes z as input, makes k classical queries to H, and outputs $\top$ or $\bot$. Let $\mathcal{B}$ be the following algorithm that makes at most k classical queries to H:*

$\mathcal{B}^H()$**:** *It does the following:*

1. *Choose a function $H' : \mathcal{X} \rightarrow \mathcal{Y}$ from a family of $2q$-wise independent hash functions.*
2. *For each $j \in [k]$, uniformly pick $(i_j, b_j) \in ([q] \times \{0,1\}) \cup \{(\bot, \bot)\}$ under the constraint that there does not exist $j \neq j'$ such that $i_j = i_{j'} \neq \bot$.*
3. *Initialize a stateful oracle $\mathcal{O}$ to be a quantumly-accessible classical oracle that computes H'.*
4. *Run $\mathcal{A}^{\mathcal{O}}()$ where $\mathcal{O}$ is simulated as follows. When $\mathcal{A}$ makes its i-th query, the oracle is simulated as follows:*
   - *(a) If $i = i_j$ for some $j \in [k]$, measure $\mathcal{A}$'s query register to obtain x'_j, query x'_j to the random oracle H to obtain $H(x'_j)$, and do either of the following.*
     - *i. If $b_j = 0$, reprogram $\mathcal{O}$ to output $H(x'_j)$ on x'_j and answer $\mathcal{A}$'s i_j-th query by using the reprogrammed oracle.*
     - *ii. If $b_j = 1$, answer $\mathcal{A}$'s i_j-th query by using the oracle before the reprogramming and then reprogram $\mathcal{O}$ to output $H(x'_j)$ on x'_j.*
   - *(b) Otherwise, answer $\mathcal{A}$'s i-th query by just using the oracle $\mathcal{O}$ without any measurement or reprogramming.*
5. *Output whatever $\mathcal{A}$ outputs.*

Then we have

$$\Pr_H[\mathcal{C}^H(z) = \top : z \leftarrow \mathcal{B}^H()] \geq \frac{1}{(2q+1)^{2k}} \Pr_H[\mathcal{C}^H(z) = \top : z \leftarrow \mathcal{A}^H()]. \tag{125}$$

Remark A.8. There are the following differences from [YZ21, Theorem 4.2] in the statement of the lemma:

1. They consider inputs to $\mathcal{A}$ and $\mathcal{B}$. We omit them because this suffices for our purpose.

2. They consider a more general setting where $\mathcal{A}$ and $\mathcal{B}$ interact with $\mathcal{C}$. We focus on the non-interactive setting.

3. They do not explicitly write how $\mathcal{B}$ works in the statement of [YZ21, Theorem 4.2]. But this is stated at the beginning of its proof.

Using the above lemma, it is easy to prove Lemma A.4.

Proof of Lemma A.4. For an algorithm $\mathcal{A}$ in Lemma A.4, let $\tilde{\mathcal{A}}$ be an oracle-aided algorithm that generates m copies of $|\psi_H\rangle$ by making m quantum queries to H on uniform superpositions of inputs and then runs $\mathcal{A}(|\psi_H\rangle^{\otimes m})$. Let $\mathcal{C}$ be an oracle-aided algorithm that takes $z = (r, y_0, y_1)$ as input, makes two classical queries $0\|r$ and $1\|r$ to H, and outputs $\top$ if and only if $y_0 = H(0\|r)$ and $y_1 = H(1\|r)$. By Lemma A.7, we have

$$\Pr_H[y_0 = H(0\|r) \ \wedge \ y_1 = H(1\|r) : (r, y_0, y_1) \leftarrow \tilde{\mathcal{B}}^H()] \tag{126}$$

$$\geq \frac{1}{(2m+1)^4} \Pr_H[y_0 = H(0\|r) \ \wedge \ y_1 = H(1\|r) : (r, y_0, y_1) \leftarrow \tilde{\mathcal{A}}^H()] \tag{127}$$

where $\tilde{\mathcal{B}}$ is to $\tilde{\mathcal{A}}$ as $\mathcal{B}$ (defined in Lemma A.7) is to $\mathcal{A}$. By the definition of $\tilde{\mathcal{B}}$, it just makes at most two classical queries to H on independently and uniformly random inputs R_0, R_1. The probability that we happen to have $\{R_0, R_1\} = \{0\|r, 1\|r\}$ for some $r \in \{0,1\}^u$ is 2^{-u}. Unless the above occurs, either $H(0\|r)$ or $H(1\|r)$ is uniformly random to $\tilde{\mathcal{B}}$ for all r, and thus the probability that its output satisfies $y_0 = H(0\|r)$ and $y_1 = H(1\|r)$ is at most 2^{-v}. Thus, we have

$$\Pr_H[y_0 = H(0\|r) \ \wedge \ y_1 = H(1\|r) : (r, y_0, y_1) \leftarrow \tilde{\mathcal{B}}^H()] \leq 2^{-u} + 2^{-v}. \tag{128}$$

Combining Equations (127) and (128), we obtain Lemma A.4. □