Quantum Encryption with Certified Deletion, Revisited: Public Key, Attribute-Based, and Classical Communication

Taiga Hiroka[1], Tomoyuki Morimae[1,2], Ryo Nishimaki[3], and Takashi Yamakawa[3]

[1] Yukawa Institute for Theoretical Physics, Kyoto University, Kyoto, Japan
[2] PRESTO, JST, Saitama, Japan
[3] NTT Corporation, Tokyo, Japan

Abstract. Broadbent and Islam (TCC '20) proposed a quantum cryptographic primitive called *quantum encryption with certified deletion*. In this primitive, a receiver in possession of a quantum ciphertext can generate a classical certificate that the encrypted message has been deleted. Although their construction is information-theoretically secure, it is limited to the setting of one-time symmetric key encryption (SKE), where a sender and receiver have to share a common key in advance and the key can be used only once. Moreover, the sender has to generate a quantum state and send it to the receiver over a quantum channel in their construction. Deletion certificates are privately verifiable, which means a verification key for a certificate must be kept secret, in the definition by Broadbent and Islam. However, we can also consider public verifiability.
In this work, we present various constructions of encryption with certified deletion.

- Quantum communication case: We achieve (reusable-key) public key encryption (PKE) and attribute-based encryption (ABE) with certified deletion. Our PKE scheme with certified deletion is constructed assuming the existence of IND-CPA secure PKE, and our ABE scheme with certified deletion is constructed assuming the existence of indistinguishability obfuscation and one-way functions. These two schemes are privately verifiable.
- Classical communication case: We also achieve interactive encryption with certified deletion that uses only classical communication. We give two schemes, a privately verifiable one and a publicly verifiable one. The former is constructed assuming the LWE assumption in the quantum random oracle model. The latter is constructed assuming the existence of one-shot signatures and extractable witness encryption.

1 Introduction

The no-cloning theorem, which states that an unknown quantum state cannot be copied in general, is one of the most fundamental principles in quantum physics. As any classical information can be trivially copied, this indicates a fundamental

difference between classical and quantum information. The no-cloning theorem has been the basis of many quantum cryptographic protocols, including quantum money [Wie83] and quantum key distribution [BB84].

Broadbent and Islam [BI20] used the principle to construct *quantum encryption with certified deletion*. In this primitive, a sender encrypts a classical message to generate a quantum ciphertext. A receiver in possession of the quantum ciphertext and a classical decryption key can either decrypt the ciphertext or "delete" the encrypted message by generating a classical certificate. After generating a valid certificate of deletion, no adversary can recover the message *even if the decryption key is given*.[4] We remark that this functionality is classically impossible to achieve since one can copy a classical ciphertext and keep it so that s/he can decrypt it at any later time. They prove the security of their construction without relying on any computational assumption, which ensures information-theoretical security. Although they achieved the exciting new functionality, their construction is limited to the one-time symmetric key encryption (SKE) setting. A sender and receiver have to share a common key in advance in one-time SKE, and the key can be used only once.

A possible application scenario of quantum encryption with certified deletion is the following. A user uploads encrypted data on a quantum cloud server. Whenever the user wishes to delete the data, the cloud generates a deletion certificate and sends it to the user. After the user verifies the validity of the certificate, s/he is convinced that the data cannot be recovered even if the decryption key accidentally leaks later. Such quantum encryption could prevent data retention and help to implement the right to be forgotten [GDP16]. In this scenario, one-time SKE is quite inconvenient. By the one-time restriction, the user has to locally keep as many decryption keys as the number of encrypted data in the cloud, in which case there seems to be no advantage of uploading the data to the cloud server: If the user has such large storage, s/he could have just locally kept the messages rather than uploading encryption of them to the cloud. Also, in some cases, a party other than the decryptor may want to upload data to the cloud. This usage would be possible if we can extend the quantum encryption with certified deletion to public key encryption (PKE). We remark that the one-time restriction is automatically resolved for PKE by a simple hybrid argument. Even more flexibly, a single encrypted data on the cloud may be supposed to be decrypted by multiple users according to some access control policy. Attribute-based encryption (ABE) [SW05, GPSW06] realizes such an access control in classical cryptography. Thus, it would be useful if we have ABE with certified deletion. Our first question in this work is:

Can we achieve PKE and ABE with certified deletion?

Moreover, a sender needs to send quantum states (random BB84 states [BB84]) over a quantum channel in the construction by Broadbent and Islam [BI20]. Al-

[4] We note that if the adversary is given the decryption key before the deletion, it can decrypt the ciphertext to obtain the message and keep it even after the deletion, but such an "attack" is unavoidable.

though generating and sending random BB84 states are not difficult tasks (and they are already possible with current technologies), a classical sender and communication over only a classical channel are much easier. Besides, communicating over a classical channel is desirable in the application scenario above since many parties want to upload data to a cloud. In addition to these practical motivations, achieving classical channel certified deletion is also an interesting theoretical research direction given the fact that many quantum cryptographic protocols have been "dequantized" recently [Mah18, CCKW19, RS19, AGKZ20, KNY20]. Thus, our second question in this work is:

Can we achieve encryption with certified deletion, a classical sender, and classical communication?

In the definition by Broadbent and Islam [BI20], a verification key for a deletion certificate must be kept secret (privately verifiable). If the verification key is revealed, the security is no longer guaranteed in their scheme. We can also consider public verifiability, which means the security holds even if a verification key is revealed to adversaries. Broadbent and Islam left the following question as an open problem:

Is publicly verifiable encryption with certified deletion possible?

1.1 Our Result

We solve the three questions above affirmatively in this work.

PKE and ABE with certified deletion and quantum communication. We present formal definitions of PKE and ABE with certified deletion, and present constructions of them:

- We construct a PKE scheme with certified deletion assuming the existence of (classical) IND-CPA secure PKE. We also observe that essentially the same construction gives a reusable SKE scheme with certified deletion if we use IND-CPA secure SKE, which exists under the existence of one-way function (OWF), instead of PKE.
- We construct a (public-key) ABE scheme with certified deletion assuming the existence of indistinguishability obfuscation (iO) [BGI+12] and OWF. This construction satisfies collusion resistance and adaptive security, i.e., it is secure against adversaries that adaptively select a target attribute and obtain arbitrarily many decryption keys.

All builiding blocks above are post-quantum secure in this work. We note that our constructions rely on computational assumptions and thus are not information-theoretically secure, unlike the construction in [BI20]. This is unavoidable since even plain PKE or ABE cannot be information-theoretically secure. We also note that the constructions above are privately verifiable as the definition of one-time SKE by Broadbent and Islam [BI20].

Our main technical insight is that we can combine the one-time secure SKE with certified deletion of [BI20] and plain PKE to construct PKE with certified deletion by a simple hybrid encryption technique if the latter satisfies *receiver non-committing* (RNC) security [CFGN96, JL00, CHK05]. Since it is known that PKE/SKE with RNC security can be constructed from any IND-CPA secure PKE/SKE [CHK05, KNTY19], our first result follows.

For the second result, we first give a suitable definition of RNC security for ABE that suffices for our purpose. Then we construct an ABE scheme with RNC security based on the existence of iO and OWF. By combining this with one-time SKE with certified deletion by hybrid encryption, we obtain an ABE scheme with certified deletion.

Interactive encryption with certified deletion, a classical sender, and classical communication. We also present formal definitions of PKE with certified deletion and classical communication, and present two constructions:

- We construct an interactive encryption scheme with privately verifiable certified deletion and classical communication in the quantum random oracle model (QROM) [BDF+11]. Our construction is secure under the LWE assumption in the QROM.
- We construct an interactive encryption scheme with publicly verifiable certified deletion and classical communication. Our construction uses one-shot signatures [AGKZ20] and extractable witness encryption [GGSW13, GKP+13]. This solves the open problem by Broadbent and Islam [BI20].

A sender is a classical algorithm in both constructions but needs to interact with a receiver during ciphertext generation.

An encryption algorithm must be interactive in the classical communication case even if we consider computationally bounded adversaries (and even in the QROM). The reason is that a malicious QPT receiver can generate two copies of a quantum ciphertext from classical messages sent from a sender. One is used for generating a deletion certificate, and the other is used for decryption.

Moreover, both constructions rely on computational assumptions and thus are not information-theoretically secure, unlike the construction by Broadbent and Islam [BI20]. This is unavoidable even if an encryption algorithm is interactive (and even in the QROM). The reason is that a computationally unbounded malicious receiver can classically simulate its honest behavior to get a classical description of the quantum ciphertext.

For the first construction, we use a new property of noisy trapdoor claw-free (NTCF) functions, *the cut-and-choose adaptive hardcore property* (Lemma 4.1), which we introduce in this work. We prove that the cut-and-choose adaptive hardcore property is reduced to the adaptive hardcore bit property [BCM+18] and injective invariance [Mah18]. Those properties hold under the LWE assumption [BCM+18, Mah18]. This new technique is of independent interest. The idea of the second construction is to encrypt a plaintext by witness encryption so that a valid witness is a one-shot signature for bit 0. We use a valid one-shot signature for bit 1 as a deletion certificate. The one-shot property of one-shot signatures

prevents decryption of witness encryption after issuing a valid deletion certificate. Georgiou and Zhandry [GZ20] used a similar combination of one-shot signatures and witness encryption to construct unclonable decryption keys.

1.2 Related work

Before the work by Broadbent and Islam [BI20], Fu and Miller [FM18] and Coiteux-Roy and Wolf [CRW19] also studied the concept of certifying deletion of information in different settings. (See [BI20] for the comparison with these works.)

The quantum encryption scheme with certified deletion by Broadbent and Islam [BI20] is based on Wiesner's conjugate coding, which is the backbone of quantum money [Wie83] and quantum key distribution [BB84]. A similar idea has been used in many constructions in quantum cryptography that include (but are not limited to) revocable quantum timed-release encryption [Unr15], uncloneable quantum encryption [BL20], single-decryptor encryption [GZ20], and copy protection/secure software leasing [CMP20]. Among them, revocable quantum timed-release encryption is conceptually similar to quantum encryption with certified deletion. In this primitive, a receiver can decrypt a quantum ciphertext only after spending a certain amount of time T. The receiver can also choose to return the ciphertext before the time T is over, in which case it is ensured that the message can no longer be recovered. As observed by Broadbent and Islam [BI20], an essential difference from quantum encryption with certified deletion is that the revocable quantum timed-release encryption does not have a mechanism to generate a *classical* certificate of deletion. Moreover, the construction by Unruh [Unr15] heavily relies on the random oracle heuristic [BR97, BDF+11], and there is no known construction without random oracles.

Kundu and Tan [KT20] constructed (one-time symmetric key) quantum encryption with certified deletion with the device-independent security, i.e., the security holds even if quantum devices are untrusted. Moreover, they show that their construction satisfies composable security.

The notion of NTCF functions was first introduced by Brakerski et al. [BCM+18], and further extended to construct a classical verification of quantum computing by Mahadev [Mah18]. (See also a related primitive so-called QFactory [CCKW19].) The adaptive hardcore bit property of NTCF functions was also used for semi-quantum money [RS19] and secure software leasing with classical communication [KNY20].

Ananth and Kaleoglu concurrently and independently present reusable secret key and public key uncloneable encryption schemes [AK21]. Uncloneable encryption [BL20] is related to but different from quantum encryption with certified deletion. Uncloneable encryption prevents adversaries from creating multiple ciphertexts whose plaintext is the same as that of the original ciphertext. Their constructions are based on a similar idea to one of our main ideas. Specifically, their construction is obtained by combining one-time secret key uncloneable encryption and standard SKE/PKE with the "fake-key property", which is similar to the RNC security.

1.3 Technical Overview Part I: Quantum Communication Case

We provide an overview of how to achieve PKE and ABE with certified deletion using quantum communication in this section. To explain our idea, we introduce the definition of PKE with certified deletion.

Definition of quantum encryption with certified deletion. A PKE with certified deletion consists of the following algorithms.

$\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$**:** This is a key generation algorithm that generates a pair of public and secret keys.

$\mathsf{Enc}(\mathsf{pk}, m) \to (\mathsf{vk}, \mathsf{CT})$**:** This is an encryption algorithm that generates a ciphertext of plaintext and a verification key for this ciphertext.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{CT}) \to m'$**:** This is a decryption algorithm that decrypts a ciphertext.

$\mathsf{Del}(\mathsf{CT}) \to \mathsf{cert}$**:** This is a deletion algorithm that generates a certificate to guarantee that the ciphertext CT was deleted.

$\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert}) \to \top$ **or** $\bot$**:** This is a verification algorithm that checks the validity of a certificate cert by using a verification key. As correctness, we require that this algorithm returns $\top$ (i.e., it accepts) if cert was honestly generated by $\mathsf{Del}(\mathsf{CT})$ and $(\mathsf{vk}, \mathsf{CT})$ was honestly generated by Enc.

Roughly speaking, certified deletion security requires that no quantum polynomial time (QPT) adversary given pk and CT can obtain any information about the plaintext in CT *even if* sk *is given after a valid certificate* $\mathsf{cert} \leftarrow \mathsf{Del}(\mathsf{CT})$ *is generated*. The difference between PKE and reusable SKE with certified deletion is that, in reusable SKE, KeyGen outputs only sk. In the one-time SKE case by Broadbent and Islam [BI20], Enc does not output vk and Vrfy uses sk instead of vk.

Our idea for PKE. We use the construction of one-time SKE with certified deletion by Broadbent and Islam [BI20]. However, we do not need to know the detail of the SKE scheme since we use it in a black-box way in our PKE scheme. What we need to understand about the SKE scheme are the following abstracted properties: (1) A secret key and a plaintext are classical strings. (2) A ciphertext is a quantum state. (3) The encryption algorithm does not output a verification key since the verification key is equal to the secret key. (4) It satisfies the verification correctness and certified deletion security explained above.

Our idea is to convert the SKE with certified deletion scheme into a PKE with certified deletion scheme by combining with a standard PKE scheme (standard hybrid encryption technique). This conversion is possible since a secret key of the SKE scheme is a classical string. Let $\mathsf{PKE}.(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ and $\mathsf{SKE}.(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ be normal PKE and one-time SKE with certified deletion schemes, respectively. Our PKE with certified deletion scheme is described as follows.

$\mathsf{KeyGen}(1^\lambda)$**:** This outputs $(\mathsf{pke.pk}, \mathsf{pke.sk}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$.

$\mathsf{Enc}(\mathsf{pk}, m)$: This generates $\mathsf{ske.sk} \leftarrow \mathsf{SKE.KeyGen}(1^\lambda)$, $\mathsf{ske.CT} \leftarrow \mathsf{SKE.Enc}(\mathsf{ske.sk}, m)$, and $\mathsf{pke.CT} \leftarrow \mathsf{PKE.Enc}(\mathsf{pke.pk}, \mathsf{ske.sk})$, and outputs $\mathsf{vk} := \mathsf{ske.sk}$ and $\mathsf{CT} := (\mathsf{ske.CT}, \mathsf{pke.CT})$.
$\mathsf{Dec}(\mathsf{sk}, \mathsf{CT})$: This computes $\mathsf{ske.sk}' \leftarrow \mathsf{PKE.Dec}(\mathsf{pke.sk}, \mathsf{pke.CT})$ and $m' \leftarrow \mathsf{SKE.Dec}(\mathsf{ske.sk}', \mathsf{ske.CT})$, and outputs m'.
$\mathsf{Del}(\mathsf{CT})$: This generates and outputs $\mathsf{cert} \leftarrow \mathsf{SKE.Del}(\mathsf{ske.CT})$.
$\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert})$: This outputs the output of $\mathsf{SKE.Vrfy}(\mathsf{ske.sk}, \mathsf{cert})$ (note that $\mathsf{vk} = \mathsf{ske.sk}$).

At first glance, this naive idea seems to work since even if $\mathsf{pke.sk}$ is given to an adversary after a valid cert is generated, $\mathsf{ske.CT}$ does not leak information about the plaintext by certified deletion security of the SKE scheme. Note that PKE is used to encrypt $\mathsf{ske.sk}$ (not m). One-time SKE is sufficient since $\mathsf{ske.sk}$ is freshly generated in Enc. The proof outline is as follows. First, we use IND-CPA security of normal PKE to erase information about $\mathsf{ske.sk}$. Then, we use the one-time certified deletion security of SKE. Unfortunately, we do not know how to prove the first step above because we must give $\mathsf{pke.sk}$ to an adversary in a security reduction. In the first step, we need to show that if a distinguisher detects that $\mathsf{PKE.Enc}(\mathsf{pke.pk}, \mathsf{ske.sk})$ is changed to $\mathsf{PKE.Enc}(\mathsf{pke.pk}, 0^{|\mathsf{ske.sk}|})$, we can break IND-CPA security of the normal PKE. However, to run the distinguisher, we need to give $\mathsf{pke.sk}$ to the distinguisher after it sends a valid certificate for deletion. The reduction has no way to give $\mathsf{pke.sk}$ to the distinguisher since the reduction is trying to break the PKE scheme!

To solve this problem, we use RNC encryption (RNCE) [JL00, CHK05]. RNCE consists of algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Reveal})$. The key generation algorithm outputs not only a key pair $(\mathsf{pk}, \mathsf{sk})$ but also an auxiliary trapdoor information aux. The fake ciphertext generation algorithm $\mathsf{Fake}(\mathsf{pk}, \mathsf{sk}, \mathsf{aux})$ can generate a fake ciphertext $\widetilde{\mathsf{CT}}$ that does not include information about a plaintext. The reveal algorithm $\mathsf{Reveal}(\mathsf{pk}, \mathsf{sk}, \mathsf{aux}, \widetilde{\mathsf{CT}}, m)$ can generate a fake secret key that decrypts $\widetilde{\mathsf{CT}}$ to m. The RNC security notion requires that $(\widetilde{\mathsf{CT}} = \mathsf{Fake}(\mathsf{pk}, \mathsf{sk}, \mathsf{aux}), \mathsf{Reveal}(\mathsf{pk}, \mathsf{sk}, \mathsf{aux}, \widetilde{\mathsf{CT}}, m))$ is computationally indistinguishable from $(\mathsf{Enc}(\mathsf{pk}, m), \mathsf{sk})$.

RNCE perfectly fits the scenario of certified deletion. We use an RNCE scheme $\mathsf{RNCE}.(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Reveal})$ instead of a normal PKE in the PKE with certified deletion scheme above. To erase $\mathsf{ske.sk}$, we use the RNC security. We change $\mathsf{RNCE.Enc}(\mathsf{rnce.pk}, \mathsf{ske.sk})$ and $\mathsf{rnce.sk}$ into $\mathsf{rnce.}\widetilde{\mathsf{CT}} = \mathsf{RNCE.Fake}(\mathsf{rnce.pk}, \mathsf{rnce.sk}, \mathsf{rnce.aux})$ and $\mathsf{RNCE.Reveal}(\mathsf{rnce.pk}, \mathsf{rnce.sk}, \mathsf{rnce.aux}, \mathsf{rnce.}\widetilde{\mathsf{CT}}, \mathsf{ske.sk})$, respectively. Thus, as long as $\mathsf{ske.sk}$ is given after a valid certification is generated, we can simulate the secret key of the PKE with certified deletion scheme. Using RNCE solves the problem above since the reduction obtains both a target ciphertext and a secret key (real or fake) in the RNC security game. To complete the security proof, we use the certified deletion security of SKE. Here, the point is that the reduction can simulate a secret key by Reveal since the reduction is given $\mathsf{ske.sk}$ after a valid certificate is sent in the certified deletion security game.

If we use secret key RNCE instead of public key RNCE, we can achieve reusable SKE with certified deletion via the design idea above. Secret/public key RNCE

can be constructed from IND-CPA SKE/PKE, respectively [CHK05, KNTY19], and SKE with certified deletion exists unconditionally [BI20]. Thus, we can achieve PKE (resp. reusable SKE) with certified deletion from IND-CPA PKE (resp. OWFs).

Note that the RNCE technique above is the fundamental technique in this work. We use this technique both in the quantum communication case and in the classical communication case.

Our idea for ABE. We can extend the idea for PKE to the ABE setting. In this work, we focus on key-policy ABE, where a policy (resp. attribute) is embedded in a secret key (resp. ciphertext). The crucial tool is (receiver) non-committing ABE (NCABE), which we introduce in this work.

Although the definition of NCABE is basically a natural extension of that of RNCE, we describe algorithms of NCABE for clarity. It helps readers who are not familiar with normal ABE. The first four algorithms below are algorithms of normal ABE.

$\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{msk})$**:** This is a setup algorithm that generates a public key and a master secret key.

$\mathsf{KeyGen}(\mathsf{msk}, P) \to \mathsf{sk}_P$**:** This is a key generation algorithm that generates a secret key for a policy P.

$\mathsf{Enc}(\mathsf{pk}, X, m) \to \mathsf{CT}_X$**:** This is an encryption algorithm that generates a ciphertext of m under an attribute X.

$\mathsf{Dec}(\mathsf{sk}_P, \mathsf{CT}_X) \to m'$ **or** $\bot$**:** This is a decryption algorithm that decrypts CT_X if $P(X) = \top$. If $P(X) = \bot$, it outputs $\bot$.

$\mathsf{FakeSetup}(1^\lambda) \to (\mathsf{pk}, \mathsf{aux})$**:** This is a fake setup algorithm that generates a public key and a trapdoor auxiliary information aux.

$\mathsf{FakeCT}(\mathsf{pk}, \mathsf{aux}, X) \to \widetilde{\mathsf{CT}}_X$**:** This is a fake ciphertext generation algorithm that generates a fake ciphertext $\widetilde{\mathsf{CT}}_X$ under an attribute X.

$\mathsf{FakeSK}(\mathsf{pk}, \mathsf{aux}, P) \to \widetilde{\mathsf{sk}}_P$**:** This is a fake key generation algorithm that generates a fake secret key $\widetilde{\mathsf{sk}}_P$ for P.

$\mathsf{Reveal}(\mathsf{pk}, \mathsf{aux}, \widetilde{\mathsf{CT}}, m) \to \widetilde{\mathsf{msk}}$**:** This is a reveal algorithm that generates a fake master secret key $\widetilde{\mathsf{msk}}$.

Roughly speaking, the NCABE security notion requires that the fake public key, master secret key, ciphertext, and secret keys are computationally indistinguishable from the normal public key, master key, ciphertext, and secret keys. It is easy to see that the hybrid encryption approach works in the ABE setting as well. Thus, the goal is to achieve an NCABE scheme.

Our NCABE construction follows the RNCE construction based on IND-CPA PKE [CHK05, KNTY19]. However, the crucial difference between the PKE and ABE settings is that, in the ABE setting, adversaries are given many secret keys for queried policies (that is, we consider collusion-resistance). There is an obstacle to achieving collusion resistance because secret keys for policies depend on a master secret key. Note that adversaries can send secret key queries *both before and after* the target ciphertext is given.

First, we explain the RNCE scheme from PKE. Although we explain the 1-bit plaintext case, it is easy to extend to the multi-bit case. The idea is the simple double encryption technique by Naor and Yung [NY90], but we do not need non-interactive zero-knowledge (NIZK). We generate two key pairs $(\mathsf{pk}_0, \mathsf{sk}_0)$ and $(\mathsf{pk}_1, \mathsf{sk}_1)$ and set $\mathsf{pk} := (\mathsf{pk}_0, \mathsf{pk}_1)$, $\mathsf{sk} := \mathsf{sk}_z$, and $\mathsf{aux} = (\mathsf{sk}_0, \mathsf{sk}_1, z^*)$ where $z, z^* \leftarrow \{0,1\}$. A ciphertext consists of $\mathsf{Enc}(\mathsf{pk}_0, b)$ and $\mathsf{Enc}(\mathsf{pk}_1, b)$. We can decrypt the ciphertext by using sk_z. A fake ciphertext $\widetilde{\mathsf{CT}}$ is $(\mathsf{Enc}(\mathsf{pk}_{z^*}, 0), \mathsf{Enc}(\mathsf{pk}_{1-z^*}, 1))$. To generate a fake secret key for a plaintext m^*, the reveal algorithm outputs $\mathsf{sk}_{z^* \oplus m^*}$. It is easy to see that decrypting $\widetilde{\mathsf{CT}}$ with $\mathsf{sk}_{z^* \oplus m^*}$ yields m^*.

Our NCABE is based on the idea above. That is, we use two key pairs $(\mathsf{pk}_0, \mathsf{msk}_0)$ and $(\mathsf{pk}_1, \mathsf{msk}_1)$ of a normal ABE scheme $\mathsf{ABE}.(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, and a ciphertext consists of $(\mathsf{ABE.Enc}(\mathsf{pk}_0, X, b), \mathsf{ABE.Enc}(\mathsf{pk}_1, X, b))$ where X is an attribute. Our reveal algorithm outputs $\mathsf{msk}_{z^* \oplus m^*}$ for a plaintext m^* as in the PKE case. The problem is a secret key for a policy P. A naive idea is that a key generation algorithm outputs $\mathsf{sk}_P \leftarrow \mathsf{ABE.KeyGen}(\mathsf{msk}_z, P)$ where $z \leftarrow \{0,1\}$ is chosen in the setup algorithm, and a fake key generation algorithm outputs $\widetilde{\mathsf{sk}}_P \leftarrow \mathsf{ABE.KeyGen}(\mathsf{msk}_{z^* \oplus m^*}, P)$. However, this apparently does not work since $\widetilde{\mathsf{sk}}_P$ depends on m^*. Unless $\widetilde{\mathsf{sk}}_P$ is independent of m^*, we cannot use NCABE to achieve ABE with certified deletion because $\mathsf{ske.sk}$ of SKE with certified deletion is sent *after* a valid certification is generated ($\mathsf{ske.sk}$ would be a plaintext of ABE in the hybrid encryption). To make a fake key generation be independent of m^*, we need to hide which master secret key is used to generate a secret key for P. If a secret key leaks information about which secret key (extracted from msk_0 or msk_1) is used, we cannot adaptively select a fake master secret key in the reveal algorithm.

iO helps us to overcome this hurdle. Our idea is as follows. A key generation algorithm outputs an obfuscated circuit of a circuit $\mathsf{D}[\mathsf{sk}_z]$ that takes a ciphertext $(\mathsf{abe.CT}_0, \mathsf{abe.CT}_1) := (\mathsf{ABE.Enc}(\mathsf{pk}_0, X, b), \mathsf{ABE.Enc}(\mathsf{pk}_1, X, b))$ and outputs $\mathsf{ABE.Dec}(\mathsf{sk}_z, \mathsf{abe.CT}_z)$ where $z \leftarrow \{0,1\}$ and $\mathsf{sk}_z \leftarrow \mathsf{ABE.KeyGen}(\mathsf{msk}_z, P)$ is hard-coded in D. A fake key generation algorithm outputs an obfuscated circuit of a circuit $\mathsf{D}_0[\mathsf{sk}_0]$ that takes $(\mathsf{abe.CT}_0, \mathsf{abe.CT}_1)$ and outputs $\mathsf{ABE.Dec}(\mathsf{sk}_0, \mathsf{abe.CT}_0)$ where $\mathsf{sk}_0 \leftarrow \mathsf{ABE.KeyGen}(\mathsf{msk}_0, P)$ is hard-coded in D_0. Note that the fake secret key cannot be used to decrypt a fake ciphertext $(\mathsf{abe.CT}_{z^*}, \mathsf{abe.CT}_{1-z^*}) := (\mathsf{ABE.Enc}(\mathsf{pk}_{z^*}, X, 0), \mathsf{ABE.Enc}(\mathsf{pk}_{1-z^*}, X, 1))$ where $z^* \leftarrow \{0,1\}$ since $P(X) = \bot$ must hold by the requirement on ABE security. Since the decryption circuits D and D_0 are obfuscated, adversaries have no idea about which secret key (sk_0 or sk_1) is used for decryption. This idea is inspired by the functional encryption (FE) scheme by Garg et al. [GGH+16].

The final issue is that adversaries can detect whether a secret key is real or fake if they use an invalid ciphertext $(\mathsf{ABE.Enc}(\mathsf{pk}_0, b), \mathsf{ABE.Enc}(\mathsf{pk}_1, 1-b))$ as an input to the obfuscated circuits. To prevent this attack, we use statistically sound NIZK to check the consistency of double encryption as the FE scheme by Garg et al. [GGH+16]. By the statistical soundness of NIZK, we can guarantee that the obfuscated decryption circuit does not accept invalid ciphertexts, and D and

D_0 are functionally equivalent. Note that a secret key for policy P outputs $\perp$ for the target ciphertext since a target attribute X^* in the target ciphertext satisfies $P(X) = \perp$. We do not need the simulation-soundness, unlike the FE scheme by Garg et al. due to the following reason. In the FE scheme, plain PKE schemes are used for the double encryption technique and a secret key sk_0 or sk_1 is hard-coded in a functional decryption key. Before we use PKE security under pk_b, we need to switch the decryption key from sk_b to sk_{1-b} by iO security. During this phase, we need to use a fake simulated proof of NIZK. Thus, the simulation-soundness is required. However, in our ABE setting, a secret key for P (not the master secret keys $\mathsf{msk}_0, \mathsf{msk}_1$) is hard-coded in D (or D_0) above. Thanks to the ABE key oracle, sk_0 and sk_1 for P are always available in reductions. We can first use iO security to switch from D to D_0. After that, we change a real NIZK proof into a fake one. Thus, our NCABE scheme does not need the simulation-soundness. This observation enables us to achieve the adaptive security rather than the selective security, unlike the FE scheme by Garg et al.[5] Thus, we can achieve NCABE from iO and OWFs since adaptively secure standard ABE can be constructed from iO and OWFs.

1.4 Technical Overview Part II: Classical Communication Case

We provide an overview of how to achieve privately verifiable and publicly verifiable interactive encryption with certified deletion using classical communication in this section. We note that both of them rely on interactive encryption algorithms.

Privately verifiable construction. For realizing a privately verifiable construction with classical communication, we rely on *NTCF functions* [BCM+18, Mah18]. In this overview, we consider an ideal version, noise-free claw-free permutations for simplicity. A trapdoor claw-free permutation is $f : \{0,1\} \times \{0,1\}^w \to \{0,1\}^w$ such that (1) $f(0,\cdot)$ and $f(1,\cdot)$ are permutations over $\{0,1\}^w$, (2) given the description of f, it is hard to find x_0 and x_1 such that $f(0,x_0) = f(1,x_1)$, and (3) there is a trapdoor td that enables one to efficiently find x_0 and x_1 such that $f(0,x_0) = f(1,x_1) = y$ for any y. In addition, the existing work showed that (a noisy version of) it satisfies a property called the *adaptive hardcore bit property* under the LWE assumption [BCM+18]. To explain this, suppose that one generates the state $\sum_{b,x} |b\rangle |x\rangle |f(b,x)\rangle$, and measures the third register in the computational basis to get a result y. Then the first and second registers collapse to the state $\frac{1}{\sqrt{2}}(|0\rangle |x_0\rangle + |1\rangle |x_1\rangle)$ with $f(0,x_0) = f(1,x_1) = y$. If one measures the state in the computational basis, the measurement outcome is $(0,x_0)$ or $(1,x_1)$. If, on the other hand, one measures the state in the Hadamard basis, the measurement outcome is (e,d) such that $e = d \cdot (x_0 \oplus x_1)$. The adaptive hardcore bit property roughly means that once one gets $(0,x_0)$ or $(1,x_1)$, it cannot output (e,d) such that $d \neq 0$ and $e = d \cdot (x_0 \oplus x_1)$ with probability better

[5] In the initial version of this work [NY21], we achieve only the selective security because we use statistical simulation-sound NIZK as the FE scheme by Garg et al. [GGH+16]. We improve the result.

than $1/2 + \mathsf{negl}(\lambda)$. Note that this is a tight bound since $e = d \cdot (x_0 \oplus x_1)$ holds with probability $1/2$ if we randomly choose e. Existing works showed that this property can be amplified by parallel repetition [RS19, KNY20]: Specifically, let $(0, x_{i,0})$ and $(1, x_{i,1})$ be the preimages of y_i under f_i for $i \in [n]$ where $n = \omega(\log \lambda)$. Then once one gets a sequence $\{b_i, x_{i,b_i}\}_{i\in[n]}$ for some $b_1\|...\|b_n \in \{0,1\}^n$, it can get a sequence $\{e_i, d_i\}_{i\in[n]}$ such that $d_i \neq 0$ and $e_i = d_i \cdot (x_{i,0} \oplus x_{i,1})$ only with negligible probability.

We use this property to construct an encryption scheme with certified deletion. A natural idea would be as follows: The sender sends n functions $\{f_i\}_{i\in[n]}$ to the receiver, the receiver generates $\{y_i\}_{i\in[n]}$ along with states $\{\frac{1}{\sqrt{2}}(|0\rangle |x_{i,0}\rangle + |1\rangle |x_{i,1}\rangle)\}_{i\in[n]}$ as above and sends $\{y_i\}_{i\in[n]}$ to the sender, and the sender sends receiver a ciphertext CT decryptable only when $\{b_i, x_{i,b_i}\}_{i\in[n]}$ for some $b_1\|...\|b_n \in \{0,1\}^n$ is available. We discuss how to implement such a ciphertext later. We use $\{e_i, d_i\}_{i\in[n]}$ such that $e_i = d_i \cdot (x_{i,0} \oplus x_{i,1})$ as a deletion certificate. The receiver can decrypt the ciphertext by measuring the states in the computational basis, and once it outputs a valid deletion certificate, it must "forget" preimages by the amplified adaptive hardcore property and thus cannot decrypt the ciphertext. This idea can be implemented in a straightforward manner if we generate CT by (extractable) witness encryption [GGSW13, GKP+13] under the corresponding **NP** language. However, since witness encryption is a strong assumption, we want to avoid this. Indeed, we can find the following candidate construction using a hash function H modeled as a random oracle. We set the ciphertext as $\mathsf{CT} := \{\mathsf{CT}_{i,b}\}_{i\in[n],b\in\{0,1\}}$ where $\{m_i\}_{i\in[n]}$ is an n-out-of-n secret sharing of the message m and $\mathsf{CT}_{i,b} := m_i \oplus H(b\|x_{i,b})$. The intuition is that an adversary has to get m_i for all $i \in [n]$ to get m and it has to know $(0, x_{i,0})$ or $(1, x_{i,1})$ to know m_i. Therefore, it seems that any adversary that gets any information of m can be used to extract a sequence $\{b_i, x_{i,b_i}\}_{i\in[n]}$ for some $b_1\|...\|b_n \in \{0,1\}^n$. If this is shown, it is straightforward to prove that the adversary can get no information of m once it submits a valid deletion certificate by the amplified adaptive hardcore property as explained above. However, turning this intuition into a formal proof seems difficult. A common technique to extract information from adversary's random oracle queries is the one-way to hiding lemma [Unr15, AHU19]. The lemma roughly claims that if the adversary distinguishes $H(X)$ from random, then we would get X with non-negligible probability by measuring a randomly chosen query. Here, a problem is that we have to extract n strings $\{b_i, x_{i,b_i}\}_{i\in[n]}$ simultaneously. On the other hand, the extraction by the one-way to hiding lemma disturbs adversary's state by a measurement, and thus we cannot use this technique sequentially.[6]

The difficulty above comes from the fact that the sender cannot know which of $(0, x_{i,0})$ and $(1, x_{i,1})$ the receiver will get, and thus it has to send a ciphertext that can be decrypted in either case. To resolve this issue, we rely on the injective invariance, which roughly says that there is an injective function g that

[6] A recent work by Coladangelo, Majenz, and Poremba [CMP20] studied what is called "simultaneous one-way to hiding lemma", but their setting is different from ours and their lemma cannot be used in our setting.

is computationally indistinguishable from f [Mah18]. First, suppose that we just use g instead of f in the above idea. Since g is injective, there is a unique preimage (b_i, x_i) of y_i, in which case the sender knows that the receiver will get $\{(b_i, x_i)\}_{i\in[n]}$ by the standard basis measurement. In this case, the aforementioned problem can be easily resolved by setting $\mathsf{CT} \coloneqq m \oplus H(b_1\|x_1\|...\|b_n\|x_n)$ as the ciphertext. In this case, it is easy to prove that we can extract $\{b_i, x_i\}_{i\in[n]}$ if an adversary obtains some information of m by applying the standard one-way to hiding lemma. However, the obvious problem is that the deletion certificate no longer works for g since the receiver's state collapses to a classical state after the measurement of $\{y_i\}_{i\in[n]}$ and thus the Hadamard basis measurement results in just uniform bits.

Our idea is to take advantages of both of them. Specifically, the sender sends functions $\{\eta_i\}_{i\in[n]}$, where η_i is the g-type function for $i \in S$ and it is the f-type function for $i \in [n] \setminus S$ with a certain set $S \subset [n]$. The receiver generates a set of states. Each state is a superposition of two preimages of a f-type function or a state encoding the unique preimage of a g-type function. The preimages of g-type functions are used for encryption/decryption, and the Hadamard measurement results are used for deletion certificate, whose validity is only checked on positions where f-type functions are used. We also include a ciphertext of the description of the subset S in the ciphertext. The ciphertext enables a legitimate receiver to know which position should be used in the decryption. More precisely, we set $\mathsf{CT} \coloneqq (\mathsf{Enc}(S), m \oplus H(\{b_i, x_i\}_{i\in[S]}))$ where Enc is a PKE scheme with the RNC security.[7][8] A deletion certificate $\{e_i, d_i\}_{i\in[n]}$ is valid if we have $d_i \neq 0$ and $e_i = d_i \cdot (x_{i,0} \oplus x_{i,1})$ for all $i \in [n] \setminus S$. For the security proof of this construction, the amplified adaptive hardcore property cannot be directly used, because it is a property about f-type functions whereas the above construction mixes f-type functions and g-type functions, and what we want to have is the mutually-exclusive property between preimages of g-type functions and deletion certificates of f-type functions. To solve the problem, we introduce a new property which we call *the cut-and-choose adaptive hardcore property* (Lemma 4.1). The cut-and-choose adaptive hardcore property intuitively means that once the receiver issues a deletion certificate $\{e_i, d_i\}_{i\in[n]}$ that is valid for all $i \in [n] \setminus S$ before knowing S, it can no longer generate correct preimages $\{b_i, x_i\}_{i\in[S]}$ even if it receives S later. Intuitively, this holds because the only way to obtain such $\{e_i, d_i\}_{i\in[n]}$ before knowing S would be to measure the states in the Hadamard basis for all $i \in [n]$, in which case the receiver should forget all preimages. We show that the cut-and-choose adaptive hardcore property can be reduced to the adaptive hardcore bit property and injective invariance. The

[7] We require Enc to satisfy the RNC security due to a similar reason to that in Sec. 1.3, which we omit to explain here.

[8] In the actual construction, there is an additional component that is needed for preventing an adversary from decrypting the ciphertext *before* outputting a valid deletion certificate without the decryption key. This is just a security as standard PKE and can be added easily. Thus, we omit this and focus on the security *after* outputting a valid deletion certificate.

new property we show itself is of independent interest, and we believe it will be useful in many other applications of quantum cryptography.

Because the only known construction of NTCF functions [BCM+18, Mah18] assumes the LWE assumption, our construction of the interactive encryption with privately verifiable certified deletion with classical communication is also based on the LWE assumption, and our security proof is done in the QROM. We note that the construction only achieves private verification because verification of deletion certificates requires both of two preimages of f-type functions, which cannot be made public.

Publicly verifiable construction. The above construction is not publicly verifiable because the verification of the validity of (e_i, d_i) requires both preimages $x_{i,0}$ and $x_{i,1}$, which cannot be made public. One might notice that the validity check of the preimage can be done publicly, and might suggest the following construction: preimages are used for deletion certificate, and Hadamard measurement outcomes $\{e_i, d_i\}_{i\in[n]}$ are used as the decryption key of the encryption. Because a valid $\{e_i, d_i\}_{i\in[n]}$ is a witness of an **NP** statement, we could use (extractable) witness encryption [GGSW13, GKP+13] to ensure that a receiver can decrypt the message only if it knows a valid $\{e_i, d_i\}_{i\in[n]}$. However, this idea does not work because the statement of the witness encryption contains private information (i.e., preimages), and witness encryption ensures nothing about privacy of the statement under which a message is encrypted.

Our idea to solve the problem is to use the one-shot signature [AGKZ20]. Roughly speaking, one-shot signatures (with a message space $\{0,1\}$) enable one to generate a classical public key pk along with a quantum secret key sk, which can be used to generate either of a signature σ_0 for message 0 or σ_1 for message 1, but not both. We note that a signature can be verified publicly.

We combine one-shot signatures with extractable witness encryption.[9] The encryption $\mathsf{Enc}(m)$ of a message m in our construction is a ciphertext of witness encryption of message m under the statement corresponding to the verification of one-shot signature for message 0. The deletion certificate is, on the other hand, a one-shot signature for message 1. Once a valid signature of 1 is issued, a valid signature of 0, which is a decryption key of our witness encryption, is no longer possible to generate due to the security of the one-shot signature. This intuitively ensures the certified deletion security of our construction. Because signatures are publicly verifiable, the verification of our construction is also publicly verifiable. In the actual construction, in order to prevent an adversary from decrypting the ciphertext before issuing the deletion certificate, we add an additional layer of encryption, for which we use RNCE due to a similar reason to that in Sec. 1.3.

Unfortunately, the only known construction of the one-shot signature needs classical oracles. Thus, the security proof of existing one-shot signature constructions is a heuristic. Our publicly verifiable construction assumes the existence

[9] We note that a combination of one-shot signatures and extractable witness encryption appeared in the work of Georgiou and Zhandry [GZ20] in a related but different context.

of provably secure one-shot signatures. It is an open question whether we can construct an interactive encryption with publicly verifiable certified deletion with classical communication based on only standard assumptions such as the LWE assumption.

2 Preliminaries

2.1 Notations and Mathematical Tools

We introduce basic notations and mathematical tools used in this paper.

In this paper, $x \leftarrow X$ denotes selecting an element from a finite set X uniformly at random, and $y \leftarrow \mathsf{A}(x)$ denotes assigning to y the output of a probabilistic or deterministic algorithm A on an input x. When we explicitly show that A uses randomness r, we write $y \leftarrow \mathsf{A}(x;r)$. When D is a distribution, $x \leftarrow D$ denotes sampling an element from D. Let $[\ell]$ denote the set of integers $\{1, \cdots, \ell\}$, λ denote a security parameter, and $y := z$ denote that y is set, defined, or substituted by z. For a string $s \in \{0,1\}^\ell$, $s[i]$ denotes i-th bit of s. QPT stands for quantum polynomial time. PPT stands for (classical) probabilistic polynomial time. For a subset $S \subseteq W$ of a set W, $\overline{S}$ is the complement of S, i.e., $\overline{S} := W \setminus S$.

A function $f : \mathbb{N} \to \mathbb{R}$ is a negligible function if for any constant c, there exists $\lambda_0 \in \mathbb{N}$ such that for any $\lambda > \lambda_0$, $f(\lambda) < \lambda^{-c}$. We write $f(\lambda) \leq \mathsf{negl}(\lambda)$ to denote $f(\lambda)$ being a negligible function. A function $g : \mathbb{N} \to \mathbb{R}$ is a noticeable function if there exist constants c and λ_0 such that for any $\lambda \geq \lambda_0$, $g(\lambda) \geq \lambda^{-c}$. The trace distance between two states ρ and σ is given by $\|\rho - \sigma\|_{\mathrm{tr}}$, where $\|A\|_{\mathrm{tr}} := \mathrm{Tr}\sqrt{A^\dagger A}$ is the trace norm. We call a function f a density on X if $f : X \to [0,1]$ such that $\sum_{x \in X} f(x) = 1$. For two densities f_0 and f_1 over the same finite domain X, the Hellinger distance between f_0 and f_1 is $\mathsf{H}^2(f_0, f_1) := 1 - \sum_{x \in X} \sqrt{f_0(x) f_1(x)}$.

2.2 Cryptographic Tools

In this section, we review cryptographic tools used in this paper. Some explanations are omitted, and given in the full version.

Encryption with certified deletion. Broadbent and Islam introduced the notion of encryption with certified deletion [BI20]. Their notion is for secret key encryption (SKE). They consider a setting where a secret key is used only once (that is, one-time SKE). Although it is easy to extend the definition to the reusable secret key setting, we describe the definition for the one-time setting in this section. We provide a definition that is accommodated to the reusable setting in the full version.

Definition 2.1 (One-Time SKE with Certified Deletion (Syntax)). *A one-time secret key encryption scheme with certified deletion is a tuple of QPT algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ *with plaintext space* $\mathcal{M}$ *and key space* $\mathcal{K}$*.*

$\mathsf{KeyGen}(1^\lambda) \to \mathsf{sk}$**:** *The key generation algorithm takes as input the security parameter* 1^λ *and outputs a secret key* $\mathsf{sk} \in \mathcal{K}$.

$\mathsf{Enc}(\mathsf{sk}, m) \to \mathsf{CT}$**:** *The encryption algorithm takes as input* sk *and a plaintext* $m \in \mathcal{M}$ *and outputs a ciphertext* CT.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{CT}) \to m'$ **or** $\bot$**:** *The decryption algorithm takes as input* sk *and* CT *and outputs a plaintext* $m' \in \mathcal{M}$ *or* $\bot$.

$\mathsf{Del}(\mathsf{CT}) \to \mathsf{cert}$**:** *The deletion algorithm takes as input* CT *and outputs a certification* cert.

$\mathsf{Vrfy}(\mathsf{sk}, \mathsf{cert}) \to \top$ **or** $\bot$**:** *The verification algorithm takes* sk *and* cert *and outputs* $\top$ *or* $\bot$.

Definition 2.2 (Correctness for One-Time SKE with Certified Deletion). *There are two types of correctness. One is decryption correctness and the other is verification correctness.*

Decryption correctness: *There exists a negligible function* negl *such that for any* $\lambda \in \mathbb{N}$, $m \in \mathcal{M}$,

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, \mathsf{CT}) \neq m \;\middle|\; \begin{matrix}\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{sk}, m)\end{matrix}\right] \leq \mathsf{negl}(\lambda).$$

Verification correctness: *There exists a negligible function* negl *such that for any* $\lambda \in \mathbb{N}$, $m \in \mathcal{M}$,

$$\Pr\left[\mathsf{Vrfy}(\mathsf{sk}, \mathsf{cert}) = \bot \;\middle|\; \begin{matrix}\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{sk}, m) \\ \mathsf{cert} \leftarrow \mathsf{Del}(\mathsf{CT})\end{matrix}\right] \leq \mathsf{negl}(\lambda).$$

Definition 2.3 (Certified Deletion Security for One-Time SKE). *Let* $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ *be a secret key encryption with certified deletion. We consider the following security experiment* $\mathsf{Exp}^{\mathsf{otsk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda, b)$.

1. *The challenger computes* $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. $\mathcal{A}$ *sends* $(m_0, m_1) \in \mathcal{M}^2$ *to the challenger.*
3. *The challenger computes* $\mathsf{CT}_b \leftarrow \mathsf{Enc}(\mathsf{sk}, m_b)$ *and sends* CT_b *to* $\mathcal{A}$.
4. $\mathcal{A}$ *sends* cert *to the challenger.*
5. *The challenger computes* $\mathsf{Vrfy}(\mathsf{sk}, \mathsf{cert})$. *If the output is* $\bot$, *the challenger sends* $\bot$ *to* $\mathcal{A}$. *If the output is* $\top$, *the challenger sends* sk *to* $\mathcal{A}$.
6. $\mathcal{A}$ *outputs* $b' \in \{0, 1\}$.

We say that the Σ *is OT-CD secure if for any QPT* $\mathcal{A}$, *it holds that*

$$\mathsf{Adv}^{\mathsf{otsk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda) := \left|\Pr\left[\mathsf{Exp}^{\mathsf{otsk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{otsk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda, 1) = 1\right]\right| \leq \mathsf{negl}(\lambda).$$

We sometimes call it one-time SKE with certified deletion if it satisfies OT-CD security.

Remark 2.1. Definition 2.3 intuitively means that once the valid certificate is issued, decrypting the ciphertext becomes impossible. One might think that it would also be possible to define the inverse: once the ciphertext is decrypted, the valid certificate can no longer be issued. However, this property is impossible to achieve due to the decryption correctness (Definition 2.2). In fact, if the quantum decryption algorithm Dec on a quantum ciphertext CT succeeds with probability at least $1 - \mathsf{negl}(\lambda)$, then the gentle measurement lemma guarantees that CT is only negligibly disturbed, from which the valid certificate can be issued.

Remark 2.2. We modified the security definition of certified deletion due to the following reason. Broadbent and Islam [BI20] require ciphertext indistinguishability, which is security as a normal one-time SKE, in addition to the certified deletion security. We observe that these two security notions can be captured in a single security game if we allow the adversary to make a guess even if the deletion certificate is invalid.

We emphasize that in the existing construction of SKE with certified deletion, a secret key is a classical string though a ciphertext must be a quantum state. Broadbent and Islam prove the following theorem.

Theorem 2.1 ([BI20]). *There exists OT-CD secure SKE with certified deletion with $\mathcal{M} = \{0,1\}^{\ell_m}$ and $\mathcal{K} = \{0,1\}^{\ell_k}$ where ℓ_m and ℓ_k are some polynomials, unconditionally.*

Receiver non-committing encryption. We introduce the notion of (public key) receiver non-committing encryption (RNCE) [CFGN96, JL00, CHK05], which is used in Sections 3.2 and 4.3. See the full version for the definition of secret key RNCE.

Definition 2.4 (RNCE (Syntax)). *An RNCE scheme is a tuple of PPT algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Reveal})$ *with plaintext space $\mathcal{M}$.*

$\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk}, \mathsf{aux})$**:** *The key generation algorithm takes as input the security parameter 1^λ and outputs a key pair $(\mathsf{pk}, \mathsf{sk})$ and an auxiliary information aux.*

$\mathsf{Enc}(\mathsf{pk}, m) \to \mathsf{CT}$**:** *The encryption algorithm takes as input pk and a plaintext $m \in \mathcal{M}$ and outputs a ciphertext CT.*

$\mathsf{Dec}(\mathsf{sk}, \mathsf{CT}) \to m'$ **or** $\perp$**:** *The decryption algorithm takes as input sk and CT and outputs a plaintext m' or $\perp$.*

$\mathsf{Fake}(\mathsf{pk}, \mathsf{sk}, \mathsf{aux}) \to \widetilde{\mathsf{CT}}$**:** *The fake encryption algorithm takes pk, sk and aux, and outputs a fake ciphertext $\widetilde{\mathsf{CT}}$.*

$\mathsf{Reveal}(\mathsf{pk}, \mathsf{sk}, \mathsf{aux}, \widetilde{\mathsf{CT}}, m) \to \widetilde{\mathsf{sk}}$**:** *The reveal algorithm takes $\mathsf{pk}, \mathsf{sk}, \mathsf{aux}, \widetilde{\mathsf{CT}}$ and m, and outputs a fake secret key $\widetilde{\mathsf{sk}}$.*

Correctness is the same as that of PKE.

Definition 2.5 (Receiver Non-Committing (RNC) Security). *An RNCE scheme is RNC secure if it satisfies the following. Let $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Reveal})$ be an RNCE scheme. We consider the following security experiment $\mathsf{Exp}^{\mathsf{rec\text{-}nc}}_{\Sigma,\mathcal{A}}(\lambda, b)$.*

1. *The challenger computes* $(\mathsf{pk}, \mathsf{sk}, \mathsf{aux}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and sends* pk *to* $\mathcal{A}$.
2. $\mathcal{A}$ *sends a query* $m \in \mathcal{M}$ *to the challenger.*
3. *The challenger does the following.*
   - *If* $b = 0$*, the challenger generates* $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$ *and returns* $(\mathsf{CT}, \mathsf{sk})$ *to* $\mathcal{A}$.
   - *If* $b = 1$*, the challenger generates* $\widetilde{\mathsf{CT}} \leftarrow \mathsf{Fake}(\mathsf{pk}, \mathsf{sk}, \mathsf{aux})$ *and* $\widetilde{\mathsf{sk}} \leftarrow \mathsf{Reveal}(\mathsf{pk}, \mathsf{sk}, \mathsf{aux}, \widetilde{\mathsf{CT}}, m)$ *and returns* $(\widetilde{\mathsf{CT}}, \widetilde{\mathsf{sk}})$ *to* $\mathcal{A}$.
4. $\mathcal{A}$ *outputs* $b' \in \{0,1\}$.

Let $\mathsf{Adv}^{\mathsf{rec\text{-}nc}}_{\Sigma,\mathcal{A}}(\lambda)$ *be the advantage of the experiment above. We say that the* Σ *is RNC secure if for any QPT adversary, it holds that*

$$\mathsf{Adv}^{\mathsf{rec\text{-}nc}}_{\Sigma,\mathcal{A}}(\lambda) := \left|\Pr\left[\mathsf{Exp}^{\mathsf{rec\text{-}nc}}_{\Sigma,\mathcal{A}}(\lambda,0)=1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{rec\text{-}nc}}_{\Sigma,\mathcal{A}}(\lambda,1)=1\right]\right| \leq \mathsf{negl}(\lambda).$$

Theorem 2.2 ([KNTY19, Section 7.2 in the eprint version]). *If there exists an IND-CPA secure SKE/PKE scheme (against QPT adversaries), there exists an RNC secure secret/public key RNCE scheme (against QPT adversaries) with plaintext space* $\{0,1\}^\ell$*, where* ℓ *is some polynomial, respectively.*

Note that Kitagawa, Nishimaki, Tanaka, and Yamakawa [KNTY19] prove the theorem above for the SKE case in the classical setting, but it is easy to extend their theorem to the post-quantum PKE setting by using post-quantum PKE schemes as building blocks. We also note that the core idea of Kitagawa et al. is based on the observation by Canetti, Halevi, and Katz [CHK05].

3 Public Key Encryption with Certified Deletion

In this section, we define the notion of PKE with certified deletion, which is a natural extension of SKE with certified deletion and present how to achieve PKE with certified deletion from OT-CD secure SKE and IND-CPA secure (standard) PKE.

3.1 Definition of PKE with Certified Deletion

The definition of PKE with certified deletion is an extension of SKE with certified deletion. Note that a verification key for verifying a certificate is generated in the encryption algorithm.

Definition 3.1 (PKE with Certified Deletion (Syntax)). *A PKE with certified deletion is a tuple of QPT algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ *with plaintext space* $\mathcal{M}$.

$\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$**:** *The key generation algorithm takes as input the security parameter* 1^λ *and outputs a classical key pair* $(\mathsf{pk}, \mathsf{sk})$.

$\mathsf{Enc}(\mathsf{pk}, m) \to (\mathsf{vk}, \mathsf{CT})$**:** *The encryption algorithm takes as input the public key* pk *and a plaintext* $m \in \mathcal{M}$ *and outputs a classical verification key* vk *and a quantum ciphertext* CT.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{CT}) \to m'$ **or** $\bot$**:** *The decryption algorithm takes as input the secret key* sk *and the ciphertext* CT*, and outputs a classical plaintext* m' *or* $\bot$*.*

$\mathsf{Del}(\mathsf{CT}) \to \mathsf{cert}$**:** *The deletion algorithm takes as input the ciphertext* CT *and outputs a classical certificate* cert*.*

$\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert}) \to \top$ **or** $\bot$**:** *The verification algorithm takes the verification key* vk *and the certificate* cert*, and outputs* $\top$ *or* $\bot$*.*

Definition 3.2 (Correctness for PKE with Certified Deletion). *There are two types of correctness. One is decryption correctness and the other is verification correctness.*

Decryption correctness: *There exists a negligible function* negl *such that for any* $\lambda \in \mathbb{N}$*,* $m \in \mathcal{M}$*,*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, \mathsf{CT}) \neq m \;\middle|\; \begin{matrix}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ (\mathsf{vk}, \mathsf{CT}) \leftarrow \mathsf{Enc}(\mathsf{pk}, m)\end{matrix}\right] \leq \mathsf{negl}(\lambda).$$

Verification correctness: *There exists a negligible function* negl *such that for any* $\lambda \in \mathbb{N}$*,* $m \in \mathcal{M}$*,*

$$\Pr\left[\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert}) = \bot \;\middle|\; \begin{matrix}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ (\mathsf{vk}, \mathsf{CT}) \leftarrow \mathsf{Enc}(\mathsf{pk}, m) \\ \mathsf{cert} \leftarrow \mathsf{Del}(\mathsf{CT})\end{matrix}\right] \leq \mathsf{negl}(\lambda).$$

Definition 3.3 (Certified Deletion Security for PKE). *Let* $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ *be a PKE with certified deletion scheme. We consider the following security experiment* $\mathsf{Exp}^{\mathsf{pk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda, b)$*.*

1. *The challenger computes* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ *and sends* pk *to* $\mathcal{A}$*.*
2. $\mathcal{A}$ *sends* $(m_0, m_1) \in \mathcal{M}^2$ *to the challenger.*
3. *The challenger computes* $(\mathsf{vk}_b, \mathsf{CT}_b) \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$ *and sends* CT_b *to* $\mathcal{A}$*.*
4. *At some point,* $\mathcal{A}$ *sends* cert *to the challenger.*
5. *The challenger computes* $\mathsf{Vrfy}(\mathsf{vk}_b, \mathsf{cert})$*. If the output is* $\bot$*, it sends* $\bot$ *to* $\mathcal{A}$*. If the output is* $\top$*, it sends* sk *to* $\mathcal{A}$*.*
6. $\mathcal{A}$ *outputs its guess* $b' \in \{0, 1\}$*.*

Let $\mathsf{Adv}^{\mathsf{pk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda)$ *be the advantage of the experiment above. We say that the* Σ *is IND-CPA-CD secure if for any QPT adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathsf{pk\text{-}cert\text{-}del}}_{\mathcal{E},\mathcal{A}}(\lambda) := \left|\Pr\left[\mathsf{Exp}^{\mathsf{pk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{pk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda, 1) = 1\right]\right| \leq \mathsf{negl}(\lambda).$$

3.2 PKE with Certified Deletion from PKE and SKE with Certified Deletion

In this section, we present how to construct a PKE scheme with certified deletion from an SKE scheme with certified deletion and an RNCE scheme, which can be constructed from standard IND-CPA PKE schemes.

Our PKE Scheme. We construct $\Sigma_{\mathsf{pkcd}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ with plaintext space $\mathcal{M}$ from an SKE with certified deletion scheme $\Sigma_{\mathsf{skcd}} = \mathsf{SKE}.(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ with plaintext space $\mathcal{M}$ and key space $\mathcal{K}$ and a public key RNCE scheme $\Sigma_{\mathsf{rnce}} = \mathsf{RNCE}.(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Reveal})$ with plaintext space $\mathcal{K}$.

$\mathsf{KeyGen}(1^\lambda)$**:**
- Generate $(\mathsf{rnce.pk}, \mathsf{rnce.sk}, \mathsf{rnce.aux}) \leftarrow \mathsf{RNCE.KeyGen}(1^\lambda)$ and output $(\mathsf{pk}, \mathsf{sk}) := (\mathsf{rnce.pk}, \mathsf{rnce.sk})$.

$\mathsf{Enc}(\mathsf{pk}, m)$**:**
- Parse $\mathsf{pk} = \mathsf{rnce.pk}$.
- Generate $\mathsf{ske.sk} \leftarrow \mathsf{SKE.Gen}(1^\lambda)$.
- Compute $\mathsf{rnce.CT} \leftarrow \mathsf{RNCE.Enc}(\mathsf{rnce.pk}, \mathsf{ske.sk})$ and $\mathsf{ske.CT} \leftarrow \mathsf{SKE.Enc}(\mathsf{ske.sk}, m)$.
- Output $\mathsf{CT} := (\mathsf{rnce.CT}, \mathsf{ske.CT})$ and $\mathsf{vk} := \mathsf{ske.sk}$.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{CT})$**:**
- Parse $\mathsf{sk} = \mathsf{rnce.sk}$ and $\mathsf{CT} = (\mathsf{rnce.CT}, \mathsf{ske.CT})$.
- Compute $\mathsf{sk}' \leftarrow \mathsf{RNCE.Dec}(\mathsf{rnce.sk}, \mathsf{rnce.CT})$.
- Compute and output $m' \leftarrow \mathsf{SKE.Dec}(\mathsf{sk}', \mathsf{ske.CT})$.

$\mathsf{Del}(\mathsf{CT})$**:**
- Parse $\mathsf{CT} = (\mathsf{rnce.CT}, \mathsf{ske.CT})$.
- Generate $\mathsf{ske.cert} \leftarrow \mathsf{SKE.Del}(\mathsf{ske.CT})$.
- Output $\mathsf{cert} := \mathsf{ske.cert}$.

$\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert})$**:**
- Parse $\mathsf{vk} = \mathsf{ske.sk}$ and $\mathsf{cert} = \mathsf{ske.cert}$.
- Output $b \leftarrow \mathsf{SKE.Vrfy}(\mathsf{ske.sk}, \mathsf{ske.cert})$.

Correctness. The decryption and verification correctness easily follow from the correctness of Σ_{rnce} and Σ_{skcd}.

Security. We prove the following theorem.

Theorem 3.1. *If Σ_{rnce} is RNC secure and Σ_{skcd} is OT-CD secure, Σ_{pkcd} is IND-CPA-CD secure.*

Proof of Theorem 3.1. Let $\mathcal{A}$ be a QPT adversary and $b \in \{0,1\}$ be a bit. We define the following hybrid game $\mathsf{Hyb}(b)$.

$\mathsf{Hyb}(b)$**:** This is the same as $\mathsf{Exp}^{\mathsf{pk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{pkcd}},\mathcal{A}}(\lambda, b)$ except that the challenger generate the target ciphertext as follows. It generates $\mathsf{ske.sk} \leftarrow \mathsf{SKE.Gen}(1^\lambda)$ and computes $\mathsf{rnce.CT}^* \leftarrow \mathsf{RNCE.Fake}(\mathsf{rnce.pk}, \mathsf{rnce.sk}, \mathsf{rnce.aux})$ and $\mathsf{ske.CT}^* \leftarrow \mathsf{SKE.Enc}(\mathsf{ske.sk}, m_b)$. The target ciphertext is $\mathsf{CT}^* := (\mathsf{rnce.CT}^*, \mathsf{ske.CT}^*)$. In addition, we reveal $\widetilde{\mathsf{sk}} \leftarrow \mathsf{Reveal}(\mathsf{rnce.pk}, \mathsf{rnce.sk}, \mathsf{rnce.aux}, \mathsf{rnce.CT}^*, \mathsf{ske.sk})$ instead of $\mathsf{rnce.sk}$.

Proposition 3.1. *If Σ_{rnce} is RNC secure,* $\left|\Pr\left[\mathsf{Exp}^{\mathsf{pk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{pkcd}},\mathcal{A}}(\lambda, b) = 1\right] - \Pr[\mathsf{Hyb}(b) = 1]\right| \leq \mathsf{negl}(\lambda)$.

Proof of Proposition 3.1. We construct an adversary $\mathcal{B}_{\mathsf{rnce}}$ that breaks the RNC security of Σ_{rnce} by assuming that $\mathcal{A}$ distinguishes these two experiments. First, $\mathcal{B}_{\mathsf{rnce}}$ is given $\mathsf{rnce.pk}$ from the challenger of $\mathsf{Exp}^{\mathsf{rec\text{-}nc}}_{\Sigma_{\mathsf{rnce}},\mathcal{B}_{\mathsf{rnce}}}(\lambda, b')$ for $b' \in \{0,1\}$. $\mathcal{B}_{\mathsf{rnce}}$ generates $\mathsf{ske.sk} \leftarrow \mathsf{SKE.Gen}(1^\lambda)$ and sends $\mathsf{rnce.pk}$ to $\mathcal{A}$. When $\mathcal{A}$ sends (m_0, m_1), $\mathcal{B}_{\mathsf{rnce}}$ sends $\mathsf{ske.sk}$ to the challenger of $\mathsf{Exp}^{\mathsf{rec\text{-}nc}}_{\Sigma_{\mathsf{rnce}},\mathcal{B}_{\mathsf{rnce}}}(\lambda, b')$, receives $(\mathsf{rnce.CT}^*, \widetilde{\mathsf{sk}})$, and generates $\mathsf{ske.CT} \leftarrow \mathsf{SKE.Enc}(\mathsf{ske.sk}, m_b)$. $\mathcal{B}_{\mathsf{rnce}}$ sends $(\mathsf{rnce.CT}^*, \mathsf{ske.CT})$ to $\mathcal{A}$ as the challenge ciphertext. At some point, $\mathcal{A}$ outputs cert. If $\mathsf{SKE.Vrfy}(\mathsf{ske.sk}, \mathsf{cert}) = \top$, $\mathcal{B}_{\mathsf{rnce}}$ sends $\widetilde{\mathsf{sk}}$ to $\mathcal{A}$. Otherwise, $\mathcal{B}_{\mathsf{rnce}}$ sends $\bot$ to $\mathcal{A}$. Finally, $\mathcal{B}_{\mathsf{rnce}}$ outputs whatever $\mathcal{A}$ outputs.

- If $b' = 0$, i.e., $(\mathsf{rnce.CT}^*, \widetilde{\mathsf{sk}}) = (\mathsf{RNCE.Enc}(\mathsf{rnce.pk}, \mathsf{ske.sk}), \mathsf{rnce.sk})$, $\mathcal{B}_{\mathsf{rnce}}$ perfectly simulates $\mathsf{Exp}^{\mathsf{pk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{pkcd}},\mathcal{A}}(\lambda, b)$.
- If $b' = 1$, i.e., $(\mathsf{rnce.CT}^*, \widetilde{\mathsf{sk}}) = (\mathsf{RNCE.Fake}(\mathsf{rnce.pk}, \mathsf{rnce.sk}, \mathsf{rnce.aux}), \mathsf{RNCE.Reveal}(\mathsf{rnce.pk}, \mathsf{rnce.sk}, \mathsf{rnce.aux}, \mathsf{rnce.CT}^*, \mathsf{ske.sk}))$, $\mathcal{B}_{\mathsf{rnce}}$ perfectly simulates $\mathsf{Hyb}(b)$.

Thus, if $\mathcal{A}$ distinguishes the two experiments, $\mathcal{B}_{\mathsf{rnce}}$ breaks the RNC security of Σ_{rnce}. This completes the proof. □

Proposition 3.2. *If Σ_{skcd} is OT-CD secure,* $|\Pr[\mathsf{Hyb}(0) = 1] - \Pr[\mathsf{Hyb}(1) = 1]| \leq \mathsf{negl}(\lambda)$.

Proof of Proposition 3.2. We construct an adversary $\mathcal{B}_{\mathsf{skcd}}$ that breaks the OT-CD security of Σ_{skcd} assuming that $\mathcal{A}$ distinguishes these two experiments. $\mathcal{B}_{\mathsf{skcd}}$ plays the experiment $\mathsf{Exp}^{\mathsf{otsk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{skcd}},\mathcal{B}_{\mathsf{skcd}}}(\lambda, b')$ for some $b' \in \{0,1\}$. First, $\mathcal{B}_{\mathsf{skcd}}$ generates $(\mathsf{rnce.pk}, \mathsf{rnce.sk}, \mathsf{rnce.aux}) \leftarrow \mathsf{RNCE.KeyGen}(1^\lambda)$ and sends $\mathsf{rnce.pk}$ to $\mathcal{A}$. When $\mathcal{A}$ sends (m_0, m_1), $\mathcal{B}_{\mathsf{skcd}}$ sends (m_0, m_1) to the challenger of $\mathsf{Exp}^{\mathsf{otsk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{skcd}},\mathcal{B}_{\mathsf{skcd}}}(\lambda, b')$, receives $\mathsf{ske.CT}^*$, and generates $\mathsf{rnce.}\widetilde{\mathsf{CT}} \leftarrow \mathsf{RNCE.Fake}(\mathsf{rnce.pk}, \mathsf{rnce.sk}, \mathsf{rnce.aux})$. $\mathcal{B}_{\mathsf{skcd}}$ sends $(\mathsf{rnce.}\widetilde{\mathsf{CT}}, \mathsf{ske.CT}^*)$ to $\mathcal{A}$ as the challenge ciphertext. At some point, $\mathcal{A}$ outputs cert. $\mathcal{B}_{\mathsf{skcd}}$ passes cert to the challenger of OT-CD SKE. If the challenger returns $\mathsf{ske.sk}$, $\mathcal{B}_{\mathsf{skcd}}$ generates $\widetilde{\mathsf{sk}} \leftarrow \mathsf{RNCE.Reveal}(\mathsf{rnce.pk}, \mathsf{rnce.sk}, \mathsf{rnce.aux}, \mathsf{rnce.}\widetilde{\mathsf{CT}}, \mathsf{ske.sk})$ and sends $\widetilde{\mathsf{sk}}$ to $\mathcal{A}$. Otherwise, $\mathcal{B}_{\mathsf{skcd}}$ sends $\bot$ to $\mathcal{A}$. Finally, $\mathcal{B}_{\mathsf{skcd}}$ outputs whatever $\mathcal{A}$ outputs.

- If $b' = 0$, i.e., $\mathsf{ske.CT}^* = \mathsf{SKE.Enc}(\mathsf{ske.sk}, m_0)$, $\mathcal{B}_{\mathsf{skcd}}$ perfectly simulates $\mathsf{Hyb}(0)$.
- If $b' = 1$, i.e., $\mathsf{ske.CT}^* = \mathsf{SKE.Enc}(\mathsf{ske.sk}, m_1)$, $\mathcal{B}_{\mathsf{skcd}}$ perfectly simulates $\mathsf{Hyb}(1)$.

Thus, if $\mathcal{A}$ distinguishes the two experiments, $\mathcal{B}_{\mathsf{skcd}}$ breaks the OT-CD security. This completes the proof. □

By Propositions 3.1 and 3.2, we immediately obtain Theorem 3.1. □

By Theorems 2.1, 2.2 and 3.1, we immediately obtain the following corollary.

Corollary 3.1. *If there exists IND-CPA secure PKE against QPT adversaries, there exists IND-CPA-CD secure PKE with certified deletion.*

Reusable SKE with certified deletion. We can construct a secret key variant of Σ_{pkcd} above (that is, reusable SKE with certified deletion) by replacing Σ_{rnce} with a secret key RNCE scheme. We omit the proof since it is almost the same as that of Theorem 3.1. By Theorem 2.2 and the fact that OWFs imply (reusable) SKE [HILL99, GGM86], we also obtain the following theorem.

Theorem 3.2. *If there exists OWF against QPT adversaries, there exists IND-CPA-CD secure SKE with certified deletion.*

See the full version for the definition and construction of reusable SKE with certified deletion.

3.3 Attribute-Based Encryption with Certified Deletion

By extending the idea in the previous subsections, we construct ABE with certified deletion based on indistinguishability obfuscation and one-way functions. See the full version for details.

4 Interactive Encryption with Certified Deletion and Classical Communication

In this section, we define the notion of interactive encryption with certified deletion and classical communication, and construct it from the LWE assumption in the QROM. In Sec. 4.1, we present the definition of the interactive encryption with certified deletion and classical communication. In Sec. 4.2, we introduce what we call the *cut-and-choose adaptive hardcore property*, which is used in the security proof of the interactive encryption with certified deletion and classical communication. In Sec. 4.3, we construct an interactive encryption with certified deletion and classical communication, and show its security.

4.1 Definition of Interactive Encryption with Certified Deletion and Classical Communication

We define interactive encryption with certified deletion and classical communication. Note that the encryption algorithm of an interactive encryption with certified deletion and classical communication is interactive unlike PKE with certified deletion and quantum communication as defined in Definition 3.1. It is easy to see that the interaction is necessary if we only allow classical communication.

Definition 4.1 (Interactive Encryption with Certified Deletion and Classical Communication (Syntax)). *An interactive encryption scheme with certified deletion and classical communication is a tuple of quantum algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ *with plaintext space* $\mathcal{M}$*.*

$\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$**:** *The key generation algorithm takes as input the security parameter* 1^λ *and outputs a classical key pair* $(\mathsf{pk}, \mathsf{sk})$*.*

$\mathsf{Enc}\langle\mathcal{S}(\mathsf{pk}, m), \mathcal{R}\rangle \to (\mathsf{vk}, \mathsf{CT})$**:** *This is an interactive process between a classical sender $\mathcal{S}$ with input pk and a plaintext $m \in \mathcal{M}$, and a quantum receiver $\mathcal{R}$ without input. After exchanging classical messages, $\mathcal{S}$ outputs a classical verification key vk and $\mathcal{R}$ outputs a quantum ciphertext CT.*

$\mathsf{Dec}(\mathsf{sk}, \mathsf{CT}) \to m'$ **or** $\bot$**:** *The decryption algorithm takes as input the secret key sk and the ciphertext CT, and outputs a plaintext m' or $\bot$.*

$\mathsf{Del}(\mathsf{CT}) \to \mathsf{cert}$**:** *The deletion algorithm takes as input the ciphertext CT and outputs a classical certificate cert.*

$\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert}) \to \top$ **or** $\bot$**:** *The verification algorithm takes the verification key vk and the certificate CT, and outputs $\top$ or $\bot$.*

Definition 4.2 (Correctness for Interactive Encryption with Certified Deletion and Classical Communication). *There are two types of correctness. One is decryption correctness and the other is verification correctness.*

Decryption correctness: *For any $\lambda \in \mathbb{N}$, $m \in \mathcal{M}$,*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, \mathsf{CT}) \neq m \;\middle|\; \begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)\\ (\mathsf{vk}, \mathsf{CT}) \leftarrow \mathsf{Enc}\langle\mathcal{S}(\mathsf{pk}, m), \mathcal{R}\rangle\end{array}\right] \leq \mathsf{negl}(\lambda).$$

Verification correctness: *For any $\lambda \in \mathbb{N}$, $m \in \mathcal{M}$,*

$$\Pr\left[\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert}) = \bot \;\middle|\; \begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)\\ (\mathsf{vk}, \mathsf{CT}) \leftarrow \mathsf{Enc}\langle\mathcal{S}(\mathsf{pk}, m), \mathcal{R}\rangle\\ \mathsf{cert} \leftarrow \mathsf{Del}(\mathsf{CT})\end{array}\right] \leq \mathsf{negl}(\lambda).$$

Definition 4.3 (Certified Deletion Security for Interactive Encryption with Classical Communication). *Let $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ be a PKE scheme with certified deletion and classical communication. We consider the following security experiment $\mathsf{Exp}^{\mathsf{ccpk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda, b)$.*

1. *The challenger computes $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and sends pk to $\mathcal{A}$.*
2. *$\mathcal{A}$ sends $(m_0, m_1) \in \mathcal{M}^2$ to the challenger.*
3. *The challenger and $\mathcal{A}$ jointly execute $(\mathsf{vk}_b, \mathsf{CT}_b) \leftarrow \mathsf{Enc}\langle\mathcal{S}(\mathsf{pk}, m_b), \mathcal{A}(\mathsf{pk})\rangle$ where the challenger plays the role of the sender and $\mathcal{A}$ plays the role of the receiver.*
4. *At some point, $\mathcal{A}$ sends cert to the challenger.*
5. *The challenger computes $\mathsf{Vrfy}(\mathsf{vk}_b, \mathsf{cert})$. If the output is $\bot$, the challenger sends $\bot$ to $\mathcal{A}$. If the output is $\top$, the challenger sends sk to $\mathcal{A}$.*
6. *$\mathcal{A}$ outputs its guess $b' \in \{0, 1\}$.*

Let $\mathsf{Adv}^{\mathsf{ccpk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda)$ be the advantage of the experiment above. We say that the Σ is IND-CPA-CD secure if for any QPT adversary $\mathcal{A}$, it holds that

$$\mathsf{Adv}^{\mathsf{ccpk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda) := \left|\Pr\left[\mathsf{Exp}^{\mathsf{ccpk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{ccpk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda, 1) = 1\right]\right| \leq \mathsf{negl}(\lambda).$$

4.2 Preparation: Cut-and-choose adaptive hardcore property

We prove that any injective invariant NTCF family satisfies a property which we call the *cut-and-choose adaptive hardcore property*, which is used in the security proof of our interactive encryption with certified deletion with classical communication.

Lemma 4.1 (Cut-and-Choose Adaptive Hardcore Property). *Let $\mathcal{F}$ be an injective invariant NTCF family and $\mathcal{G}$ be the corresponding trapdoor injective family. Then $\mathcal{F}$ and $\mathcal{G}$ satisfy what we call the* cut-and-choose adaptive hardcore property *defined below. For a QPT adversary $\mathcal{A}$ and a positive integer n, we consider the following experiment $\mathsf{Exp}^{\mathsf{cut\text{-}and\text{-}choose}}_{(\mathcal{F},\mathcal{G}),\mathcal{A}}(\lambda,n)$.*

1. *The challenger chooses a uniform subset $S \subseteq [4n]$ such that $|S| = 2n$.*[10]
2. *The challenger generates $(\mathsf{k}_i, \mathsf{td}_i) \leftarrow \mathsf{Gen}_{\mathcal{G}}(1^\lambda)$ for all $i \in S$ and $(\mathsf{k}_i, \mathsf{td}_i) \leftarrow \mathsf{Gen}_{\mathcal{F}}(1^\lambda)$ for all $i \in \overline{S}$ and sends $\{\mathsf{k}_i\}_{i\in[4n]}$ to $\mathcal{A}$.*
3. *$\mathcal{A}$ sends $\{y_i, d_i, e_i\}_{i\in[4n]}$ to the challenger.*
4. *The challenger computes $x_{i,\beta} \leftarrow \mathsf{Inv}_{\mathcal{F}}(\mathsf{td}_i, \beta, y_i)$ for all $(i,\beta) \in \overline{S} \times \{0,1\}$ and checks if $d_i \in G_{\mathsf{k}_i,0,x_{i,0}} \cap G_{\mathsf{k}_i,1,x_{i,1}}$ and $e_i = d_i \cdot (J(x_{i,0}) \oplus J(x_{i,1}))$ hold for all $i \in \overline{S}$. If they do not hold for some $i \in \overline{S}$, the challenger immediately aborts and the experiment returns 0.*
5. *The challenger sends S to $\mathcal{A}$.*
6. *$\mathcal{A}$ sends $\{b_i, x_i\}_{i\in S}$ to the challenger.*
7. *The challenger checks if $\mathsf{Chk}_{\mathcal{G}}(\mathsf{k}_i, b_i, x_i, y_i) = 1$ holds for all $i \in S$. If this holds for all $i \in S$, the experiment returns 1. Otherwise, it returns 0.*

Then for any n such that $n \leq \mathrm{poly}(\lambda)$ and $n = \omega(\log\lambda)$, it holds that

$$\mathsf{Adv}^{\mathsf{cut\text{-}and\text{-}choose}}_{(\mathcal{F},\mathcal{G}),\mathcal{A}}(\lambda,n) := \Pr\left[\mathsf{Exp}^{\mathsf{cut\text{-}and\text{-}choose}}_{(\mathcal{F},\mathcal{G}),\mathcal{A}}(\lambda,n) = 1\right] \leq \mathsf{negl}(\lambda).$$

Its proof is given in the full version.

4.3 Construction

We construct an interactive encryption scheme with certified deletion and classical communication $\Sigma_{\mathsf{cccd}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ with plaintext space $\mathcal{M} = \{0,1\}^\ell$ from an NTCF family $\mathcal{F}$ with the corresponding trapdoor injective family $\mathcal{G}$ for which we use similar notations as in [Mah18], a public key RNCE scheme $\Sigma_{\mathsf{rnce}} = \mathsf{RNCE}.(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Reveal})$ with plaintext space $\{S \subseteq [4n] : |S| = 2n\}$ where n is a positive integer such that $n \leq \mathrm{poly}(\lambda)$ and $n = \omega(\log\lambda)$ and we just write S to mean the description of the set S by abuse of notation, a OW-CPA secure PKE scheme $\Sigma_{\mathsf{ow}} = \mathsf{OW}.(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with plaintext space $\{0,1\}^\lambda$, and a hash function H from $\{0,1\}^\lambda \times (\{0,1\} \times \mathcal{X})^{2n}$ to $\{0,1\}^\ell$ modeled as a quantumly-accessible random oracle.

[10] We can also take $S \subseteq [2n]$ such that $|S| = n$, but we do as above just for convenience in the proof.

$\mathsf{KeyGen}(1^\lambda)$**:**

- Generate $(\mathsf{rnce.pk}, \mathsf{rnce.sk}, \mathsf{rnce.aux}) \leftarrow \mathsf{RNCE.KeyGen}(1^\lambda)$ and $(\mathsf{ow.pk}, \mathsf{ow.sk}) \leftarrow \mathsf{OW.KeyGen}(1^\lambda)$ and output $(\mathsf{pk}, \mathsf{sk}) := ((\mathsf{rnce.pk}, \mathsf{ow.pk}), (\mathsf{rnce.sk}, \mathsf{ow.sk}))$.

$\mathsf{Enc}\langle\mathcal{S}(\mathsf{pk}, m), \mathcal{R}\rangle$**:** This is an interactive protocol between a sender $\mathcal{S}$ with input (pk, m) and a receiver $\mathcal{R}$ without input that works as follows.

- $\mathcal{S}$ parses $\mathsf{pk} = (\mathsf{rnce.pk}, \mathsf{ow.pk})$.
- $\mathcal{S}$ chooses a uniformly random subset $S \subseteq [4n]$ such that $|S| = 2n$, generates

$$(\mathsf{k}_i, \mathsf{td}_i) \leftarrow \begin{cases} \mathsf{Gen}_{\mathcal{G}}(1^\lambda) & i \in S \\ \mathsf{Gen}_{\mathcal{F}}(1^\lambda) & i \in \overline{S} \end{cases}$$

  for $i \in [4n]$, and sends $\{\mathsf{k}_i\}_{i\in[4n]}$ to $\mathcal{R}$.
- For $i \in [4n]$, $\mathcal{R}$ generates a quantum state

$$|\psi'_i\rangle = \begin{cases} \frac{1}{\sqrt{|\mathcal{X}'|}} \sum_{x\in\mathcal{X}, y\in\mathcal{Y}, b\in\{0,1\}} \sqrt{(g_{\mathsf{k}_i,b}(x))(y)}|b, x\rangle|y\rangle & (i \in S) \\ \frac{1}{\sqrt{|\mathcal{X}'|}} \sum_{x\in\mathcal{X}, y\in\mathcal{Y}, b\in\{0,1\}} \sqrt{(f'_{\mathsf{k}_i,b}(x))(y)}\,|b, x\rangle\,|y\rangle & (i \in \overline{S}) \end{cases}$$

  by using Samp, measure the last register to obtain $y_i \in \mathcal{Y}$, and let $|\phi'_i\rangle$ be the post-measurement state where the measured register is discarded. Note that this can be done without knowing S since $\mathsf{Samp}_{\mathcal{F}} = \mathsf{Samp}_{\mathcal{G}}$, which is just denoted by Samp. Then, we can see that for all $i \in [4n]$, $|\phi'_i\rangle$ has a negligible trace distance from the following state:

$$|\phi_i\rangle = \begin{cases} |b_i\rangle\,|x_i\rangle & (i \in S) \\ \frac{1}{\sqrt{2}}\left(|0\rangle\,|x_{i,0}\rangle + |1\rangle\,|x_{i,1}\rangle\right) & (i \in \overline{S}) \end{cases}$$

  where $(x_i, b_i) \leftarrow \mathsf{Inv}_{\mathcal{G}}(\mathsf{td}_i, y_i)$ for $i \in S$ and $x_{i,\beta} \leftarrow \mathsf{Inv}_{\mathcal{F}}(\mathsf{td}_i, \beta, y_i)$ for $(i, \beta) \in \overline{S} \times \{0, 1\}$.[11] $\mathcal{R}$ sends $\{y_i\}_{i\in[4n]}$ to $\mathcal{S}$ and keeps $\{|\phi'_i\rangle\}_{i\in[4n]}$.
- $\mathcal{S}$ chooses $K \leftarrow \{0,1\}^\lambda$ and computes $(b_i, x_i) \leftarrow \mathsf{Inv}_{\mathcal{G}}(\mathsf{td}_i, y_i)$ for all $i \in S$. If $\mathsf{Chk}_{\mathcal{G}}(\mathsf{k}_i, b_i, x_i, y_i) = 0$ for some $i \in S$, $\mathcal{S}$ returns $\bot$ to $\mathcal{R}$. Otherwise, let $i_1, ..., i_{2n}$ be the elements of S in the ascending order. $\mathcal{S}$ sets $Z := (K, (b_{i_1}, x_{i_1}), (b_{i_2}, x_{i_2}), ..., (b_{i_{2n}}, x_{i_{2n}}))$, computes

$$\begin{aligned} &\mathsf{rnce.CT} \leftarrow \mathsf{RNCE.Enc}(\mathsf{rnce.pk}, S), \\ &\mathsf{ow.CT} \leftarrow \mathsf{OW.Enc}(\mathsf{ow.pk}, K), \\ &\mathsf{CT}_{\mathsf{msg}} := m \oplus H(Z), \end{aligned}$$

  and sends $(\mathsf{rnce.CT}, \mathsf{ow.CT}, \mathsf{CT}_{\mathsf{msg}})$ to $\mathcal{R}$.
- $\mathcal{S}$ outputs $\mathsf{vk} := \{\mathsf{td}_i, y_i\}_{i\in\overline{S}}$ and $\mathcal{R}$ outputs $\mathsf{CT} := (\{|\phi'_i\rangle\}_{i\in[4n]}, \mathsf{rnce.CT}, \mathsf{ow.CT}, \mathsf{CT}_{\mathsf{msg}})$.

$\mathsf{Dec}(\mathsf{sk}, \mathsf{CT})$**:**

- Parse $\mathsf{sk} = (\mathsf{rnce.sk}, \mathsf{ow.sk})$ and $\mathsf{CT} = (\{|\phi'_i\rangle\}_{i\in[4n]}, \mathsf{rnce.CT}, \mathsf{ow.CT}, \mathsf{CT}_{\mathsf{msg}})$.

[11] Indeed, $|\phi'_i\rangle = |\phi_i\rangle$ for $i \in S$.

– Compute $S' \leftarrow \mathsf{RNCE.Dec}(\mathsf{rnce.sk}, \mathsf{rnce.CT})$.
– Compute $K' \leftarrow \mathsf{OW.Dec}(\mathsf{ow.sk}, \mathsf{ow.CT})$.
– For all $i \in S'$, measure $|\phi'_i\rangle$ in the computational basis and let (b'_i, x'_i) be the outcome.
– Compute and output $m' := \mathsf{CT}_{\mathsf{msg}} \oplus H(K', (b'_{i_1}, x'_{i_1}), (b'_{i_2}, x'_{i_2}), ..., (b'_{i_{2n}}, x'_{i_{2n}}))$ where $i_1, ..., i_{2n}$ are the elements of S' in the ascending order.[12]

$\mathsf{Del}(\mathsf{CT})$:
– Parse $\mathsf{CT} = (\{|\phi'_i\rangle\}_{i\in[4n]}, \mathsf{rnce.CT}, \mathsf{ow.CT}, \mathsf{CT}_{\mathsf{msg}})$.
– For all $i \in [4n]$, evaluate the function J on the second register of $|\phi'_i\rangle$. That is, apply an isometry that maps $|b, x\rangle$ to $|b, J(x)\rangle$ to $|\phi'_i\rangle$. (Note that this can be done efficiently since J is injective and efficiently invertible.) Let $|\phi''_i\rangle$ be the resulting state.
– For all $i \in [4n]$, measure $|\phi''_i\rangle$ in the Hadamard basis and let (e_i, d_i) be the outcome.
– Output $\mathsf{cert} := \{(e_i, d_i)\}_{i\in[4n]}$.

$\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert})$:
– Parse $\mathsf{vk} = \{\mathsf{td}_i, y_i\}_{i\in\overline{S}}$ and $\mathsf{cert} = \{(e_i, d_i)\}_{i\in[4n]}$.
– Compute $x_{i,\beta} \leftarrow \mathsf{Inv}_{\mathcal{F}}(\mathsf{td}_i, \beta, y_i)$ for all $(i, \beta) \in \overline{S} \times \{0, 1\}$.
– Output $\top$ if $d_i \in G_{\mathsf{k}_i,0,x_{i,0}} \cap G_{\mathsf{k}_i,1,x_{i,1}}$ and $e_i = d_i \cdot (J(x_{i,0}) \oplus J(x_{i,1}))$ hold for all $i \in \overline{S}$ and output $\bot$ otherwise.

Correctness. As observed in the description, $|\phi'_i\rangle$ in the ciphertext has a negligible trace distance from $|\phi_i\rangle$. Therefore, it suffices to prove correctness assuming that $|\phi'_i\rangle$ is replaced with $|\phi_i\rangle$. After this replacement, decryption correctness clearly holds assuming correctness of Σ_{rnce} and Σ_{ow}.

We prove verification correctness below. For $i \in \overline{S}$, if we apply J to the second register of $|\phi_i\rangle$ and then apply Hadamard transform for both registers as in Del, then the resulting state can be written as

$$\begin{aligned}
&2^{-\frac{w+2}{2}} \sum_{d,b,e} (-1)^{d\cdot J(x_{i,b})\oplus eb} |e\rangle |d\rangle \\
&= 2^{-\frac{w}{2}} \sum_{d\in\{0,1\}^w} (-1)^{d\cdot J(x_{i,0})} |d \cdot (J(x_{i,0}) \oplus J(x_{i,1}))\rangle |d\rangle .
\end{aligned}$$

Therefore, the measurement result is (e_i, d_i) such that $e_i = d_i \cdot (J(x_{i,0}) \oplus J(x_{i,1}))$ for a uniform $d_i \leftarrow \{0, 1\}^w$. By the definition of an NTCF family [Mah18], it holds that $d_i \in G_{\mathsf{k}_i,0,x_{i,0}} \cap G_{\mathsf{k}_i,1,x_{i,1}}$ except for a negligible probability. Therefore, the certificate $\mathsf{cert} = \{(e_i, d_i)\}_{i\in[4n]}$ passes the verification by Vrfy with overwhelming probability.

Security. We prove the following theorem.

Theorem 4.1. *If Σ_{rnce} is RNC secure, Σ_{ow} is OW-CPA secure, and $\mathcal{F}$ is an injective invariant NTCF family with the corresponding injective trapdoor family $\mathcal{G}$, Σ_{cccd} is IND-CPA-CD secure in the QROM where H is modeled as a quantumly-accessible random oracle.*

[12] If $S' = \bot$ or $K' = \bot$, output $\bot$.

Proof of Theorem 4.1. What we need to prove is that for any QPT adversary $\mathcal{A}$, it holds that

$$\mathsf{Adv}^{\mathsf{ccpk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{cccd}},\mathcal{A}}(\lambda) := \left|\Pr\left[\mathsf{Exp}^{\mathsf{ccpk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{cccd}},\mathcal{A}}(\lambda,0)=1\right]-\Pr\left[\mathsf{Exp}^{\mathsf{ccpk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{cccd}},\mathcal{A}}(\lambda,1)=1\right]\right| \leq \mathsf{negl}(\lambda).$$

Let $q=\mathrm{poly}(\lambda)$ be the maximum number of $\mathcal{A}$'s random oracle queries. For clarity, we describe how $\mathsf{Exp}^{\mathsf{ccpk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{cccd}},\mathcal{A}}(\lambda,b)$ works below.

1. A uniformly random function H from $\{0,1\}^\lambda\times(\{0,1\}\times\mathcal{X})^{2n}$ to $\{0,1\}^\ell$ is chosen, and $\mathcal{A}$ can make arbitrarily many quantum queries to H at any time in the experiment.
2. The challenger generates $(\mathsf{rnce.pk},\mathsf{rnce.sk},\mathsf{rnce.aux})\leftarrow\mathsf{RNCE.KeyGen}(1^\lambda)$ and $(\mathsf{ow.pk},\mathsf{ow.sk})\leftarrow\mathsf{OW.KeyGen}(1^\lambda)$ and sends $\mathsf{pk}:=(\mathsf{rnce.pk},\mathsf{ow.pk})$ to $\mathcal{A}$.
3. $\mathcal{A}$ sends $(m_0,m_1)\in\mathcal{M}^2$ to the challenger.
4. The challenger chooses a uniform subset $S\subseteq[4n]$ such that $|S|=2n$, generates
$$(\mathsf{k}_i,\mathsf{td}_i)\leftarrow\begin{cases}\mathsf{Gen}_{\mathcal{G}}(1^\lambda) & i\in S\\ \mathsf{Gen}_{\mathcal{F}}(1^\lambda) & i\in\overline{S}\end{cases}$$
for $i\in[4n]$, and sends $\{\mathsf{k}_i\}_{i\in[4n]}$ to $\mathcal{A}$.
5. $\mathcal{A}$ sends $\{y_i\}_{i\in[4n]}$ to the challenger.
6. The challenger chooses $K\leftarrow\{0,1\}^\lambda$ and computes $(b_i,x_i)\leftarrow\mathsf{Inv}_{\mathcal{G}}(\mathsf{td}_i,y_i)$ for all $i\in S$. If $\mathsf{Chk}_{\mathcal{G}}(\mathsf{k}_i,b_i,x_i,y_i)=0$ for some $i\in S$, the challenger sets $Z:=\mathsf{null}$ and returns $\bot$ to $\mathcal{A}$ where null is a special symbol indicating that Z is undefined. Otherwise, let $i_1,\ldots,i_{2n}$ be the elements of S in the ascending order. The challenger sets $Z:=(K,(b_{i_1},x_{i_1}),(b_{i_2},x_{i_2}),...,(b_{i_{2n}},x_{i_{2n}}))$, computes
$$\begin{aligned}&\mathsf{rnce.CT}\leftarrow\mathsf{RNCE.Enc}(\mathsf{rnce.pk},S),\\ &\mathsf{ow.CT}\leftarrow\mathsf{OW.Enc}(\mathsf{ow.pk},K),\\ &\mathsf{CT}_{\mathsf{msg}}:=m_b\oplus H(Z),\end{aligned}$$
and sends $(\mathsf{rnce.CT},\mathsf{ow.CT},\mathsf{CT}_{\mathsf{msg}})$ to $\mathcal{A}$.
7. $\mathcal{A}$ sends $\mathsf{cert}=\{(e_i,d_i)\}_{i\in[4n]}$ to the challenger.
8. The challenger computes $x_{i,\beta}\leftarrow\mathsf{Inv}_{\mathcal{F}}(\mathsf{td}_i,\beta,y_i)$ for all $(i,\beta)\in\overline{S}\times\{0,1\}$. If $d_i\in G_{\mathsf{k}_i,0,x_{i,0}}\cap G_{\mathsf{k}_i,1,x_{i,1}}$ and $e_i=d_i\cdot(J(x_{i,0})\oplus J(x_{i,1}))$ hold for all $i\in\overline{S}$, sends $\mathsf{sk}:=(\mathsf{rnce.sk},\mathsf{ow.sk})$ to $\mathcal{A}$, and otherwise sends $\bot$ to $\mathcal{A}$.
9. $\mathcal{A}$ outputs b'. The output of the experiment is b'.

We define the following sequence of hybrids.

$\mathsf{Hyb}_1(b)$**:** Let $\mathtt{Reveal}_{\mathsf{sk}}$ be the event that the challenger sends sk in Step 8. $\mathsf{Hyb}_1(b)$ is identical to $\mathsf{Exp}^{\mathsf{ccpk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{cccd}},\mathcal{A}}(\lambda,b)$ except that K is chosen at the beginning and the oracle given to $\mathcal{A}$ before $\mathtt{Reveal}_{\mathsf{sk}}$ occurs is replaced with $H_{K\|*\to H'}$, which is H reprogrammed according to H' on inputs whose first entry is K

where H' is another independent random function. More formally, $H_{K\|*\to H'}$ is defined by

$$H_{K\|*\to H'}(K',(b_1,x_1),...,(b_{2n},x_{2n})) \coloneqq \begin{cases} H(K',(b_1,x_1),...,(b_{2n},x_{2n})) & (K' \neq K) \\ H'(K',(b_1,x_1),...,(b_{2n},x_{2n})) & (K' = K) \end{cases}.$$

We note that the challenger still uses H to generate $\mathsf{CT}_{\mathsf{msg}}$ and the oracle after $\mathtt{Reveal}_{\mathsf{sk}}$ occurs is still H similarly to the real experiment. On the other hand, if $\mathtt{Reveal}_{\mathsf{sk}}$ does not occur, the oracle $H_{K\|*\to H'}$ is used throughout the experiment except for the generation of $\mathsf{CT}_{\mathsf{msg}}$.

$\mathsf{Hyb}_2(b)$**:** This is identical to $\mathsf{Hyb}_1(b)$ except that $\mathsf{rnce.CT}$ and $\mathsf{rnce.sk}$ that may be sent to $\mathcal{A}$ in Step 6 and 8 are replaced by

$$\mathsf{rnce.}\widetilde{\mathsf{CT}} \leftarrow \mathsf{RNCE.Fake}(\mathsf{rnce.pk},\mathsf{rnce.sk},\mathsf{rnce.aux}),$$
$$\mathsf{rnce.}\widetilde{\mathsf{sk}} \leftarrow \mathsf{RNCE.Reveal}(\mathsf{rnce.pk},\mathsf{rnce.sk},\mathsf{rnce.aux},\mathsf{rnce.}\widetilde{\mathsf{CT}},S).$$

$\mathsf{Hyb}_3(b)$**:** This is identical to $\mathsf{Hyb}_2(b)$ except that the oracle given to $\mathcal{A}$ after $\mathtt{Reveal}_{\mathsf{sk}}$ occurs is replaced with $H_{Z\to r}$, which is H reprogrammed to output r on input $Z = (K,(b_{i_1},x_{i_1}),...,(b_{i_{2n}},x_{i_{2n}}))$ where r is an independently random ℓ-bit string. More formally, $H_{Z\to r}$ is defined by

$$H_{Z\to r}(Z') \coloneqq \begin{cases} H(Z') & (Z' \neq Z) \\ r & (Z' = Z) \end{cases}.$$

Note that we have $H_{Z\to r} = H$ if $Z = \mathsf{null}$, i.e., if $\mathsf{Chk}_{\mathcal{G}}(\mathsf{k}_i,b_i,x_i,y_i) = 0$ for some $i \in S$ in Step 6.

Proposition 4.1. *If Σ_{ow} is OW-CPA secure,* $\left|\Pr\left[\mathsf{Exp}^{\mathsf{ccpk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{cccd}},\mathcal{A}}(\lambda,b)=1\right] - \Pr[\mathsf{Hyb}_1(b)=1]\right| \leq \mathsf{negl}(\lambda)$.

The only difference between $\mathsf{Exp}^{\mathsf{ccpk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{cccd}},\mathcal{A}}(\lambda,b)$ and $\mathsf{Hyb}_1(b)$ is that the random oracle is reprogrammed on inputs with prefix K before $\mathtt{Reveal}_{\mathsf{sk}}$ occurs. By applying the one-way to hiding lemma [AHU19], if $\mathcal{A}$ distinguishes these two games, then we can use it to extract K before $\mathtt{Reveal}_{\mathsf{sk}}$ occurs. This contradicts the OW-CPA security of Σ_{ow}. Therefore, these two games are indistinguishable. The full proof is given in the full version.

Proposition 4.2. *If Σ_{rnce} is RNC secure,* $|\Pr[\mathsf{Hyb}_1(b)=1] - \Pr[\mathsf{Hyb}_2(b)=1]| \leq \mathsf{negl}(\lambda)$.

This can be reduced to the RNC security of Σ_{rnce} in a similar manner to that in Proposition 3.1. The full proof is given in the full version.

Proposition 4.3. *If $\mathcal{F}$ and $\mathcal{G}$ satisfy the cut-and-choose adaptive hardcore property described in Lemma 4.1,* $|\Pr[\mathsf{Hyb}_2(b)=1] - \Pr[\mathsf{Hyb}_3(b)=1]| \leq \mathsf{negl}(\lambda)$.

The only difference between $\mathsf{Hyb}_2(b)$ and $\mathsf{Hyb}_3(b)$ is that the random oracle is reprogrammed on $Z = (K, (b_{i_1}, x_{i_1}), ..., (b_{i_{2n}}, x_{i_{2n}}))$ after $\mathtt{Reveal}_{\mathsf{sk}}$ occurs. By applying the one-way to hiding lemma [AHU19], if $\mathcal{A}$ distinguishes these two games, then we can use it to extract $Z = (K, (b_{i_1}, x_{i_1}), ..., (b_{i_{2n}}, x_{i_{2n}}))$ after $\mathtt{Reveal}_{\mathsf{sk}}$ occurs. This can be used to break the cut-and-choose adaptive hardcore property of $(\mathcal{F}, \mathcal{G})$. Therefore, these two games are indistinguishable. The full proof is given in the full version.

Proposition 4.4. *It holds that* $\Pr[\mathsf{Hyb}_3(0) = 1] = \Pr[\mathsf{Hyb}_3(1) = 1]$.

Proof of Proposition 4.4. In Hyb_3, the challenger queries H while the adversary queries $H_{K\|*\to H'}$ or $H_{Z\to r}$. Therefore, $H(Z)$ is used only for generating $\mathsf{CT}_{\mathsf{msg}}$ in Hyb_3 and thus $\mathsf{CT}_{\mathsf{msg}}$ is an independently uniform string regardless of b from the view of the adversary. Therefore Proposition 4.4 holds. □

By combining Propositions 4.1 to 4.4 Theorem 4.1 is proven. □

4.4 Publicly Verifiable Construction

The scheme given in the previous subsection is privately verifiable. We construct publicly verifiable interactive encryption with certified deletion and classical communication based on extractable witness encryption and one-shot signatures. See the full version for details.

Acknowledgement

TM is supported by the MEXT Q-LEAP, JST FOREST, JST PRESTO No.JPMJPR176A, and the Grant-in-Aid for Scientific Research (B) No.JP19H04066 of JSPS.

References

AGKZ20. R. Amos, M. Georgiou, A. Kiayias, and M. Zhandry. One-shot signatures and applications to hybrid quantum/classical authentication. In *52nd ACM STOC*, pages 255–268. 2020.

AHU19. A. Ambainis, M. Hamburg, and D. Unruh. Quantum Security Proofs Using Semi-classical Oracles. In *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 269–295. 2019.

AK21. P. Ananth and F. Kaleoglu. Uncloneable Encryption, Revisited. *IACR Cryptol. ePrint Arch.*, 2021:412, 2021.

BB84. C. H. Bennett and G. Brassard. Quantum Cryptography: Public Key Distribution and Coin Tossing. In *IEEE International Conference on Computers Systems and Signal Processing*, pages 175–179. IEEE, 1984.

BCM+18. Z. Brakerski, P. Christiano, U. Mahadev, U. V. Vazirani, and T. Vidick. A Cryptographic Test of Quantumness and Certifiable Randomness from a Single Quantum Device. In *59th FOCS*, pages 320–331. 2018.

BDF+11. D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random Oracles in a Quantum World. In *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. 2011.

BGI+12. B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2):6:1–6:48, 2012.

BI20. A. Broadbent and R. Islam. Quantum Encryption with Certified Deletion. In *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 92–122. 2020.

BL20. A. Broadbent and S. Lord. Uncloneable Quantum Encryption via Oracles. In *15th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2020, June 9-12, 2020, Riga, Latvia*, volume 158 of *LIPIcs*, pages 4:1–4:22. 2020.

BR97. M. Bellare and P. Rogaway. Collision-Resistant Hashing: Towards Making UOWHFs Practical. In *CRYPTO'97*, volume 1294 of *LNCS*, pages 470–484. 1997.

CCKW19. A. Cojocaru, L. Colisson, E. Kashefi, and P. Wallden. QFactory: Classically-Instructed Remote Secret Qubits Preparation. In *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 615–645. 2019.

CFGN96. R. Canetti, U. Feige, O. Goldreich, and M. Naor. Adaptively Secure Multi-Party Computation. In *28th ACM STOC*, pages 639–648. 1996.

CHK05. R. Canetti, S. Halevi, and J. Katz. Adaptively-Secure, Non-interactive Public-Key Encryption. In *TCC 2005*, volume 3378 of *LNCS*, pages 150–168. 2005.

CMP20. A. Coladangelo, C. Majenz, and A. Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. *arXiv*, 2009.13865, 2020.

CRW19. X. Coiteux-Roy and S. Wolf. Proving Erasure. *2019 IEEE International Symposium on Information Theory (ISIT)*, 2019.

FM18. H. Fu and C. A. Miller. Local randomness: Examples and application. *Physical Review A*, 97(3), 2018.

GDP16. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46 (general data protection regulation). *Official Journal of the European Union (OJ)*, pages 1–88, 2016.

GGH+16. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate Indistinguishability Obfuscation and Functional Encryption for All Circuits. *SIAM J. Comput.*, 45(3):882–929, 2016.

GGM86. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

GGSW13. S. Garg, C. Gentry, A. Sahai, and B. Waters. Witness encryption and its applications. In *45th ACM STOC*, pages 467–476. 2013.

GKP+13. S. Goldwasser, Y. T. Kalai, R. A. Popa, V. Vaikuntanathan, and N. Zeldovich. How to Run Turing Machines on Encrypted Data. In *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 536–553. 2013.

GPSW06. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In *ACM CCS 2006*, pages 89–98. 2006. Available as Cryptology ePrint Archive Report 2006/309.

GZ20. M. Georgiou and M. Zhandry. Unclonable Decryption Keys. Cryptology ePrint Archive, Report 2020/877, 2020. https://eprint.iacr.org/2020/877.

HILL99. J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A Pseudorandom Generator from any One-way Function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

JL00. S. Jarecki and A. Lysyanskaya. Adaptively Secure Threshold Cryptography: Introducing Concurrency, Removing Erasures. In *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 221–242. 2000.

KNTY19. F. Kitagawa, R. Nishimaki, K. Tanaka, and T. Yamakawa. Adaptively Secure and Succinct Functional Encryption: Improving Security and Efficiency, Simultaneously. In *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 521–551. 2019.

KNY20. F. Kitagawa, R. Nishimaki, and T. Yamakawa. Secure Software Leasing from Standard Assumptions. Cryptology ePrint Archive, Report 2020/1314, 2020. https://eprint.iacr.org/2020/1314.

KT20. S. Kundu and E. Tan. Composably secure device-independent encryption with certified deletion. *arXiv*, 2011.12704, 2020.

Mah18. U. Mahadev. Classical Verification of Quantum Computations. In *59th FOCS*, pages 259–267. 2018.

NY90. M. Naor and M. Yung. Public-key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In *22nd ACM STOC*, pages 427–437. 1990.

NY21. R. Nishimaki and T. Yamakawa. Quantum Encryption with Certified Deletion: Public Key and Attribute-Based. *IACR Cryptol. ePrint Arch.*, 2021:394, 2021.

RS19. R. Radian and O. Sattath. Semi-Quantum Money. *arXiv*, abs/1908.08889, 2019.

SW05. A. Sahai and B. R. Waters. Fuzzy Identity-Based Encryption. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. 2005.

Unr15. D. Unruh. Revocable Quantum Timed-Release Encryption. *J. ACM*, 62(6):49:1–49:76, 2015.

Wie83. S. Wiesner. Conjugate Coding. *SIGACT News*, 15(1):78–88, 1983.