Quantum Encryption with Certified Deletion: Public Key and Attribute-Based

Ryo Nishimaki[1] and Takashi Yamakawa[1]

[1]NTT Secure Platform Laboratories, Tokyo, Japan
{ryo.nishimaki.zk,takashi.yamakawa.ga}@hco.ntt.co.jp

March 24, 2021

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 is deleted. Though they proved that their construction is information theoretically secure, a drawback is that the construction 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.

In this paper, we construct a (reusable-key) public key encryption (PKE) and attribute-based encryption (ABE) with certified deletion. Our PKE with certified deletion is constructed assuming the existence of IND-CPA secure PKE, and our ABE with certified deletion is constructed assuming the existence of indistinguishability obfuscation and one-way function.

Contents

1 Introduction

The no-cloning theorem, which means that a 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 a basis of many quantum cryptographic protocols including quantum money [Wie83] and quantum key distribution [BB84].

Recently, 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, the receiver (or any other party) can no longer recover the message *even if the decryption key is given*. We remark that this functionality is classically impossible to achieve since one can simply copy a classical ciphertext and keep it so that s/he can decrypt it in any later time. They prove security of their construction without relying on any computational assumption, which ensures the information theoretical security. On the other hand, a drawback is that the construction 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.

A possible application scenario of quantum encryption with certified deletion is the following. A user upload encrypted data on quantum cloud. Whenever the user wishes to delete the data, the cloud generates the certificate of the deletion 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 is accidentally leaked later. 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 no advantage of uploading the data to cloud: If the user has such a 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 would be possible if we can extend the quantum encryption with certified deletion to public key encryption (PKE). Remark that the one-time restriction is automatically resolved for PKE, which can be seen 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. Such an access control has been realized by attribute-based encryption (ABE) [SW05, GPSW06] in classical cryptography. Thus, it would be useful if we have ABE with certified deletion.

1.1 Our Result

We give formal definitions of PKE and ABE with certified deletion, and give 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 the collusion-resistance, i.e., it is secure against adversaries that obtain arbitrarily many decryption keys.

We note that our constructions rely on computational assumptions and thus not information theoretically secure unlike the construction in [BI20]. This is unavoidable since even plain PKE or ABE cannot be information theoretically secure.

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 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.

1.2 Related work

Before the work of [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 construction of quantum encryption with certified deletion in [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 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 massage can no longer be recovered. As observed in [BI20], an important 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 in [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.

2 Preliminaries

2.1 Notations

We introduce basic notations 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. QPT stands for quantum polynomial time. 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.

2.2 Cryptographic Tools

In this section, we review cryptographic tools used in this paper.

Public key encryption.

Definition 2.1. *A public key encryption scheme* $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is a triple of algorithm: a key generation* KeyGen*, an encryption algorithm* Enc *and a decryption algorithm* Dec*.*

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

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

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

Definition 2.2 (Correctness for PKE). *For any* $\lambda \in \mathbb{N}$*,* $m \in \mathcal{M}$*,*

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

Definition 2.3 (IND-CPA security). *Let $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a PKE scheme. For QPT adversaries $\mathcal{A}$, we define the following security experiment $\mathsf{Exp}^{\mathsf{ind\text{-}cpa}}_{\Sigma,\mathcal{A}}(\lambda, b)$.*

1. *The challenger generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, and sends the pk to the $\mathcal{A}$.*
2. *The $\mathcal{A}$ sends (m_0, m_1) to the challenger.*
3. *The challenger computes $\mathsf{CT}_b \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$, and sends CT_b to the $\mathcal{A}$.*
4. *The $\mathcal{A}$ outputs $b' \in \{0, 1\}$. This is the output of the experiment.*

Let $\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\Sigma,\mathcal{A}}(\lambda)$ be the advantage of the game. We say that the Σ is ϵ-IND-CPA secure if for any QPT $\mathcal{A}$, it holds that

$$\mathsf{Adv}^{\mathsf{ind\text{-}cpa}}_{\Sigma,\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{ind\text{-}cpa}}_{\Sigma,\mathcal{A}}(\lambda, 0) = 1] - \Pr[\mathsf{Exp}^{\mathsf{ind\text{-}cpa}}_{\Sigma,\mathcal{A}}(\lambda, 1) = 1]| \leq \epsilon.$$

When ϵ is negligible, we omit ϵ and say Σ is IND-CPA secure.

There are many IND-CPA secure PKE schemes against QPT adversaries under standard cryptographic assumptions. A famous one is Regev PKE scheme, which is IND-CPA secure if the learning with errors (LWE) assumption holds against QPT adversaries [Reg09]. See the references for the LWE assumption and constructions of post-quantum secure PKE [Reg09, GPV08].

Attribute-based encryption. We review the notion of *(key-policy) attribute-based encryption (ABE) [SW05, GPSW06].

Definition 2.4 (Attribute-Based Encryption (Syntax)). *An ABE scheme is a tuple of algorithms $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with plaintext space $\mathcal{M}$, attribute space $\mathcal{X}$, and policy space $\mathcal{P}$.*

$\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{msk})$**:** *The setup algorithm takes as input the security parameter and outputs a public key pk and a master secret key msk.*

$\mathsf{KeyGen}(\mathsf{msk}, P) \to \mathsf{sk}_P$**:** *The key generation algorithm takes as msk and a policy $P \in \mathcal{P}$, and outputs a secret key sk_P.*

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

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

Definition 2.5 (Correctness for ABE). *For any $\lambda \in \mathbb{N}$, $m \in \mathcal{M}$, $P \in \mathcal{P}$, and $X \in \mathcal{X}$ such that $P(X) = \top$,*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_P, \mathsf{CT}_X) \neq m \;\middle|\; \begin{array}{l}(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{sk}_P \leftarrow \mathsf{KeyGen}(\mathsf{msk}, P) \\ \mathsf{CT}_X \leftarrow \mathsf{Enc}(\mathsf{pk}, X, m)\end{array}\right] \leq \mathsf{negl}(\lambda).$$

Definition 2.6 (IND-sel-CPA Security for ABE). *Let $\Sigma = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an ABE scheme with plaintext space $\mathcal{M}$, attribute space $\mathcal{X}$, and policy space $\mathcal{P}$. We consider the following the security experiment $\mathsf{Exp}^{\mathsf{ind\text{-}sel\text{-}}cpa}_{\Sigma,\mathcal{A}}(\lambda, b)$.*

1. *$\mathcal{A}$ declare the target attribute $X^* \in \mathcal{X}$ and sends it to the challenger.*
2. *The challenger computes $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends pk to $\mathcal{A}$.*
3. *$\mathcal{A}$ sends a query $P \in \mathcal{P}$ to the challenger and it returns $\mathsf{sk}_P \leftarrow \mathsf{KeyGen}(\mathsf{msk}, P)$ to $\mathcal{A}$. This process can be repeated polynomially many times.*
4. *$\mathcal{A}$ sends $(m_0, m_1) \in \mathcal{M}^2$ to the challenger.*

5. *The challenger computes* $\mathsf{CT}_b \leftarrow \mathsf{Enc}(\mathsf{pk}, X^*, m_b)$ *and sends* CT_b *to the* $\mathcal{A}$.

6. *Again,* $\mathcal{A}$ *can send key queries.*

7. *At some point,* $\mathcal{A}$ *outputs the* $b' \in \{0,1\}$.

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

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

Theorem 2.7 ([GVW15]). *If the LWE assumption holds against QPT adversaries, there exists IND-sel-CPA secure ABE scheme with* $\mathcal{P} = \mathsf{P/poly}$ *against QPT adversaries.*

Encryption with certified deletion. Broadbent and Islam introduced the notion of encryption with certified deletion [BI20]. Their notion is for secret key encryption (SKE).

Definition 2.8 (Secret Key Encryption with Certified Deletion (Syntax)). *A secret key encryption scheme with certified deletion is a tuple of quantum 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 and outputs a secret key* $\mathsf{sk} \in \mathcal{K}$.

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

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

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

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

Definition 2.9 (Correctness for SKE with Certified Deletion). *There are two types of correctness. One is decryption correctness and the other is verification correctness.*

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

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

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

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

Broadbent and Islam consider a setting where a secret key is used only once (that is, one time SKE), but it is easy to extend the definition to reusable secret key setting.

Definition 2.10 (Certified Deletion Security for 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 the security experiment* $\mathsf{Exp}^{\mathsf{sk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda, b)$.

1. *The challenger computes* $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda)$.

2. $\mathcal{A}$ *sends an encryption query* m *to the challenger and the it returns* $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{sk}, m)$ *to* $\mathcal{A}$*. This process can be repeated polynomially many times.*

3. *$\mathcal{A}$ sends (m_0, m_1) to the challenger.*
4. *The challenger computes $\mathsf{CT}_b \leftarrow \mathsf{Enc}(\mathsf{sk}, m_b)$ and sends CT_b to the $\mathcal{A}$.*
5. *Again, $\mathcal{A}$ can send encryption queries.*
6. *At some point, $\mathcal{A}$ sends cert to the challenger.*
7. *The challenger computes $\mathsf{Vrfy}(\mathsf{sk}, \mathsf{cert})$. If the output is $\bot$, then abort. Otherwise (the output is $\top$), the challenger sends the sk to the $\mathcal{A}$.*
8. *$\mathcal{A}$ receives sk and outputs the $b' \in \{0,1\}$.*

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

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

If $\mathcal{A}$ is not allowed to send any encryption query at the second and fifth items of $\mathsf{Exp}^{\mathsf{sk\text{-}cert\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda,b)$, we say that Σ is ϵ-OT-CD secure. When ϵ is negligible, we omit ϵ and say Σ is IND-CPA-CD (or OT-CD) secure.

We sometimes say reusable (resp. one-time) SKE with certified deletion if it satisfies IND-CPA-CD (resp. OT-CD) security.

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 2.11 ([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 receiver non-committing encryption (RNCE) [CFGN96, JL00, CHK05], which is used in Section 3.2. We sometimes simply write NCE to mean RNCE since we consider only RNCE in this paper.

Definition 2.12 (RNCE (syntax)). *An NCE scheme is a tuple of 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 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 and outputs a ciphertext CT.*

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

$\mathsf{Fake}(\mathsf{pk}, \mathsf{sk}, \mathsf{aux}) \to \widetilde{\mathsf{CT}}$**:** *The fake encryption algorithm takes pk, sk, and an auxiliary input 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 $\mathsf{pk}, \mathsf{sk}, \mathsf{aux}$, a fake ciphertext $\widetilde{\mathsf{CT}}$, and a plaintext m, and outputs a fake secret key $\widetilde{\mathsf{sk}}$.*

Correctness is the same as that of PKE.

Definition 2.13 (Receiver Non-Committing (RNC) Security). *An NCE scheme is RNC secure if it satisfies the following. Let $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Reveal})$ be an NCE scheme. We consider the following the security experiment $\mathsf{Exp}^{\mathsf{rec\text{-}nc}}_{\Sigma,\mathcal{A}}(\lambda, b)$.*

1. *The challenger computes $(\mathsf{pk}, \mathsf{sk}, \mathsf{aux}) \leftarrow \mathsf{KeyGen}(1^\lambda)$.*

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 the $b' \in \{0, 1\}$.*

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

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

Theorem 2.14 ([KNTY19]). *If there exists an IND-CPA secure SKE/PKE scheme (against QPT adversaries), there exists an RNC secure secret/public key NCE 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, but it is easy to extend their theorem to the PKE setting. We also note that the core idea of Kitagawa et al. is based on the observation by Canetti, Halevi, and Katz [CHK05].

Non-interactive zero-knowledge. We review non-interactive zero-knowledge (NIZK) which is used in Section 4.2.

Definition 2.15 (Non-Interactive Zero-Knowledge Proofs). *A non-interactive zero-knowledge (NIZK) proof for an **NP** language $\mathcal{L}$ consists of PPT algorithms* $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Vrfy})$.

$\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$*: The setup algorithm takes as input the security parameter 1^λ and outputs a common reference string* crs.

$\mathsf{Prove}(\mathsf{crs}, x, w) \to \pi$*: The prover's algorithm takes as input a common reference string* crs*, a statement x, and a witness w and outputs a proof π.*

$\mathsf{Vrfy}(\mathsf{crs}, x, \pi) \to \top$ *or* $\bot$*: The verifier's algorithm takes as input a common reference string* crs*, a statement x, and a proof π and outputs $\top$ to indicate acceptance of the proof and $\bot$ otherwise.*

A non-interactive proof must satisfy the following requirements.

Completeness: *For all $\lambda \in \mathbb{N}$ and all pairs $(x, w) \in \mathcal{R}$, we have*

$$\Pr[\mathsf{Vrfy}(\mathsf{crs}, x, \pi) = \top \mid \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda), \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w)] = 1.$$

Statistical Soundness. *For all unbounded time adversaries $\mathcal{A}$, if we run $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$, then we have*

$$\Pr[x \notin \mathcal{L} \wedge \mathsf{Vrfy}(\mathsf{crs}, x, \pi) = \top \mid (x, \pi) \leftarrow \mathcal{A}(\mathsf{crs}, \cdot, \cdot)(1^\lambda, \mathsf{crs})] \leq \mathsf{negl}(\lambda).$$

(Computational) Zero-Knowledge: *If there exists a PPT simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$ such that for all QPT adversaries $\mathcal{A}$ and for all $(x, w) \in \mathcal{R}$, we have*

$$\left|\Pr\left[\mathcal{A}(1^\lambda, \mathsf{crs}, x, \pi) = 1 \;\middle|\; \begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda),\\ \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w)\end{array}\right] - \Pr\left[\mathcal{A}(1^\lambda, \widetilde{\mathsf{crs}}, x, \pi) = 1 \;\middle|\; \begin{array}{l}(\widetilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{Sim}_1(1^\lambda, x),\\ \pi \leftarrow \mathsf{Sim}_2(\widetilde{\mathsf{crs}}, \mathsf{td}, x)\end{array}\right]\right| \leq \mathsf{negl}(\lambda).$$

Theorem 2.16 ([PS19]). *If the LWE assumption holds against QPT adversaries, then there exists computational NIZK proof for **NP** against QPT adversaries.*

Definition 2.17 (Statistical Simulation-Sound NIZK). *A NIZK proof system Π_{nizk} is statistical simulation-sound if it is hard to generate a convincing proof for a false statement even if an adversary is given a simulated proof. That is, for all statements x and all unbounded time adversaries $\mathcal{A}$, we have*

$$\Pr[x^* \neq x \wedge x^* \notin \mathcal{L} \wedge \mathsf{Vrfy}(\widetilde{\mathsf{crs}}, x^*, \pi^*) = \top \mid (\widetilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{Sim}_1(1^\lambda, x), \pi \leftarrow \mathsf{Sim}_2(\widetilde{\mathsf{crs}}, \mathsf{td}, x), (x^*, \pi^*) \leftarrow \mathcal{A}(\widetilde{\mathsf{crs}}, x, \pi)] \leq \mathsf{negl}(\lambda).$$

Theorem 2.18 ([GGH+16]). *If there exist computational NIZK proof systems for $\mathcal{L}$ and non-interactive perfectly binding commitment schemes, there exists a statistical simulation-sound NIZK proof system for $\mathcal{L}$.*

Non-interactive perfectly binding commitment can be constructed from IND-CPA PKE with perfect correctness [LS19]. We can obtain such PKE schemes by slightly modifying IND-CPA PKE with imperfect correctness. For example, we can obtain a variant of Regev PKE scheme that has perfect correctness from Regev PKE scheme. Thus, we obtain the following corollary from Theorems 2.16 and 2.18.

Corollary 2.19. *If the LWE assumption holds against QPT adversaries, there exists a statistical simulation-sound NIZK proof system for **NP**.*

Definition 2.20 (Indistinguishability Obfuscator [BGI+12]). *A PPT algorithm $i\mathcal{O}$ is an IO for a circuit class $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}}$ if it satisfies the following two conditions.*

Functionality: *For any security parameter $\lambda \in \mathbb{N}$, circuit $C \in \mathcal{C}_\lambda$, and input x, we have that*

$$\Pr[C'(x) = C(x) \mid C' \leftarrow i\mathcal{O}(C)] = 1 \ .$$

Indistinguishability: *For any QPT distinguisher $\mathcal{D}$ and for any pair of circuits $C_0, C_1 \in \mathcal{C}_\lambda$ such that for any input x, $C_0(x) = C_1(x)$ and $|C_0| = |C_1|$, it holds that*

$$\mathsf{Adv}^{\mathsf{io}}_{i\mathcal{O},\mathcal{A}}(\lambda) := |\Pr[\mathcal{D}(i\mathcal{O}(C_0)) = 1] - \Pr[\mathcal{D}(i\mathcal{O}(C_1)) = 1]| \leq \mathsf{negl}(\lambda) \ .$$

There exist candidate constructions of IO against QPT adversaries [GP20, WW20, BDGM20].

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 (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 certification is generated in the encryption algorithm.

Definition 3.1 (PKE with Certified Deletion (Syntax)). *A certified deletion public-key encryption scheme 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 and outputs a 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 and a plaintext and outputs a verification key* vk *and a ciphertext* CT.

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

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

$\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert}) \to \top$ **or** $\bot$ *The verification algorithm takes the verification key and a certificate 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: *For any* $\lambda \in \mathbb{N}$, $m \in \mathcal{M}$,

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

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

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

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 the 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* $(\mathsf{vk}_b, \mathsf{CT}_b) \leftarrow \mathsf{Enc}(\mathsf{sk}, m_b)$ *and sends* CT_b *to the* $\mathcal{A}$.
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* $\bot$*, then abort. Otherwise (the output is* $\top$*), the challenger sends the* sk *to the* $\mathcal{A}$.
6. $\mathcal{A}$ *receives* sk *and outputs the* $b' \in \{0, 1\}$.

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* ϵ*-IND-CPA-CD secure if for any QPT adversary* $\mathcal{A}$*, it holds that*

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

When ϵ *is negligible, we omit* ϵ *and say* Σ *is IND-CPA-CD secure.*

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 NCE 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 NCE scheme $\Sigma_{\mathsf{nce}} = \mathsf{NCE}.(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Reveal})$ with plaintext space $\mathcal{K}$.

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

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

$\mathsf{Enc}(\mathsf{pk}, m)$**:**

- Parse $\mathsf{pk} = \mathsf{nce.pk}$.
- Generate $\mathsf{ske.sk} \leftarrow \mathsf{SKE.Gen}(1^\lambda)$.

- Compute $\mathsf{nce.CT} \leftarrow \mathsf{NCE.Enc}(\mathsf{nce.pk}, \mathsf{ske.sk})$ and $\mathsf{ske.CT} \leftarrow \mathsf{SKE.Enc}(\mathsf{ske.sk}, m)$.
- Output $\mathsf{CT} := (\mathsf{nce.CT}, \mathsf{ske.CT})$ and $\mathsf{vk} := \mathsf{ske.sk}$.

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

- Parse $\mathsf{sk} = \mathsf{nce.sk}$ and $\mathsf{CT} = (\mathsf{nce.CT}, \mathsf{ske.CT})$.
- Compute $\mathsf{sk}' \leftarrow \mathsf{NCE.Dec}(\mathsf{nce.sk}, \mathsf{nce.CT})$.
- Compute and output $m' \leftarrow \mathsf{SKE.Dec}(\mathsf{sk}', \mathsf{ske.CT})$.

$\mathsf{Del}(\mathsf{CT})$**:**

- Parse $\mathsf{CT} = (\mathsf{nce.CT}, \mathsf{ske.CT})$.
- 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 follows from the correctness of Σ_{nce} and Σ_{skcd}.

Security. We prove the following theorem.

Theorem 3.4. *If Σ_{nce} is RNC secure and Σ_{skcd} is OT-CD secure, Σ_{pkcd} is IND-CPA-CD secure.*

Proof. 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 $\mathsf{nce.CT}^* \leftarrow \mathsf{NCE.Fake}(\mathsf{nce.pk}, \mathsf{nce.sk}, \mathsf{aux})$ and $\mathsf{ske.CT}^* \leftarrow \mathsf{SKE.Enc}(\mathsf{ske.sk}, m_b)$. The target ciphertext is $\mathsf{CT}^* := (\mathsf{nce.CT}^*, \mathsf{ske.CT}^*)$. In addition, we reveal $\widetilde{\mathsf{sk}} \leftarrow \mathsf{Reveal}(\mathsf{nce.pk}, \mathsf{nce.sk}, \mathsf{nce.aux}, \mathsf{nce.CT}^*, \mathsf{ske.sk})$ instead of $\mathsf{nce.sk}$.

Proposition 3.5. *If Σ_{nce} is RNC secure, $|\Pr[\mathsf{Exp}^{\mathsf{pk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{pkcd}},\mathcal{A}}(\lambda, b) = 1] - \Pr[\mathsf{Hyb}(b) = 1]| \leq \mathsf{Adv}^{\mathsf{pk\text{-}nce}}_{\Sigma_{\mathsf{nce}},\mathcal{B}_1}(\lambda)$.*

Proof. We construct an adversary $\mathcal{B}_1$ that breaks the RNC security of Σ_{nce} by using the distinguisher $\mathcal{D}$ for these two games. First, $\mathcal{B}_1$ is given $\mathsf{nce.pk}$. $\mathcal{B}_1$ generates $\mathsf{ske.sk}_b \leftarrow \mathsf{SKE.Gen}(1^\lambda)$ and sends $\mathsf{nce.pk}$ to $\mathcal{D}$. When $\mathcal{D}$ sends (m_0, m_1), $\mathcal{B}_1$ sends $\mathsf{ske.sk}$ to the challenger of NCE, receives $(\mathsf{nce.CT}^*, \widetilde{\mathsf{sk}})$ and generates $\mathsf{ske.CT} \leftarrow \mathsf{SKE.Enc}(\mathsf{ske.sk}, m_b)$. $\mathcal{B}_1$ sends $(\mathsf{nce.CT}^*, \mathsf{ske.CT})$ to $\mathcal{D}$ as the challenge ciphertext. At some point, $\mathcal{D}$ outputs cert. If $\mathsf{SKE.Vrfy}(\mathsf{cert}) = \top$, $\mathcal{B}_1$ sends $\widetilde{\mathsf{sk}}$ to $\mathcal{D}$.

- If $(\mathsf{nce.CT}^*, \widetilde{\mathsf{sk}}) = (\mathsf{NCE.Enc}(\mathsf{nce.pk}, \mathsf{ske.sk}), \mathsf{nce.sk})$, $\mathcal{B}_1$ perfectly simulates $\mathsf{Exp}^{\mathsf{pk\text{-}cert\text{-}del}}_{\Sigma_{\mathsf{pkcd}},\mathcal{A}}(\lambda, b)$.
- If $(\mathsf{nce.CT}^*, \widetilde{\mathsf{sk}}) = (\mathsf{NCE.Fake}(\mathsf{nce.pk}, \mathsf{nce.sk}, \mathsf{nce.aux}), \mathsf{NCE.Reveal}(\mathsf{nce.pk}, \mathsf{nce.sk}, \mathsf{nce.aux}, \mathsf{nce.CT}^*, \mathsf{ske.sk}))$, $\mathcal{B}_1$ perfectly simulates $\mathsf{Hyb}(b)$.

Thus, if $\mathcal{D}$ distinguishes the two games, we can break the RNC security. This completes the proof. □

Proposition 3.6. *If Σ_{skcd} is OT-CD secure, $|\Pr[\mathsf{Hyb}(0) = 1] - \Pr[\mathsf{Hyb}(1) = 1]| \leq \mathsf{Adv}^{\mathsf{sk\text{-}ot\text{-}cd}}_{\Sigma_{\mathsf{skcd}},\mathcal{B}_2}(\lambda)$.*

Proof. We construct an adversary $\mathcal{B}_2$ that breaks the OT-CD security of Σ_{skcd} by using the distinguisher $\mathcal{D}$ for these two games. First, $\mathcal{B}_2$ generates $(\mathsf{nce.pk}, \mathsf{nce.sk}, \mathsf{nce.aux}) \leftarrow \mathsf{NCE.KeyGen}(1^\lambda)$ and sends $\mathsf{nce.pk}$ to $\mathcal{D}$. When $\mathcal{D}$ sends (m_0, m_1), $\mathcal{B}_2$ sends (m_0, m_1) to the challenger of OT-CD SKE, receives $\mathsf{ske.CT}^*$, and generates $\mathsf{nce.}\widetilde{\mathsf{CT}} \leftarrow \mathsf{NCE.Fake}(\mathsf{nce.pk}, \mathsf{nce.sk}, \mathsf{nce.aux})$. $\mathcal{B}_2$ sends $(\mathsf{nce.}\widetilde{\mathsf{CT}}, \mathsf{ske.CT}^*)$ to $\mathcal{D}$ as the challenge ciphertext. At some point, $\mathcal{D}$ outputs cert. $\mathcal{B}_2$ passes cert to the challenger of OT-CD SKE. If the challenger returns $\mathsf{ske.sk}$, $\mathcal{B}_2$ generates $\widetilde{\mathsf{sk}} \leftarrow \mathsf{NCE.Reveal}(\mathsf{nce.pk}, \mathsf{nce.sk}, \mathsf{nce.aux}, \mathsf{nce.}\widetilde{\mathsf{CT}}, \mathsf{ske.sk})$ and sends $\widetilde{\mathsf{sk}}$ to $\mathcal{D}$.

- If $\mathsf{ske.CT}^* = \mathsf{SKE.Enc}(\mathsf{ske.sk}, m_0)$, $\mathcal{B}_2$ perfectly simulates $\mathsf{Hyb}(0)$.
- If $\mathsf{ske.CT}^* = \mathsf{SKE.Enc}(\mathsf{ske.sk}, m_1)$, $\mathcal{B}_2$ perfectly simulates $\mathsf{Hyb}(1)$.

Thus, if $\mathcal{D}$ distinguishes the two games, we can break the OT-CD security. This completes the proof. □

By Propositions 3.5 and 3.6, we immediately obtain Theorem 3.4.

□

By Theorems 2.11, 2.14 and 3.4, we immediately obtain the following corollary.

Corollary 3.7. *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_{nce} with a secret key NCE scheme. We omit the proof since it is almost the same as that of Theorem 3.4. By Theorem 2.14 and the fact that OWFs imply (reusable) SKE [HILL99, GGM86], we also obtain the following theorem.

Theorem 3.8. *If there exists OWF against QPT adversaries, there exists IND-CPA-CD secure SKE with certified deletion.*

4 Attribute-Based Encryption with Certified Deletion

In this section, we define the notion of attribute-based encryption (ABE) with certified deletion, which is a natural extension of ABE and PKE with certified deletion and present how to achieve ABE with certified deletion from OT-CD secure SKE, IO, and OWFs. In Section 4.1, we present the definition of ABE with certified deletion and non-committing ABE (NCABE), which is a crucial tool to achieve ABE with certified deletion. In Section 4.2, we present how to achieve NCABE from IO and standard ABE. In Section 4.3, we present how to achieve ABE with certified deletion from NCABE and OT-CD secure SKE with certified deletion.

4.1 Definition of ABE with Certified Deletion

The definition of ABE with certified deletion is a natural combination of ABE and PKE with certified deletion.

Definition 4.1 (Attribute-Based Encryption with Certified Deletion (Syntax)). *A certified deletion ABE scheme is a tuple of quantum algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ *with plaintext space* $\mathcal{M}$*, attribute space* $\mathcal{X}$*, and policy space* $\mathcal{P}$*.*

$\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{msk})$**:** *The setup algorithm takes as input the security parameter and outputs a public key* pk *and a master secret key* msk*.*

$\mathsf{KeyGen}(\mathsf{msk}, P) \to \mathsf{sk}_P$**:** *The key generation algorithm takes as* msk *and a policy* $P \in \mathcal{P}$*, and outputs a secret key* sk_P*.*

$\mathsf{Enc}(\mathsf{pk}, X, m) \to (\mathsf{vk}, \mathsf{CT}_X)$**:** *The encryption algorithm takes as input* pk*, an attribute* $X \in \mathcal{X}$*, and a plaintext* $m \in \mathcal{M}$*, and outputs a verification key* vk *and ciphertext* CT_X*.*

$\mathsf{Dec}(\mathsf{sk}_P, \mathsf{CT}_X) \to m'$**:** *The decryption algorithm takes as input* sk *and* CT*, and output 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*.*

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

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

Decryption correctness: *For any $\lambda \in \mathbb{N}$, $m \in \mathcal{M}$, $P \in \mathcal{P}$, and $X \in \mathcal{X}$ such that $P(X) = \top$,*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}_P, \mathsf{CT}_X) \neq m \;\middle|\; \begin{array}{l}(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{sk}_P \leftarrow \mathsf{KeyGen}(\mathsf{msk}, P) \\ (\mathsf{vk}, \mathsf{CT}_X) \leftarrow \mathsf{Enc}(\mathsf{pk}, X, \mathsf{msg})\end{array}\right] \leq \mathsf{negl}(\lambda).$$

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

$$\Pr\left[\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert}) = \bot \;\middle|\; \begin{array}{l}(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathsf{vk}, \mathsf{CT}_X) \leftarrow \mathsf{Enc}(\mathsf{pk}, X, \mathsf{msg}) \\ \mathsf{cert} \leftarrow \mathsf{Del}(\mathsf{CT}_X)\end{array}\right] \leq \mathsf{negl}(\lambda).$$

Definition 4.3 (ABE Certified Deletion Security). *Let $\Sigma = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ be an ABE with certified deletion. We consider the following the security experiment $\mathsf{Exp}^{\mathsf{ind\text{-}sel\text{-}cpa\text{-}cd}}_{\Sigma,\mathcal{A}}(\lambda, b)$.*

1. *$\mathcal{A}$ declare the target attribute $X^* \in \mathcal{X}$ and sends it to the challenger.*
2. *The challenger computes $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends pk to $\mathcal{A}$.*
3. *$\mathcal{A}$ sends a key query $P_i \in \mathcal{P}$ to the challenger and it returns $\mathsf{sk}_{P_i} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, P_i)$ to $\mathcal{A}$. This process can be repeated polynomially many times.*
4. *$\mathcal{A}$ sends $(m_0, m_1) \in \mathcal{M}^2$ to the challenger.*
5. *The challenger computes $(\mathsf{vk}_b, \mathsf{CT}_b) \leftarrow \mathsf{Enc}(\mathsf{pk}, X^*, m_b)$ and sends CT_b to the $\mathcal{A}$.*
6. *Again, $\mathcal{A}$ can send key queries.*
7. *$\mathcal{A}$ computes $\mathsf{cert} \leftarrow \mathsf{Del}(\mathsf{CT}_b)$ and sends cert to the challenger.*
8. *The challenger computes $\mathsf{Vrfy}(\mathsf{vk}_b, \mathsf{cert})$. If the output is $\bot$, then abort. Otherwise, the challenger sends msk to the $\mathcal{A}$.*
9. *$\mathcal{A}$ receives msk and outputs the $b' \in \{0, 1\}$.*

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

$$\mathsf{Adv}^{\mathsf{ind\text{-}sel\text{-}cpa\text{-}cd}}_{\Sigma,\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{ind\text{-}sel\text{-}cpa\text{-}del}}_{\Sigma,\mathcal{A}}(\lambda, 0) = 1] - \Pr[\mathsf{Exp}^{\mathsf{ind\text{-}sel\text{-}cpa\text{-}cd}}_{\Sigma,\mathcal{A}}(\lambda, 1) = 1]| \leq \mathsf{negl}(\lambda).$$

Next, we define non-committing ABE, which is a non-committing encryption version of ABE.

Definition 4.4 (Non-Committing Attribute-Based Encryption (Syntax)). *A non-committing (key policy) attributed-based encryption (NCABE) is a tuple of algorithms $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{FakeSetup}, \mathsf{FakeSK}, \mathsf{FakeCT}, \mathsf{Reveal})$ with plaintext space $\mathcal{M}$, attribute space $\mathcal{X}$, and policy space $\mathcal{P}$.*

$\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{msk})$**:** *The setup algorithm takes as input the security parameter and outputs a public key pk and a master secret key msk.*

$\mathsf{KeyGen}(\mathsf{msk}, P) \to \mathsf{sk}_P$**:** *The key generation algorithm takes as input msk and a policy $P \in \mathcal{P}$, and outputs a secret key sk_P.*

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

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

$\mathsf{FakeSetup}(1^\lambda, \mathsf{aux}') \to (\mathsf{pk}, \mathsf{aux})$**:** *The setup algorithm takes as input the security parameter and a pre-auxiliary information* aux' *and outputs a public key* pk*, a post-auxiliary information* aux*.*

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

$\mathsf{FakeSK}(\mathsf{pk}, \mathsf{aux}, P) \to \widetilde{\mathsf{sk}}$**:** *The fake key generation algorithm takes* pk*,* aux*, and* $P \in \mathcal{P}$*, and outputs a fake secret key* $\widetilde{\mathsf{sk}}$*.*

$\mathsf{Reveal}(\mathsf{pk}, \mathsf{aux}, \widetilde{\mathsf{CT}}, m) \to \widetilde{\mathsf{msk}}$**:** *The reveal algorithm takes* $\mathsf{pk}, \mathsf{aux}$*, a fake ciphertext* $\widetilde{\mathsf{CT}}$*, and a plaintext* $m \in \mathcal{M}$*, and outputs a fake master secret key* $\widetilde{\mathsf{msk}}$*.*

Correctness is the same as that of ABE.

Definition 4.5 (RNC Security for ABE). *An NCABE scheme is RNC secure if it satisfies the following. Let* $\Sigma = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Reveal})$ *be an NCE scheme. We consider the following the security experiment* $\mathsf{Exp}^{\mathsf{rnc\text{-}sel\text{-}cpa}}_{\Sigma,\mathcal{A}}(\lambda, b)$*.*

1. $\mathcal{A}$ *declared the target attribute* $X^* \in \mathcal{X}$ *and sends it to the challenger.*
2. *The challenger sets* $\mathsf{aux}' := X^*$*.*
   - *If* $b = 0$*, The challenger computes* $(\mathsf{pk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ *and sends* pk *to* $\mathcal{A}$*.*
   - *If* $b = 1$*, The challenger computes* $(\mathsf{pk}, \mathsf{aux}) \leftarrow \mathsf{FakeSetup}(1^\lambda, \mathsf{aux}')$ *and sends* pk *to* $\mathcal{A}$*.*
3. $\mathcal{A}$ *sends a query* $P_i \in \mathcal{P}$ *to the challenger. If* $P_i(X^*) = \top$*, returns nothing. Else:*
   - *If* $b = 0$*, the challenger returns a secret key* $\mathsf{sk}_{P_i} \leftarrow \mathsf{KeyGen}(\mathsf{msk}, P_i)$*.*
   - *If* $b = 1$*, the challenger returns a fake secret key* $\widetilde{\mathsf{sk}}_{P_i} \leftarrow \mathsf{FakeSK}(\mathsf{pk}, \mathsf{aux}, P_i)$*.*

   $\mathcal{A}$ *can send polynomially many key queries.*
4. *At some point,* $\mathcal{A}$ *sends a query* $m \in \mathcal{M}$ *to the challenger. The challenger does the following.*
   - *If* $b = 0$*, the challenger generates* $\mathsf{CT}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, X^*, m)$ *and returns* $(\mathsf{CT}, \mathsf{msk})$ *to* $\mathcal{A}$*.*
   - *If* $b = 1$*, the challenger generates* $\widetilde{\mathsf{CT}}^* \leftarrow \mathsf{FakeCT}(\mathsf{pk}, \mathsf{aux}, X^*)$ *and* $\widetilde{\mathsf{msk}} \leftarrow \mathsf{Reveal}(\mathsf{pk}, \mathsf{aux}, \widetilde{\mathsf{CT}}, m)$ *and returns* $(\widetilde{\mathsf{CT}}, \widetilde{\mathsf{msk}})$ *to* $\mathcal{A}$*.*
5. *Again* $\mathcal{A}$ *can send key queries.*
6. $\mathcal{A}$ *outputs the* $b' \in \{0, 1\}$*.*

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

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

If $\mathcal{A}$ *also declares the target plaintext* m *at the beginning of the game above, we say that* Σ *is selective-message RNC secure and write its advantages as* $\mathsf{Adv}^{\mathsf{rnc\text{-}sel2\text{-}cpa}}_{\Sigma,\mathcal{A}}(\lambda)$*.*

Note that a selective-message RNC secure scheme is sufficient for our purpose (ABE with certified deletion) since we use the hybrid encryption technique as in Section 3.2.

4.2 Non-Committing ABE from IO

In this section, we construct NCABE scheme with plaintext space $\{0,1\}^{\ell_m}$, attribute space $\mathcal{X} = \{0,1\}^{\ell_x}$ where ℓ_m and ℓ_x are some polynomials, and policy space $\mathcal{P} = \mathsf{P/poly}$ from IO for $\mathsf{P/poly}$ and PKE scheme with plaintext space $\{0,1\}$.

Our NCABE scheme. We present an NCABE scheme based on any IND-CPA-secure PKE scheme and IO. Let $\Sigma_{\mathsf{abe}} = \mathsf{ABE}.(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-sel-CPA-secure ABE scheme on the message space $\{0,1\}$ and Π_{nizk} a statistically simulation-sound NIZK protocol. Let $\mathsf{pk} := \{\mathsf{abe.pk}_{i,0}, \mathsf{abe.pk}_{i,1}\}_{i\in\ell_m}$. We define an NP relation $\mathcal{R}_L$ defined as follows.

$$\mathcal{R}_L := \{((\mathsf{pk}, \{\mathsf{CT}_{i,0}, \mathsf{CT}_{i,1}\}_{i\in[\ell_m]}, X), \{(m[i], r_{i,0}, r_{i,1})\}_{i\in[\ell_m]}) \mid \forall i \forall b\ \mathsf{CT}_{i,b} = \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,b}, X, m[i]; r_{i,b})\}.$$

We construct an NCABE scheme $\Sigma_{\mathsf{nce}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{FakeSetup}, \mathsf{FakeCT}, \mathsf{FakeSK}, \mathsf{Reveal})$ as follows.

$\mathsf{Setup}(1^\lambda, 1^{\ell_m})$:

1. Generate $(\mathsf{abe.pk}_{i,b}, \mathsf{abe.msk}_{i,b}) \leftarrow \mathsf{ABE.Setup}(1^\lambda)$ for every $i \in [\ell_m]$ and $b \in \{0,1\}$.
2. Choose $z \leftarrow \{0,1\}^{\ell_m}$.
3. Computes $\mathsf{crs} \leftarrow \mathsf{NIZK.Setup}(1^\lambda)$.
4. Output $\mathsf{pk} := (\{\mathsf{abe.pk}_{i,b}\}_{i\in[\ell_m],b\in\{0,1\}}, \mathsf{crs})$ and $\mathsf{msk} := (\{\mathsf{abe.msk}_{i,z[i]}\}_{i\in[\ell_m]}, z)$.

$\mathsf{KeyGen}(\mathsf{msk}, P)$**:**

1. Parse $\mathsf{msk} = (\{\mathsf{abe.msk}_i\}_{i\in[\ell_m]}, z)$.
2. Generate $\mathsf{sk}_i \leftarrow \mathsf{ABE.KeyGen}(\mathsf{abe.msk}_i, P)$ for every $i \in [\ell_m]$.
3. Generate and output $\mathsf{sk}_P := i\mathcal{O}(\mathsf{D}[\mathsf{crs}, \{\mathsf{sk}_i\}_i, z])$, where circuit D is described in Figure 1.

$\mathsf{Enc}(\mathsf{pk}, X, m)$:

1. Parse $\mathsf{pk} = (\{\mathsf{abe.pk}_{i,b}\}_{i\in[\ell_m],b\in\{0,1\}}, \mathsf{crs})$.
2. Generate $\mathsf{CT}_{i,b} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,b}, X, m[i])$ for every $i \in [\ell_m]$ and $b \in \{0,1\}$.
3. Generate $\pi \leftarrow \mathsf{NIZK.Prove}(\mathsf{crs}, d, w)$ where $d = (\{(\mathsf{abe.pk}_{i,0}, \mathsf{abe.pk}_{i,1}, \mathsf{CT}_{i,0}, \mathsf{CT}_{i,1})\}_i, X)$ and $w = (m, \{r_{i,0}, r_{i,1}\}_i)$.
4. Output $\mathsf{CT}_X := (\{\mathsf{CT}_{i,0}, \mathsf{CT}_{i,1}\}_{i\in[\ell_m]}, \pi)$.

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

1. Parse $\mathsf{sk}_P = \widetilde{\mathsf{D}}$ and $\mathsf{CT}_X = (\{\mathsf{CT}_{i,0}, \mathsf{CT}_{i,1}\}_{i\in[\ell_m]}, \pi)$.
2. If $\mathsf{NIZK.Vrfy}(\mathsf{crs}, d, \pi) \neq \top$, output $\perp$.
3. Compute and output $m := \widetilde{\mathsf{D}}(\mathsf{CT}_X)$.

$\mathsf{FakeSetup}(1^\lambda, 1^{\ell_m}, \mathsf{aux}')$:

1. Parse $\mathsf{aux}' = X^*$.
2. Generate $(\mathsf{abe.pk}_{i,b}, \mathsf{abe.msk}_{i,b}) \leftarrow \mathsf{ABE.Setup}(1^\lambda)$ for every $i \in [\ell_m]$ and $b \in \{0,1\}$.
3. Choose $z^* \leftarrow \{0,1\}^{\ell_m}$.
4. Compute $\mathsf{CT}^*_{i,z^*[i]} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,z^*[i]}, X^*, 0)$ and $\mathsf{CT}^*_{i,1-z^*[i]} \leftarrow \mathsf{ABE.Enc}(\mathsf{pk}_{i,1-z^*[i]}, X^*, 1)$ for every $i \in [\ell_m]$.
5. Computes $(\widetilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{Sim}_1(1^\lambda, d^*)$ where $d^* = (\{(\mathsf{abe.pk}_{i,0}, \mathsf{abe.pk}_{i,1}, \mathsf{CT}^*_{i,0}, \mathsf{CT}^*_{i,1})\}_i, X^*)$.
6. Output $\mathsf{pk} := (\{\mathsf{abe.pk}_{i,b}\}_{i\in[\ell_m],b\in\{0,1\}}, \widetilde{\mathsf{crs}})$ and $\mathsf{aux} := (\{\mathsf{abe.msk}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}, \{\mathsf{CT}^*_{i,b}\}_{i\in[\ell_m]}, z^*)$.

$\mathsf{FakeSK}(\mathsf{pk}, \mathsf{aux}, P)$:

1. Parse $\mathsf{aux} = (\{\mathsf{abe.msk}_{i,b}\}_{i\in[\ell_m],b\in\{0,1\}}, \{\mathsf{CT}^*_{i,b}\}_{i\in[\ell_m]}, z^*)$.
2. Generate $\mathsf{sk}^0_i \leftarrow \mathsf{ABE.KeyGen}(\mathsf{abe.msk}_{i,0}, P)$ for every $i \in [\ell_m]$ and set $\mathsf{sk}^0_P := \{\mathsf{sk}^0_i\}_i$.

3. Generate and output $\widetilde{\mathsf{sk}} := i\mathcal{O}(\mathsf{D}_0[\widetilde{\mathsf{crs}}, \mathsf{sk}_P^0])$, where circuit D_0 is described in Figure 2.

$\mathsf{FakeCT}(\mathsf{pk}, \mathsf{aux}, X)$:

1. Parse $\mathsf{pk} = (\{\mathsf{abe.pk}_{i,b}\}_{i\in[\ell_m],b\in\{0,1\}}, \widetilde{\mathsf{crs}})$ and $\mathsf{aux} = (\{\mathsf{abe.msk}_{i,b}\}_{i\in[\ell],b\in\{0,1\}}, \{\mathsf{CT}^*_{i,b}\}_{i\in[\ell_m]}, z^*)$.
2. Compute $\widetilde{\pi} \leftarrow \mathsf{Sim}_2(\widetilde{\mathsf{crs}}, \mathsf{td}, d^*)$.
3. Outputs $\widetilde{\mathsf{CT}}_X := (\{\mathsf{CT}^*_{i,b}\}_{i\in[\ell_m],b\in\{0,1\}}, \widetilde{\pi})$.

$\mathsf{Reveal}(\mathsf{pk}, \mathsf{aux}, \widetilde{\mathsf{CT}}_X, m)$:

1. Parses $\mathsf{aux} = (\{\mathsf{abe.msk}_{i,b}\}_{i\in[\ell_m],b\in\{0,1\}}, \{\mathsf{CT}^*_{i,b}\}_{i\in[\ell_m]}, z^*)$.
2. Outputs $\widetilde{\mathsf{msk}} := (\{\mathsf{abe.msk}_{i,z^*[i]\oplus m[i]}\}_{i\in[\ell_m]}, z^* \oplus m)$.

Left-or-Right Decryption Circuit D

Input: A ciphertext CT_X.

Hardwired value: crs, z, and $\{\mathsf{sk}_i\}_i$.

1. Parse $\mathsf{CT}_X = (\{\mathsf{CT}_{i,0}, \mathsf{CT}_{i,1}\}_i, \pi)$
2. If $\mathsf{NIZK.Vrfy}(\mathsf{crs}, d, \pi) \neq \top$, output $\bot$.
3. Compute $m[i] \leftarrow \mathsf{ABE.Dec}(\mathsf{sk}_i, \mathsf{CT}_{i,z[i]})$ for $i \in [\ell_m]$.
4. Output $m := m[1]\| \cdots \|m[\ell_m]$.

Figure 1: The description of the left-or-right decryption circuit

Left Decryption Circuit D_0

Input: A ciphertext CT_X.

Hardwired value: $\widetilde{\mathsf{crs}}$ and $\mathsf{sk}_P^0 = \{\mathsf{sk}_i^0\}_i$.

1. Parse $\mathsf{CT}_X = (\{\mathsf{CT}_{i,0}, \mathsf{CT}_{i,1}\}_i, \pi)$
2. If $\mathsf{NIZK.Vrfy}(\mathsf{crs}, d, \pi) \neq \top$, output $\bot$.
3. Compute $m[i] \leftarrow \mathsf{ABE.Dec}(\mathsf{sk}_i^0, \mathsf{CT}_{i,0})$ for $i \in [\ell_m]$.
4. Output $m := m[1]\| \cdots \|m[\ell_m]$.

Figure 2: The description of the left decryption circuit

Correctness. Correctness of Σ_{nce} easily follows from correctness of Σ_{abe} and completeness of Π_{nizk}.

Security. We prove the following theorem.

Theorem 4.6. *If Σ_{abe} is IND-sel-CPA and $i\mathcal{O}$ is secure IO for $\mathsf{P/poly}$, and Π_{nizk} is an SSS-NIZK proof system for **NP**, Σ_{nce} is selective-message RNC secure NCABE.*

Proof. We define a sequence of hybrid games.

- Hyb_0: This is the same as $\mathsf{Exp}^{\mathsf{rnc\text{-}sel\text{-}cpa}}_{\Sigma,\mathcal{A}}(\lambda, 0)$. Let X^*, m^*, and $d^* = (\{(\mathsf{abe.pk}_{i,0}, \mathsf{abe.pk}_{i,1}, \mathsf{CT}^*_{i,0}, \mathsf{CT}^*_{i,1})\}_i, X^*)$ be the target attribute, target message, and statement of NIZK used in the target ciphertext, respectively.

- Hyb_1: This is the same as Hyb_0 except that the challenger generates a simulated NIZK proof for the target ciphertext instead of real one. That is, it generates $\widetilde{\pi} \leftarrow \mathsf{Sim}_2(\widetilde{\mathsf{crs}}, \mathsf{td}, d^*)$ where $(\widetilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{Sim}_1(1^\lambda, d^*)$, $d^* = (\{\mathsf{abe.pk}_{i,0}, \mathsf{abe.pk}_{i,1}, \mathsf{CT}^*_{i,0}, \mathsf{CT}^*_{i,1}\}_i, X^*)$, and X^* is the target attribute that the adversary declares at the beginning of the game. In addition, it uses $\widetilde{\mathsf{crs}}$ instead of crs as a part of pk. This change is indistinguishable by the computational ZK property of Π_{nizk}.
- Hyb_2: This is the same as Hyb_1 except that the challenger uses circuit $\mathsf{D}_0[\widetilde{\mathsf{crs}}, \mathsf{sk}^0_P]$ instead of $\mathsf{D}[\mathsf{crs}, \{\mathsf{sk}_i\}_i, z]$ to generate secret keys for key queries. That is, it returns $\widetilde{\mathsf{sk}} = i\mathcal{O}(\mathsf{D}_0[\widetilde{\mathsf{crs}}, \mathsf{sk}^0_P])$ instead of $\mathsf{sk} = i\mathcal{O}(\mathsf{D}[\mathsf{crs}, \{\mathsf{sk}_i\}_i, z])$. This change is indistinguishable by the IO security and statistical simulation-soundness of Π_{nizk}. Note that secret keys do not depend on z in this game. Let q be the total number of key queries.
- Hyb_3: This is the same as Hyb_2 except that the challenger generates an inconsistent target ciphertext. That is, it generates $\mathsf{CT}^*_{i,1-z[i]} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,1-z[i]}, X^*, 1 - m^*[i])$ and $\mathsf{CT}^*_{i,z[i]} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,z[i]}, X^*, m^*[i])$ instead of double encryption of $m[i]$ for all i. Note that the NIZK proof in the target ciphertext is simulated in this game.
- Hyb_4: This is the same as Hyb_3 except that the challenger chooses $z^* \leftarrow \{0,1\}^{\ell_m}$, computes $\mathsf{CT}^*_{i,z[i]^*} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,z^*[i]}, X^*, 0)$ and $\mathsf{CT}^*_{i,1-z^*[i]} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,1-z^*[i]}, X^*, 1)$, and sets $\widetilde{\mathsf{msk}} := (z^* \oplus m^*, \{\mathsf{msk}_{i,z[i]^* \oplus m^*[i]}\}_{i \in [\ell_m]})$ as a master secret key.

We prove Propositions 4.7 to 4.10.

Proposition 4.7. $|\Pr[\mathsf{Hyb}_0 = 1] - \Pr[\mathsf{Hyb}_1 = 1]| \leq \mathsf{Adv}^{\mathsf{c\text{-}zk}}_{\mathcal{B}_1, \Pi_{\mathsf{nizk}}}(\lambda)$.

Proof of Proposition 4.7. The only difference of these two games is the NIZK proof in the target ciphertext. The distinguisher declares the target attribute X^* and message m^* at the beginning of the game since we consider the selective setting. We can construct an adversary $\mathcal{B}_1$ for computational ZK as follows by using the distinguisher $\mathcal{D}$. $\mathcal{B}_1$ generates $(\mathsf{abe.pk}_{i,b}, \mathsf{abe.msk}_{i,b}) \leftarrow \mathsf{ABE.Setup}(1^\lambda)$, $\mathsf{CT}^*_{i,b} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,b}, X^*, m^*[i]; r_{i,b})$ for $i \in [\ell_m]$ and $b \in \{0,1\}$. $\mathcal{B}_1$ sets $d^* := (\{\mathsf{abe.pk}_{i,0}, \mathsf{abe.pk}_{i,1}, \mathsf{CT}^*_{i,0}, \mathsf{CT}^*_{i,1}\}_i, X^*)$, sends (d^*, w^*) to the challenger where $w^* = (m^*, \{r_{i,0}, r_{i,1}\}_i)$ is the witness for d^* and receives (crs^*, π^*). $\mathcal{B}_1$ sets $\mathsf{pk} := (\{\mathsf{abe.pk}_{i,b}\}_{i,b}, \mathsf{crs}^*)$ and $\mathsf{CT}^* := (\{\mathsf{CT}^*_{i,b}\}_{i,b}, \pi^*)$. $\mathcal{B}_1$ passes pk to $\mathcal{D}$ and simulate secret keys by using $\{\mathsf{msk}_{i,b}\}_{i,b}$.

- If (crs^*, π^*) consists of $\mathsf{crs}^* \leftarrow \mathsf{NIZK.Setup}(1^\lambda)$ and $\pi^* \leftarrow \mathsf{NIZK.Prove}(\mathsf{crs}, d^*, w^*)$, $\mathcal{B}_1$ perfectly simulates Hyb_0
- If (crs^*, π^*) consists of $(\mathsf{crs}^*, \mathsf{td}) \leftarrow \mathsf{Sim}_1(1^\lambda, d^*)$ and $\pi^* \leftarrow \mathsf{Sim}_2(\mathsf{crs}^*, \mathsf{td}, d^*)$, $\mathcal{B}_1$ perfectly simulates Hyb_1

Thus, if $\mathcal{D}$ distinguishes these two games, $\mathcal{B}_1$ breaks the computational ZK property of Π_{nizk}. We complete the proof. □

Proposition 4.8. *If* Π_{nizk} *is statistically simulation-sound,* $|\Pr[\mathsf{Hyb}_1 = 1] - \Pr[\mathsf{Hyb}_2 = 1]| \leq q \cdot \mathsf{Adv}^{\mathsf{io}}_{\mathcal{B}_2, i\mathcal{O}}(\lambda) + \mathsf{negl}(\lambda)$.

Proof of Proposition 4.8. We define more hybrid games. Recall q is the total number of key queries.

Hyb^j_1**:** This is the same as Hyb_1 except that

- for $j < k \leq q$, the challenger generates $\mathsf{sk} = i\mathcal{O}(\mathsf{D}[\mathsf{crs}, \{\mathsf{sk}_i\}_i, z])$ for the k-th key query.
- for $1 \leq k \leq j$, the challenger generates $\widetilde{\mathsf{sk}} = i\mathcal{O}(\mathsf{D}_0[\widetilde{\mathsf{crs}}, \mathsf{sk}^0_P])$ for the k-th key query.

Clearly, $\mathsf{Hyb}^0_1 = \mathsf{Hyb}_1$ and $\mathsf{Hyb}^q_1 = \mathsf{Hyb}_2$.

Let $\mathsf{Invalid}$ be an event that the adversary generates $\mathsf{CT}^\dagger_{i,0} = \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,0}, X^\dagger, m_0[i])$, $\mathsf{CT}^\dagger_{i,1} = \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,1}, X^\dagger, m_1[i])$, and $\pi^\dagger$ such that $\mathsf{NIZK.Vrfy}(\widetilde{\mathsf{crs}}, d^\dagger, \pi^\dagger) = \top$ where $d^\dagger = (\{\mathsf{abe.pk}_{i,0}, \mathsf{abe.pk}_{i,1}, \mathsf{CT}^\dagger_{i,0}, \mathsf{CT}^\dagger_{i,1}\}_i, X^\dagger)$, $m_0[i] \neq m_1[i]$ for some $i \in [\ell_m]$, and $(\widetilde{\mathsf{crs}}, \mathsf{td}) \leftarrow \mathsf{Sim}_1(1^\lambda, d^*)$ such that $d^* \neq d^\dagger$. By the definition of D and D_0, their functionalities are equivalent for all inputs as long as $\mathsf{Invalid}$ does not happen. By the statistical simulation-soundness, there is no false statement $d^\dagger$ such that $d^\dagger \neq d^*$ and a proof for $d^\dagger$ can pass the verification except negligible probability.

Note that if $d^\dagger = d^*$, $X^\dagger = X^*$ and the decryption algorithm must output $\bot$ since $P(X^*) = \bot$ by the admissible property of the ABE security. This means both D and D_0 output $\bot$ for the target ciphertext. That is, $\mathsf{Invalid}$ happens with negligible probability by the statistical simulation-soundness. Thus, D and D_0 are functionally equivalent except negligible probability.

The difference between Hyb_1^{j-1} and Hyb_1^j is that the j-th key query answer is generated by D_0 instead of D. These are indistinguishable by the IO security. That is, $|\Pr[\mathsf{Hyb}_i^{j-1} = 1] - \Pr[\mathsf{Hyb}_i^j = 1]| \leq \mathsf{Adv}^{\mathsf{io}}_{\mathcal{B}_3, i\mathcal{O}}(\lambda) + \mathsf{negl}(\lambda)$. □

Proposition 4.9. $|\Pr[\mathsf{Hyb}_2 = 1] - \Pr[\mathsf{Hyb}_3 = 1]| \leq \ell_m \cdot \mathsf{Adv}^{\mathsf{ind\text{-}sel\text{-}cpa}}_{\mathcal{B}_3, \Sigma_{\mathsf{abe}}}(\lambda)$.

Proof of Proposition 4.9. We define more hybrid games. Recall ℓ_m is the length of plaintexts.

Hyb_2^j**:** This is the same as Hyb_2 except that

- for $j < i \leq \ell_m$, the challenger generates $\mathsf{CT}^*_{i,b} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,b}, X^*, m[i])$ for $b \in \{0,1\}$.
- for $1 \leq i \leq j$, the challenger generates $\mathsf{CT}^*_{i,1-z[i]} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,1-z[i]}, X^*, 1-m[i])$ and $\mathsf{CT}^*_{i,z[i]} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{i,z[i]}, m[i])$.

Clearly, $\mathsf{Hyb}_2^0 = \mathsf{Hyb}_2$ and $\mathsf{Hyb}_2^{\ell_m} = \mathsf{Hyb}_3$.

The difference between Hyb_2^j and Hyb_2^{j-1} is the j-th component of the target ciphertext is valid or invalid. However, in these two games, $\mathsf{msk} \coloneqq (\{\mathsf{abe.msk}_{i,z[i]}\}_{i\in[\ell_m]}, z)$. That is, $\{\mathsf{msk}_{j,1-z[j]}\}_j$ is never revealed to the adversary.

We can construct an adversary $\mathcal{B}_3$ that breaks IND-sel-CPA security of Σ_{abe} under key $\mathsf{abe.pk}_{j,1-z[j]}$ by using the distinguisher $\mathcal{D}$ of these two games. $\mathcal{D}$ first declares the target attribute X^* and target message m^*. $\mathcal{B}_3$ passes X^* to the challenger, receives $\mathsf{abe.pk}$, and sets $\mathsf{abe.pk}_{j,1-z[j]} \coloneqq \mathsf{abe.pk}$. For other public keys (that is, $\{\mathsf{abe.pk}_{i,b}\}_{i,b} \setminus \{\mathsf{abe.pk}_{j,1-z[j]}\}$), $\mathcal{B}_3$ generates them by itself. $\mathcal{B}_3$ sends $\{\mathsf{abe.pk}_{i,b}\}_{i,b}$ to the distinguisher. When the distinguisher sends a key query P, $\mathcal{B}_3$ passes P to the challenger and receives $\mathsf{sk}_{j,1-z[j]} \leftarrow \mathsf{ABE.KeyGen}(\mathsf{msk}_{j,1-z[j]}, P)$. For other secret keys for P (that is, $\{\mathsf{sk}_{i,b} \leftarrow \mathsf{ABE.KeyGen}(\mathsf{msk}_{i,b}, P)\}_{i,b} \setminus \{\mathsf{sk}_{j,1-z[j]}\}$), $\mathcal{B}_3$ generates them by itself since it has $\{\mathsf{msk}_{i,b}\}_{i,b}$ except $\mathsf{msk}_{j,1-z[j]}$. Thus, $\mathcal{B}_3$ can compute $\widetilde{\mathsf{sk}} = i\mathcal{O}(\mathsf{D}_0[\widetilde{\mathsf{crs}}, \mathsf{sk}^0_P])$.

At some point, $\mathcal{B}_3$ sends $(m^*[j], 1-m^*[j])$ to the challenger and receives $\mathsf{CT}^*_{j,1-z[j]}$. For $(i,b) \in [\ell_m] \times \{0,1\} \setminus (j, 1-z[j])$, $\mathcal{B}_3$ generates $\mathsf{CT}^*_{j,z[j]} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{j,z[j]}, X^*, m^*[j])$ and $\{\mathsf{CT}_{i,b}\}_{i\in[\ell_m]\setminus\{j\},b\in\{0,1\}}$ as in Hyb_2^j and Hyb_2^{j-1}. Note that the difference between two games is the j-th component (and in particular $(j, 1-z[j])$ part) of the target ciphertext. Again, $\mathcal{B}_3$ simulates answers for secret key queries as above. $\mathcal{B}_3$ outputs whatever $\mathcal{D}$ outputs.

- If $\mathsf{CT}^*_{j,1-z[j]} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{j,1-z[j]}, X^*, m^*[j])$, $\mathcal{B}_3$ perfectly simulates Hyb_i^{j-1}.
- If $\mathsf{CT}^*_{j,1-z[j]} \leftarrow \mathsf{ABE.Enc}(\mathsf{abe.pk}_{j,1-z[j]}, X^*, 1-m^*[j])$, $\mathcal{B}_3$ perfectly simulates Hyb_i^j.

Thus, if $\mathcal{D}$ distinguishes these two games, $\mathcal{B}_3$ breaks IND-sel-CPA security of Σ_{abe}. This completes the proof. □

Proposition 4.10. $\mathsf{Hyb}_3 = \mathsf{Hyb}_4$.

Proof of Proposition 4.10. This is a conceptual change. The advantage of distinguishing these two games is 0 since we can see that these two games are identical if we set $z \coloneqq z^* \oplus m^*$. Note that secret keys do not depend on z in these games. □

Clearly, $\mathsf{Hyb}_4 = \mathsf{Exp}^{\mathsf{rnc\text{-}sel\text{-}cpa}}_{\Sigma,\mathcal{A}}(\lambda, 1)$. Therefore, we complete the proof by Propositions 4.7 to 4.10. □

4.3 ABE with Certified Deletion from NCABE and SKE with Certified Deletion

In this section, we construct ABE with certified deletion from NCABE and OT-CD secure SKE with certified deletion.

Our ABE with certified deletion scheme. We construct an ABE with certified deletion scheme $\Sigma_{\mathsf{cd}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Del}, \mathsf{Vrfy})$ with plaintext space $\mathcal{M}$, attribute space $\mathcal{X}$, policy space $\mathcal{P}$ from an NCABE scheme $\Sigma_{\mathsf{nce}} = \mathsf{NCE}.(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{FakeSetup}, \mathsf{FakeSK}, \mathsf{FakeCT}, \mathsf{Reveal})$ with plaintext space $\{0,1\}^\ell$, attribute space $\mathcal{X}$, policy space $\mathcal{P}$ and 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 $\{0,1\}^\ell$.

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

- Generate $(\mathsf{nce.pk}, \mathsf{nce.msk}, \mathsf{nce.aux}) \leftarrow \mathsf{NCE.Setup}(1^\lambda)$.
- Output $(\mathsf{pk}, \mathsf{msk}) \coloneqq (\mathsf{nce.pk}, \mathsf{nce.msk})$.

$\mathsf{KeyGen}(\mathsf{msk}, P)$**:**

- Generate $\mathsf{nce.sk}_P \leftarrow \mathsf{NCE.KeyGen}(\mathsf{nce.msk}, P)$ and output $\mathsf{sk}_P \coloneqq \mathsf{nce.sk}_P$.

$\mathsf{Enc}(\mathsf{pk}, X, m)$**:**

- Parse $\mathsf{pk} = \mathsf{nce.pk}$.
- Generate $\mathsf{ske.sk} \leftarrow \mathsf{SKE.Gen}(1^\lambda)$.
- Compute $\mathsf{nce.CT}_X \leftarrow \mathsf{NCE.Enc}(\mathsf{nce.pk}, X, \mathsf{ske.sk})$ and $\mathsf{ske.CT} \leftarrow \mathsf{SKE.Enc}(\mathsf{ske.sk}, m)$.
- Output $\mathsf{CT}_X \coloneqq (\mathsf{nce.CT}_X, \mathsf{ske.CT})$ and $\mathsf{vk} \coloneqq \mathsf{ske.sk}$.

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

- Parse $\mathsf{sk}_P = \mathsf{nce.sk}_P$ and $\mathsf{CT}_X = (\mathsf{nce.CT}_X, \mathsf{ske.CT})$.
- Compute $\mathsf{sk}' \leftarrow \mathsf{NCE.Dec}(\mathsf{nce.sk}_P, \mathsf{nce.CT}_X)$.
- Compute and output $m' \leftarrow \mathsf{SKE.Dec}(\mathsf{sk}', \mathsf{ske.CT})$.

$\mathsf{Del}(\mathsf{CT})$**:**

- Parse $\mathsf{CT}_X = (\mathsf{nce.CT}_X, \mathsf{ske.CT})$.
- Generate $\mathsf{ske.cert} \leftarrow \mathsf{SKE.Del}(\mathsf{ske.CT})$.
- Output $\mathsf{cert} \coloneqq \mathsf{ske.cert}$.

$\mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert})$**:**

- 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. Correctness easily follows from correctness of Σ_{skcd} and Σ_{nce}.

Theorem 4.11. *If Σ_{nce} is selective-message RNC secure NCABE and Σ_{skcd} is OT-CD secure, Σ_{cd} is IND-sel-CPA-CD secure ABE.*

Proof. We define the following hybrid game $\mathsf{Hyb}(b)$.

$\mathsf{Hyb}(b)$**:** This is the same as $\mathsf{Exp}^{\mathsf{ind\text{-}sel\text{-}cpa\text{-}cd}}_{\Sigma_{\mathsf{cd}},\mathcal{A}}(\lambda, b)$ except that following two differences: (1) the challenger generates the target ciphertext as follows. It generates $\mathsf{ske.sk} \leftarrow \mathsf{SKE.Gen}(1^\lambda)$ and $\mathsf{nce.aux} \leftarrow \mathsf{FakeSetup}(1^\lambda, X^*)$, and computes $\mathsf{nce.CT}^*_X \leftarrow \mathsf{NCE.Fake}(\mathsf{nce.pk}, \mathsf{nce.aux}, X^*)$ and $\mathsf{ske.CT}^* \leftarrow \mathsf{SKE.Enc}(\mathsf{ske.sk}, m_b)$. The target ciphertext is $\mathsf{CT}^*_X \coloneqq (\mathsf{nce.CT}^*, \mathsf{ske.CT}^*)$. (2) the challenger generates secret keys as follows. It generates $\widetilde{\mathsf{nce.sk}}_P \leftarrow \mathsf{NCE.FakeSK}(\mathsf{nce.pk}, \mathsf{nce.aux}, P)$ and returns it. (3) The challenger reveals $\widetilde{\mathsf{msk}} \leftarrow \mathsf{Reveal}(\mathsf{nce.pk}, \mathsf{nce.aux}, \mathsf{nce.CT}^*_X, \mathsf{ske.sk})$ instead of $\mathsf{nce.msk}$.

Proposition 4.12. $|\Pr[\mathsf{Exp}^{\mathsf{ind\text{-}sel\text{-}cpa\text{-}cd}}_{\Sigma_{\mathsf{cd}},\mathcal{A}}(\lambda, b) = 1] - \Pr[\mathsf{Hyb}(b) = 1]| \leq \mathsf{Adv}^{\mathsf{rnc\text{-}sel2\text{-}cpa}}_{\Sigma_{\mathsf{nce}},\mathcal{B}_1}(\lambda)$.

Proof. We construct an adversary $\mathcal{B}_1$ that breaks the selective-message RNC security of Σ_{nce} by using the distinguisher $\mathcal{D}$ for these two games. First, the distinguisher declares the target attribute X^*. $\mathcal{B}_1$ generates $\mathsf{ske.sk} \leftarrow \mathsf{SKE.Gen}(1^\lambda)$ and passes $(X^*, \mathsf{ske.sk})$ as the target attribute and message to the challenger, receives $\mathsf{nce.pk}$, and passes $\mathsf{nce.pk}$ to $\mathcal{D}$.

When $\mathcal{D}$ sends a key query P, $\mathcal{B}_1$ returns $\bot$ if $P(X^*) = \bot$, otherwise it passes P to the challenger, receives sk_P, and passes it to $\mathcal{D}$. When the challenger sends $(\mathsf{nce.CT}^*_X, \mathsf{nce.msk}^*)$ to $\mathcal{B}_1$, $\mathcal{B}_1$ generates $\mathsf{ske.CT} \leftarrow \mathsf{SKE.Enc}(\mathsf{ske.sk}, m_b)$ and sends $(\mathsf{nce.CT}^*_X, \mathsf{ske.CT})$ to $\mathcal{D}$ as the challenge ciphertext. At some point, $\mathcal{D}$ outputs cert. If $\mathsf{SKE.Vrfy}(\mathsf{cert}) = \top$, $\mathcal{B}_1$ sends msk^* to $\mathcal{D}$.

- If $(\mathsf{nce.CT}^*_X, \mathsf{msk}^*) = (\mathsf{NCE.Enc}(\mathsf{nce.pk}, X^*, \mathsf{ske.sk}), \mathsf{nce.msk})$ and $\mathsf{sk}_P = \mathsf{NCE.KeyGen}(\mathsf{nce.msk}, P)$, $\mathcal{B}_1$ perfectly simulates $\mathsf{Exp}^{\mathsf{ind\text{-}sel\text{-}cpa\text{-}cd}}_{\Sigma_{\mathsf{cd}},\mathcal{A}}(\lambda, b)$.
- If $(\mathsf{nce.CT}^*_X, \mathsf{msk}^*) = (\mathsf{NCE.FakeCT}(\mathsf{nce.pk}, \mathsf{nce.aux}, X^*), \mathsf{NCE.Reveal}(\mathsf{nce.pk}, \mathsf{nce.aux}, \mathsf{nce.CT}^*_X, \mathsf{ske.sk}))$ and $\mathsf{sk}_P = \mathsf{NCE.FakeSK}(\mathsf{nce.pk}, \mathsf{nce.aux}, P)$, $\mathcal{B}_1$ perfectly simulates $\mathsf{Hyb}(b)$.

Thus, if $\mathcal{D}$ distinguishes the two games, $\mathcal{B}_1$ breaks the selective-message RNC security. This completes the proof. □

Proposition 4.13. $|\Pr[\mathsf{Hyb}(0) = 1] - \Pr[\mathsf{Hyb}(1) = 1]| \leq \mathsf{Adv}^{\mathsf{sk\text{-}ot\text{-}cd}}_{\Sigma_{\mathsf{skcd}},\mathcal{B}_2}(\lambda)$.

Proof. We construct an adversary $\mathcal{B}_2$ that breaks the OT-CD security of Σ_{skcd} by using the distinguisher $\mathcal{D}$ for these two games. First, $\mathcal{D}$ declares the target attribute X^*. $\mathcal{B}_2$ generates $(\mathsf{nce.pk}, \mathsf{nce.aux}) \leftarrow \mathsf{NCE.Setup}(1^\lambda, X^*)$ and sends $\mathsf{nce.pk}$ to $\mathcal{D}$. When $\mathcal{D}$ sends a key query P, $\mathcal{B}_2$ generates $\widetilde{\mathsf{sk}}_P \leftarrow \mathsf{NCE.FakeSK}(\mathsf{nce.pk}, \mathsf{nce.aux}, P)$ and returns it to $\mathcal{D}$. When $\mathcal{D}$ sends (m_0, m_1), $\mathcal{B}_2$ sends (m_0, m_1) to the challenger of OT-CD SKE, receives $\mathsf{ske.CT}^*$ and generates $\mathsf{nce.}\widetilde{\mathsf{CT}}_X \leftarrow \mathsf{NCE.FakeCT}(\mathsf{nce.pk}, \mathsf{nce.aux}, X^*)$. $\mathcal{B}_2$ sends $(\mathsf{nce.}\widetilde{\mathsf{CT}}_X, \mathsf{ske.CT}^*)$ to $\mathcal{D}$ as the challenge ciphertext. At some point, $\mathcal{D}$ outputs cert. $\mathcal{B}_2$ passes cert to the challenger of OT-CD SKE. If the challenger returns $\mathsf{ske.sk}$, $\mathcal{B}_2$ generates $\widetilde{\mathsf{msk}} \leftarrow \mathsf{NCE.Reveal}(\mathsf{nce.pk}, \mathsf{nce.msk}, \mathsf{nce.aux}, \mathsf{nce.}\widetilde{\mathsf{CT}}_X, \mathsf{ske.sk})$ and sends $\widetilde{\mathsf{msk}}$ to $\mathcal{D}$.

- If $\mathsf{ske.CT}^* = \mathsf{SKE.Enc}(\mathsf{ske.sk}, m_0)$, $\mathcal{B}_2$ perfectly simulates $\mathsf{Hyb}(0)$.
- If $\mathsf{ske.CT}^* = \mathsf{SKE.Enc}(\mathsf{ske.sk}, m_1)$, $\mathcal{B}_2$ perfectly simulates $\mathsf{Hyb}(1)$.

Thus, if $\mathcal{D}$ distinguishes the two games, we can break the OT-CD security. This completes the proof. □

By Propositions 4.12 and 4.13, we immediately obtain Theorem 3.4. □

Summary of this section. Since IO and OWFs imply computational NIZK proof for **NP** [BP15] and IND-sel-CPA secure ABE with $\mathcal{P} = \mathsf{P/poly}$, we immediately obtain the following corollary by using Theorems 2.11, 2.18, 4.6 and 4.11.

Corollary 4.14. *If there exist secure IO for* $\mathsf{P/poly}$ *against QPT adversaries, there exists ABE with certified deletion with policy space* $\mathsf{P/poly}$.

Note that we can instantiate all building blocks except NCABE without IO (that is, can construct from the LWE assumption against QPT adversaries).

References

[BB84] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *IEEE International Conference on Computers Systems and Signal Processing*, pages 175–179. IEEE, 1984. (Cited on page 1, 2.)

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011. (Cited on page 2.)

[BDGM20] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. Factoring and pairings are not necessary for iO: Circular-secure LWE suffices. Cryptology ePrint Archive, Report 2020/1024, 2020. https://eprint.iacr.org/2020/1024. (Cited on page 7.)

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2):6:1–6:48, 2012. (Cited on page 1, 7.)

[BI20] Anne Broadbent and Rabib Islam. Quantum encryption with certified deletion. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 92–122. Springer, Heidelberg, November 2020. (Cited on page 1, 2, 4, 5.)

[BL20] Anne Broadbent and Sébastien Lord. Uncloneable quantum encryption via oracles. In Steven T. Flammia, editor, *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. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. (Cited on page 2.)

[BP15] Nir Bitansky and Omer Paneth. ZAPs and non-interactive witness indistinguishability from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 401–427. Springer, Heidelberg, March 2015. (Cited on page 18.)

[BR97] Mihir Bellare and Phillip Rogaway. Collision-resistant hashing: Towards making UOWHFs practical. In Burton S. Kaliski Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 470–484. Springer, Heidelberg, August 1997. (Cited on page 2.)

[CFGN96] Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *28th ACM STOC*, pages 639–648. ACM Press, May 1996. (Cited on page 1, 5.)

[CHK05] Ran Canetti, Shai Halevi, and Jonathan Katz. Adaptively-secure, non-interactive public-key encryption. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 150–168. Springer, Heidelberg, February 2005. (Cited on page 1, 5, 6.)

[CMP20] Andrea Coladangelo, Christian Majenz, and Alexander Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. *arXiv*, 2009.13865, 2020. (Cited on page 2.)

[CRW19] Xavier Coiteux-Roy and Stefan Wolf. Proving erasure. *2019 IEEE International Symposium on Information Theory (ISIT)*, Jul 2019. (Cited on page 2.)

[FM18] Honghao Fu and Carl A. Miller. Local randomness: Examples and application. *Physical Review A*, 97(3), Mar 2018. (Cited on page 2.)

[GGH+16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *SIAM J. Comput.*, 45(3):882–929, 2016. (Cited on page 7.)

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986. (Cited on page 10.)

[GP20] Romain Gay and Rafael Pass. Indistinguishability obfuscation from circular security. Cryptology ePrint Archive, Report 2020/1010, 2020. https://eprint.iacr.org/2020/1010. (Cited on page 7.)

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 2006*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309. (Cited on page 1, 3.)

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008. (Cited on page 3.)

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. *J. ACM*, 62(6):45:1–45:33, 2015. (Cited on page 4.)

[GZ20] Marios Georgiou and Mark Zhandry. Unclonable decryption keys. Cryptology ePrint Archive, Report 2020/877, 2020. https://eprint.iacr.org/2020/877. (Cited on page 2.)

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. (Cited on page 10.)

[JL00] Stanislaw Jarecki and Anna Lysyanskaya. Adaptively secure threshold cryptography: Introducing concurrency, removing erasures. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 221–242. Springer, Heidelberg, May 2000. (Cited on page 1, 5.)

[KNTY19] Fuyuki Kitagawa, Ryo Nishimaki, Keisuke Tanaka, and Takashi Yamakawa. Adaptively secure and succinct functional encryption: Improving security and efficiency, simultaneously. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 521–551. Springer, Heidelberg, August 2019. (Cited on page 1, 6.)

[KT20] Srijita Kundu and Ernest Tan. Composably secure device-independent encryption with certified deletion. *arXiv*, 2011.12704, 2020. (Cited on page 2.)

[LS19] Alex Lombardi and Luke Schaeffer. A note on key agreement and non-interactive commitments. Cryptology ePrint Archive, Report 2019/279, 2019. https://eprint.iacr.org/2019/279. (Cited on page 7.)

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 89–114. Springer, Heidelberg, August 2019. (Cited on page 6.)

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009. (Cited on page 3.)

[SW05] Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005. (Cited on page 1, 3.)

[Unr15] Dominique Unruh. Revocable quantum timed-release encryption. *J. ACM*, 62(6):49:1–49:76, 2015. (Cited on page 2.)

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983. (Cited on page 1, 2.)

[WW20] Hoeteck Wee and Daniel Wichs. Candidate obfuscation via oblivious LWE sampling. Cryptology ePrint Archive, Report 2020/1042, 2020. https://eprint.iacr.org/2020/1042. (Cited on page 7.)