Receiver Selective Opening CCA Secure Public Key Encryption from Various Assumptions

Yi Lu[1,2](✉), Keisuke Hara[1,2], and Keisuke Tanaka[1]

[1] Tokyo Institute of Technology, Tokyo, Japan
{lu.y.ai,hara.k.am}@m.titech.ac.jp, keisuke@is.titech.ac.jp

[2] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

Abstract. Receiver selective opening (RSO) attacks for public key encryption (PKE) capture a situation where one sender sends messages to multiple receivers, and an adversary can corrupt a set of receivers and get their messages and secret keys. Security against RSO attack for a PKE scheme ensures confidentiality of other uncorrupted receivers' ciphertexts. Among all of the RSO security notions, simulation-based RSO security against chosen ciphertext attack (SIM-RSO-CCA security) is the strongest notion. In this paper, we explore constructions of SIM-RSO-CCA secure PKE from various computational assumptions. Toward this goal, we show that a SIM-RSO-CCA secure PKE scheme can be constructed based on an IND-CPA secure PKE scheme and a designated-verifier non-interactive zero-knowledge (DV-NIZK) argument satisfying one-time simulation soundness. Moreover, we give the first construction of DV-NIZK argument satisfying one-time simulation soundness. Consequently, through our generic construction, we obtain the first SIM-RSO-CCA secure PKE scheme under the computational Diffie-Hellman (CDH) or learning parity with noise (LPN) assumption.

Keywords: Public-key encryption · Receiver selective opening security · Chosen ciphertext attacks

1 Introduction

1.1 Background and Motivation

In the context of security notions of public key encryption (PKE), there are a lot of formulations considering different attack scenarios, such as chosen plaintext attacks (CPA) and chosen ciphertext attacks (CCA), and different attacker goals, such as one-wayness, indistinguishability (IND), and non-malleability. However, Bellare, Hofheinz, and Yilek [3] claimed that IND−CPA or IND−CCA security [7,9], which are the most accepted security notions for PKE, can not provide adequate security in a multi-user scenario. Concretely, they showed that when some of users has been corrupted, there are situations where we cannot preserve the other users' confidentiality of ciphertexts by using only IND−CPA or IND−CCA secure PKE schemes. Then, they proposed *selective opening (SO)* security for PKE which can ensure that the uncorrupted users' ciphertexts leak no information about their secrets.

Depending on different attack scenarios, SO security is divided into two settings: *sender selective opening (SSO)* security [3,4] and *receiver selective opening (RSO)* security [2,14]. In this paper, we focus on RSO security. In RSO security, we consider a situation where there are one sender and multiple receivers. An adversary can corrupt some receivers, which means he gets their secret keys and plaintexts. RSO security ensures the confidentiality of uncorrupted receivers' ciphertexts. Here, if we also consider an active adversary who can execute CCA, we can also consider RSO−CCA security for PKE.

From another point of view, there are two flavors of definitions for SO security: indistinguishability-based SO security and simulation-based SO security. As mentioned in some previous works [2,14], simulation-based SO security is more desirable than indistinguishability-based SO security, because the definition of indistinguishability-based SO security can support only a plaintext space which satisfies a notion called *efficient resamplability* [3]. Roughly, efficient resamplability refers to when a part of plaintexts are fixed, the remaining plaintexts can be resampled efficiently. This requirement is somewhat artificial and limits real-world applications because a plaintext distribution in practice scenarios do not necessarily satisfy this requirement.

From the above arguments, simulation-based RSO−CCA (SIM−RSO−CCA) security is the most favorable notion among all of the RSO security. Recently, some works [10–12] proposed constructions of SIM−RSO−CCA secure PKE under standard computational assumptions, such as the decisional Diffie-Hellman (DDH) assumption and the decisional composite residuosity (DCR) assumption. One of the important research area for cryptography is making a cryptographic primitive under the various assumptions. More specifically, we have two main problems in this area: Can we construct a cryptographic primitive under a *weaker computational assumption* or a *post-quantum computational assumption ?* In particular, National Institute of Standards and Technology (NIST) launched the Post-Quantum Cryptography Standardization in 2016, and thus post-quantum cryptography has been attracting more attention. Hence, in this paper, we tackle the following question:

> Is it possible to construct a SIM-RSO-CCA secure PKE scheme from weaker or post-quantum computational assumptions ?

1.2 Our Contribution

Based on the above motivation, we give affirmative answers to the question. More precisely, we show that SIM−RSO−CCA secure PKE can be constructed under the computational Diffie-Hellman (CDH) assumption (weaker computational assumption) or the learning parity with noise (LPN) assumption (new post-quantum computational assumption). In the following, we explain the details of our contribution.

Hara et al.'s Approach and Its Limitation. Toward our goal, we focus on the Hara et al.'s work [10,11]. In [10,11], they introduced the *receiver non-committing CCA (*RNC−CCA*)* security for receiver non-committing encryption (RNCE), which is a variant of PKE with a special non-committing property, then showed that RNC−CCA secure RNCE implies SIM−RSO−CCA secure PKE. Moreover, they proposed a construction of RNC−CCA secure RNCE by using an IND−CPA secure PKE scheme and a non-interactive zero-knowledge (NIZK) proof system satisfying one-time simulation soundness.[1] In a nutshell, their construction is obtained by combining the classical Naor-Yung paradigm [21] and a trick for a non-committing property that the decryption key used in a decryption algorithm of their RNCE scheme is chosen at random from two decryption keys of an underlying IND−CPA secure PKE scheme.

In order to obtain a SIM−RSO−CCA secure PKE scheme under the CDH or LPN assumption through their generic construction, all of the components of their construction should be realized under the CDH or LPN assumption. Actually, we can construct an IND−CPA secure PKE scheme based on the CDH assumption [13] or the LPN assumption [1,25]. However, NIZK proof system has not been proposed under these assumptions so far, and thus we cannot obtain a CDH or LPN based SIM−RSO−CCA secure PKE scheme through this generic construction.

Our Approach. In order to circumvent the above problem, we show that an NIZK proof system is not needed, but a *designated-verifier* NIZK (DV-NIZK) argument is sufficient for our goal. More specifically, we show that RNC−CCA secure RNCE can be obtained from IND−CPA secure PKE and DV-NIZK argument satisfying one-time simulation soundness. Roughly, a DV-NIZK argument is a relaxation of an NIZK proof system to the designated-verifier model, in other words, the model which only a user who has a secret verification key can verify a proof correctly. Although it is known that IND−CCA secure PKE scheme can be constructed from these two primitives [8], we have the following nebulous point for proving the RNC−CCA security for RNCE.

In contrast to IND−CCA security for PKE, when showing RNC−CCA security for RNCE, we have to consider a situation where an adversary can get a decryption key in a security game. For checking the validity of a ciphertext, we need to include a secret verification key of DV-NIZK argument into a decryption key of our RNCE scheme. Furthermore, as well as the original Naor-Yung

[1] Due to the previous works [22,24], it is known that both of an IND−CPA secure PKE scheme and an NIZK proof system can be constructed based on the learning with errors (LWE) assumption, which is one of the post-quantum computational assumption. Thus, by combining with the result [10], we can obtain a SIM−RSO−CCA secure PKE scheme based on the LWE assumption.

paradigm [21], we need to use zero-knowledge and (one-time simulation) soundness of a DV-NIZK argument in our security proof. However, in DV-NIZK setting, we cannot ensure soundness if a secret verification key is revealed to an adversary, while zero-knowledge still holds. Thus, it seems that our strategy does not make sense at first glance. However, by focusing on the details of a security proof, we have seen through that there is no problem. The main reason is that soundness is only used to prevent an adversary from making "unfavorable" decryption queries in a security proof, and thus soundness need to be held only while it makes decryption queries. Here, in RNC-CCA security, we consider decryption queries only before a decryption key (including a secret verification key) is revealed. Therefore, it is possible to prevent unfavorable decryption queries without a secret verification key, that is, soundness of DV-NIZK argument is sufficient for proving RNC−CCA security of RNCE. See Sect. 4.2 for more details.

Construction of One-time Simulation Sound DV-NIZK. Recently, a lot of works [6,17,18,20,23] showed that a DV-NIZK argument can be constructed under the CDH and LPN assumption. The notion of a one-time simulation sound DV-NIZK argument was considered in the Elkind et al.'s work [8]. However, a concrete construction of one-time simulation sound DV-NIZK argument has not been proposed so far. Then, in order to complete our RNC−CCA secure RNCE scheme, we propose a construction of one-time simulation sound DV-NIZK argument by combining (ordinary) DV-NIZK argument, strong one-time signature, and commitment based on the Lindell's approach [19]. See Sect. 3 for more details. (Note that we can construct strong one-time signature and commitment based on the one-way function, which is obtained under the CDH or LPN assumption.)

By combining the above results, we can obtain a SIM−RSO−CCA secure PKE scheme based on the CDH or LPN assumption.

1.3 Related Work

Jia et al. [15] proposed the first construction of SIM−RSO−CCA secure PKE using indistinguishability obfuscation. Moreover, Jia et al. [16] proposed indistinguishability-based RSO-CCA (IND-RSO-CCA) secure PKE schemes based on standard computational assumptions. Concretely, they showed two generic constructions of IND−RSO−CCA secure PKE. First, they gave a generic construction based on an IND−RSO−CPA secure PKE scheme, an IND−CCA secure PKE scheme, an NIZK proof system, and a strong one-time signature scheme. Second, they gave a generic construction based on a universal hash proof system. Recently, Huang et al. [12] showed that a SIM−RSO−CCA secure PKE scheme can be constructed under the DDH or DCR assumption. Moreover, they showed that a SIM−RSO−CCA secure PKE scheme can be constructed from an identity-based encryption scheme satisfying RSO security for a master secret key in the ideal cipher model.

2 Preliminaries

In this section, we define notations and recall the definitions for some cryptographic primitives.

2.1 Notations

In this paper, $x \leftarrow X$ denotes sampling an element x from a finite set X uniformly at random. $y \leftarrow \mathcal{A}(x; r)$ denotes that a probabilistic algorithm $\mathcal{A}$ outputs y for an input x using a randomness r, and we simply denote $y \leftarrow \mathcal{A}(x)$ when we do not need to write an internal randomness explicitly. For strings x and y, $x \| y$ denotes the concatenation of x and y. Also, $x := y$ denotes that x is defined by y. λ denotes a security parameter. A function $f(\lambda)$ is a negligible function in λ, if $f(\lambda)$ tends to 0 faster than $\frac{1}{\lambda^c}$ for every constant $c > 0$. $\mathsf{negl}(\lambda)$ denotes an unspecified negligible function. PPT stands for probabilistic polynomial time. If n is a natural number, $[n]$ denotes a set of integers $\{1, \cdots, n\}$. Also, if a and b are integers such that $a \leq b$, $[a, b]$ denotes a set of integers $\{a, \cdots, b\}$. If $\mathbf{m} = (m_1, \cdots, m_n)$ is an n-dimensional vector, $\mathbf{m}_J$ denotes a subset $\{m_j\}_{j \in J}$ where $J \subseteq [n]$. If $\mathcal{O}$ is a function or an algorithm and $\mathcal{A}$ is an algorithm, $\mathcal{A}^{\mathcal{O}}$ denotes that $\mathcal{A}$ has an oracle access to $\mathcal{O}$.

2.2 Public Key Encryption

Here, we review the definition of public key encryption (PKE).

Definition 1 (Public key encryption). *A PKE scheme with a plaintext space $\mathcal{M}$ consists of a tuple of the following three PPT algorithms $\Pi = (\mathbf{KG}, \mathbf{Enc}, \mathbf{Dec})$.*

KG*: The key generation algorithm, given a security parameter 1^λ, outputs a public key pk and a secret key sk.*

Enc*: The encryption algorithm, given a public key pk and a plaintext $m \in \mathcal{M}$, outputs a ciphertext c.*

Dec*: The (deterministic) decryption algorithm, given a public key pk, a secret key sk, and a ciphertext c, outputs a plaintext $m \in \{\bot\} \cup \mathcal{M}$.*

As the correctness for Π, we require that $\mathbf{Dec}(pk, sk, \mathbf{Enc}(pk, m)) = m$ holds for all $\lambda \in \mathbb{N}$, $m \in \mathcal{M}$, and $(pk, sk) \leftarrow \mathbf{KG}(1^\lambda)$.

Then, we recall IND−CPA security and SIM−RSO−CCA security for PKE.[2]

Definition 2 (IND-CPA security). *We say that $\Pi = (\mathbf{KG}, \mathbf{Enc}, \mathbf{Dec})$ is* IND−CPA *secure if for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,*

$$\mathsf{Adv}^{\mathsf{ind-cpa}}_{\Pi,\mathcal{A}}(\lambda) := 2 \cdot \Big| \Pr[b \leftarrow \{0,1\}; (pk, sk) \leftarrow \mathbf{KG}(1^\lambda); (m_0^*, m_1^*, \mathsf{st}_1) \leftarrow \mathcal{A}_1(pk);$$
$$c^* \leftarrow \mathbf{Enc}(pk, m_b^*); b' \leftarrow \mathcal{A}_2(c^*, \mathsf{st}_1) : b = b'] - \frac{1}{2} \Big| = \mathsf{negl}(\lambda),$$

[2] In this paper, as mentioned in Sect. 1.2, we focus on RNC−CCA secure RNCE to obtain a new SIM−RSO−CCA secure PKE scheme. Although we do not use a SIM−RSO−CCA security for PKE, we recall the definition here for completeness.

Definition 3 (SIM-RSO-CCA security). *Let n be the number of users. For a PKE scheme $\Pi = (\mathbf{KG}, \mathbf{Enc}, \mathbf{Dec})$, an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$, and a simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$, we define the following pair of experiments.*

```
$\mathsf{Exp}^{\mathsf{rso-cca-real}}_{n,\Pi,\mathcal{A}}(\lambda)$ :
    $(\mathbf{pk}, \mathbf{sk}) := (pk_j, sk_j)_{j\in[n]} \leftarrow (\mathbf{KG}(1^\lambda))_{j\in[n]}$
    $(\mathsf{Dist}, \mathsf{st}_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathbf{Dec}}(\cdot,\cdot)}(\mathbf{pk})$
    $\mathbf{m}^* := (m^*_j)_{j\in[n]} \leftarrow \mathsf{Dist}$
    $\mathbf{c}^* := (c^*_j)_{j\in[n]} \leftarrow (\mathbf{Enc}(pk_j, m^*_j))_{j\in[n]}$
    $(J, \mathsf{st}_2) \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathbf{Dec}}(\cdot,\cdot)}(\mathbf{c}^*, \mathsf{st}_1)$
    $\mathsf{out} \leftarrow \mathcal{A}_3^{\mathcal{O}_{\mathbf{Dec}}(\cdot,\cdot)}(\mathbf{sk}_J, \mathbf{m}^*_J, \mathsf{st}_2)$
    Return $(\mathbf{m}^*, \mathsf{Dist}, J, \mathsf{out})$

$\mathsf{Exp}^{\mathsf{rso-cca-sim}}_{n,\Pi,\mathcal{S}}(\lambda)$ :
    $(\mathsf{Dist}, \mathsf{st}_1) \leftarrow \mathcal{S}_1(1^\lambda)$
    $\mathbf{m}^* := (m^*_j)_{j\in[n]} \leftarrow \mathsf{Dist}$
    $(J, \mathsf{st}_2) \leftarrow \mathcal{S}_2(\mathsf{st}_1)$
    $\mathsf{out} \leftarrow \mathcal{S}_3(\mathbf{m}^*_J, \mathsf{st}_2)$
    Return $(\mathbf{m}^*, \mathsf{Dist}, J, \mathsf{out})$
```

In both of the experiments, we require that the distributions Dist *output by $\mathcal{A}$ and $\mathcal{S}$ be efficiently samplable. In $\mathsf{Exp}^{\mathsf{rso-cca-real}}_{n,\Pi,\mathcal{A}}(\lambda)$, a decryption query (c, j) is answered by $\mathbf{Dec}(pk_j, sk_j, c)$. $\mathcal{A}_2$ and $\mathcal{A}_3$ are not allowed to make a decryption query (c^*_j, j) for any $j \in [n]$. Furthermore, $\mathcal{A}_3$ is not allowed to make a decryption query (c, j) satisfying $j \in J$. (This is without losing generality, since $\mathcal{A}_3$ can decrypt any ciphertext using the given secret keys.)*

We say that Π is SIM−RSO−CCA *secure if for any PPT adversary $\mathcal{A}$ and any positive integer $n = n(\lambda)$, there exists a PPT simulator $\mathcal{S}$ such that for any PPT distinguisher $\mathcal{D}$,*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{rso-cca}}_{n,\Pi,\mathcal{A},\mathcal{S},\mathcal{D}}(\lambda) &:= |\Pr[\mathcal{D}(\mathsf{Exp}^{\mathsf{rso-cca-real}}_{n,\Pi,\mathcal{A}}(\lambda)) = 1] - \Pr[\mathcal{D}(\mathsf{Exp}^{\mathsf{rso-cca-sim}}_{n,\Pi,\mathcal{S}}(\lambda)) = 1]| \\ &= \mathsf{negl}(\lambda).\end{aligned}$$

2.3 Receiver Non-committing Encryption

Here, we review receiver non-committing encryption (RNCE) [5]. Informally, RNCE is public key encryption (PKE) having the property that it can generate a fake ciphertext which can be later opened to any plaintext (by showing an appropriate secret key). In the following, we give a syntax of RNCE and RNC−CCA security for it [10].

Definition 4 (Receiver non-committing encryption). *An RNCE scheme Π with a plaintext space $\mathcal{M}$ consists of the following seven PPT algorithms $(\mathbf{KG}, \mathbf{Enc}, \mathbf{Dec}, \mathbf{FKG}, \mathbf{Fake}, \mathbf{Open}, \mathbf{FDec})$. $(\mathbf{KG}, \mathbf{Enc}, \mathbf{Dec})$ are the same algorithms as those of a PKE scheme. $(\mathbf{FKG}, \mathbf{Fake}, \mathbf{Open}, \mathbf{FDec})$ are defined as follows.*

FKG*: The fake key generation algorithm, given a security parameter 1^λ, outputs a public key pk and a trapdoor td.*

Fake*: The fake encryption algorithm, given a public key pk and a trapdoor td, outputs a fake ciphertext $\widetilde{c}$.*

Open*: The opening algorithm, given a public key pk, a trapdoor td, a fake ciphertext $\widetilde{c}$, and a plaintext m, outputs a fake secret key $\widetilde{sk}$.*

FDec: *The fake decryption algorithm, given a public key pk, a trapdoor td, and a ciphertext c, outputs $m \in \{\perp\} \cup \mathcal{M}$.*

Definition 5 (RNC-CCA security). *For an RNCE scheme $\Pi = (\mathbf{KG}, \mathbf{Enc}, \mathbf{Dec}, \mathbf{FKG}, \mathbf{Fake}, \mathbf{Open}, \mathbf{FDec})$ and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$, we consider the following pair of experiments.*

```
$\mathsf{Exp}^{\mathsf{rnc-real}}_{\Pi,\mathcal{A}}(\lambda)$ :
    $(pk, sk) \leftarrow \mathbf{KG}(1^\lambda)$
    $(m^*, \mathsf{st}_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathbf{Dec}}(\cdot)}(pk)$
    $c^* \leftarrow \mathbf{Enc}(pk, m^*)$
    $\mathsf{st}_2 \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathbf{Dec}}(\cdot)}(c^*, \mathsf{st}_1)$
    $sk^* := sk$
    Return $b' \leftarrow \mathcal{A}_3(sk^*, \mathsf{st}_2)$

$\mathsf{Exp}^{\mathsf{rnc-sim}}_{\Pi,\mathcal{A}}(\lambda)$ :
    $(pk, td) \leftarrow \mathbf{FKG}(1^\lambda)$
    $(m^*, \mathsf{st}_1) \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathbf{Dec}}(\cdot)}(pk)$
    $c^* \leftarrow \mathbf{Fake}(pk, td)$
    $\mathsf{st}_2 \leftarrow \mathcal{A}_2^{\mathcal{O}_{\mathbf{Dec}}(\cdot)}(c^*, \mathsf{st}_1)$
    $sk^* \leftarrow \mathbf{Open}(pk, td, c^*, m^*)$
    Return $b' \leftarrow \mathcal{A}_3(sk^*, \mathsf{st}_2)$
```

In $\mathsf{Exp}^{\mathsf{rnc-real}}_{\Pi,\mathcal{A}}(\lambda)$, a decryption query c is answered by $\mathbf{Dec}(pk, sk, c)$. On the other hand, in $\mathsf{Exp}^{\mathsf{rnc-sim}}_{\Pi,\mathcal{A}}(\lambda)$, a decryption query c is answered by $\mathbf{FDec}(pk, td, c)$. In both of the experiments, $\mathcal{A}_2$ is not allowed to make a decryption query $c = c^*$ and $\mathcal{A}_3$ is not allowed to make any decryption query. We say that Π is RNC−CCA *secure if for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{rnc-cca}}_{\Pi,\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{rnc-real}}_{\Pi,\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{rnc-sim}}_{\Pi,\mathcal{A}}(\lambda) = 1]| = \mathsf{negl}(\lambda)$ holds.*

In the previous work [10], the following theorem was shown.

Theorem 1 ([10]). *If an RNCE scheme $\Pi = (\mathbf{KG}, \mathbf{Enc}, \mathbf{Dec}, \mathbf{FKG}, \mathbf{Fake}, \mathbf{Open}, \mathbf{FDec})$ is* RNC−CCA *secure, then a PKE scheme $\Pi_{\mathsf{rso}} := (\mathbf{KG}, \mathbf{Enc}, \mathbf{Dec})$ is* SIM−RSO−CCA *secure.*

2.4 Signature

Here, we review the definition of a signature scheme.

Definition 6 (Signature). *A signature scheme Σ with a message space $\mathcal{M}$ consists of the following three PPT algorithms.*

SKG: *The key generation algorithm, given a security parameter 1^λ, outputs a verification key vk and a signing key $sigk$.*

Sign: *The signing algorithm, given a signing key $sigk$ and a message m, and outputs a signature σ.*

SVer: *The verification algorithm, given a verification key vk, a message m, and a signature σ, outputs either* 1 *(meaning "accept") or* 0 *(meaning "reject").*

As the correctness for Σ, we require that for all $\lambda \in \mathbb{N}$, $(vk, sigk) \leftarrow \mathbf{SKG}(1^\lambda)$, and messages $m \in \mathcal{M}$, it holds that $\mathbf{SVer}(vk, m, \mathbf{Sign}(sigk, m)) = 1$.

Next, we define strong one-time unforgeability under chosen message attacks for a signature scheme.

Definition 7 (Strong one-time unforgeability). *We say that a signature scheme $\Sigma = (\mathbf{SKG}, \mathbf{Sign}, \mathbf{SVer})$ satisfies strong one-time unforgeability if for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,*

$$\mathsf{Adv}^{\mathsf{unf}}_{\Omega,\mathcal{A}}(\lambda) := \Pr[(vk, sigk) \leftarrow \mathbf{SKG}(1^\lambda); (m, \mathsf{st}_1) \leftarrow \mathcal{A}_1(vk); \sigma \leftarrow \mathbf{Sign}(sigk, m); (m', \sigma') \leftarrow \mathcal{A}_2(\sigma, \mathsf{st}_1) : ((m', \sigma') \neq (m, \sigma)) \wedge (\mathbf{SVer}(vk, m', \sigma') = 1)] = \mathsf{negl}(\lambda)$$

holds.

2.5 Commitment

Here, we review the definition of a commitment scheme.

Definition 8 (Commitment). *A Commitment scheme Ω with a plaintext space $\mathcal{M}$ consists of the following two PPT algorithms.*

CKG: *The key generation algorithm, given a security parameter 1^λ, outputs a public commitment key ck.*

Commit: *The commit algorithm, given a public commitment key ck and a plaintext m, outputs a commitment c.*

Next, we define the following two security properties for commitment: *statistical binding* and *computationally hiding.*

Definition 9 (Statistical binding). *Let $\Omega = (\mathbf{CKG}, \mathbf{Commit})$ be a commitment scheme. We say that Ω satisfies statistical binding if for any computationally unbounded adversary $\mathcal{A}$,*

$$\mathsf{Adv}^{\mathsf{bind}}_{\Omega,\mathcal{A}}(\lambda) := \Pr[ck \leftarrow \mathbf{CKG}(1^\lambda); (m_0, m_1, r_0, r_1) \leftarrow \mathcal{A}(ck) : \mathbf{Commit}(ck, m_0; r_0) = \mathbf{Commit}(ck, m_1; r_1)] = \mathsf{negl}(\lambda)$$

holds.

Definition 10 (Computational hiding). *We say that a commitment scheme $\Omega = (\mathbf{CKG}, \mathbf{Commit})$ satisfies computationally hiding if for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,*

$$\mathsf{Adv}^{\mathsf{hide}}_{\Omega,\mathcal{A}}(\lambda) := \left| \Pr[b \leftarrow \{0,1\}; ck \leftarrow \mathbf{CKG}(1^\lambda); (m_0, m_1, \mathsf{st}_1) \leftarrow \mathcal{A}_1(ck); c \leftarrow \mathbf{Commit}(ck, m_b); b' \leftarrow \mathcal{A}_2(c, \mathsf{st}_1) : b = b'] - \frac{1}{2} \right| = \mathsf{negl}(\lambda)$$

holds.

2.6 Designated-Verifier Non-interactive Zero-Knowledge Arguments

Here, we review the definition of a designated-verifier non-interactive zero-knowledge (DV-NIZK) argument [6,17,18,20,23].

Definition 11 (DV-NIZK argument). *Let $\mathcal{R}$ be an efficiently computable binary relation and $\mathcal{L} := \{x \mid \exists w \text{ s.t.} (x, w) \in \mathcal{R}\}$. A DV-NIZK argument for $\mathcal{L}$ consists of a tuple of the following five PPT algorithms $\Phi = (\mathbf{CRSGen}, \mathbf{Prove}, \mathbf{Verify}, \mathbf{SimCRS}, \mathbf{SimPrv})$.*

CRSGen*: The common reference string (CRS) generation algorithm takes a security parameter 1^λ as input, and outputs a CRS crs and a secret verification key vsk.*
Prove*: The proving algorithm takes a CRS crs, a statement x, and a witness w as input, and outputs a proof π.*
Verify*: The (deterministic) verification algorithm takes a CRS crs, a secret verification key vsk, a statement x, and a proof π as input, outputs a bit $v \in \{0, 1\}$, which is either 1 (meaning "accept") or 0 (meaning "reject").*
SimCRS*: The simulator's CRS generation algorithm takes a security parameter 1^λ as input, outputs a simulated CRS crs, a simulated secret verification key vsk, and a trapdoor key tk.*
SimPrv*: The simulator's proving algorithm takes a trapdoor key tk and a statement x as input, and outputs a simulated proof π.*

We say that a DV-NIZK argument Φ is correct if we have $\mathbf{Verify}(crs, vsk, x, \mathbf{Prove}(crs, x, w)) = 1$ for all $\lambda \in \mathbb{N}$, $(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda)$, and valid statement / witness pairs $(x, w) \in \mathcal{R}$.

Next, we define (standard) soundness and one-time simulation soundness for a DV-NIZK argument. We adopt a definition of soundness which was considered in recent works [6,17,18,20,23]. Moreover, we adopt a definition of one-time simulation soundness proposed in [8]. We note that in both of security definitions, an adversary can make multiple verification queries.

Definition 12 (Soundness). *We say that a DV-NIZK argument $\Phi = (\mathbf{CRSGen}, \mathbf{Prove}, \mathbf{Verify}, \mathbf{SimCRS}, \mathbf{SimPrv})$ satisfies soundness if for any PPT adversary $\mathcal{A}$,*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{sound}}_{\Phi,\mathcal{A}}(\lambda) := \Pr[(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda); (x, \pi) \leftarrow \mathcal{A}^{\mathcal{O}(\cdot,\cdot)}(crs)\\ : (x \notin \mathcal{L}) \wedge (\mathbf{Verify}(crs, vsk, x, \pi) = 1)] = \mathsf{negl}(\lambda)\end{aligned}$$

holds, where $\mathcal{O}(\cdot, \cdot)$ is a verification oracle which receives a query (x, π) and returns $v \leftarrow \mathbf{Verify}(vsk, x, \pi)$.

Definition 13 (One-time simulation soundness). *We say that a DV-NIZK argument $\Phi = (\mathbf{CRSGen}, \mathbf{Prove}, \mathbf{Verify}, \mathbf{SimCRS}, \mathbf{SimPrv})$ satisfies one-time simulation soundness if for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ot-ss}}_{\Phi,\mathcal{A}}(\lambda) := \Pr[(crs, tk, vsk) \leftarrow \mathbf{SimCRS}(1^\lambda); (x', st_1) \leftarrow \mathcal{A}_1^{\mathcal{O}(\cdot,\cdot)}(crs);\\ \pi' \leftarrow \mathbf{SimPrv}(tk, x'); (x, \pi) \leftarrow \mathcal{A}_2^{\mathcal{O}(\cdot,\cdot)}(\pi', st_1)\\ : ((x,\pi) \neq (x', \pi')) \wedge (x \notin \mathcal{L}) \wedge (\mathbf{Verify}(crs, vsk, x, \pi) = 1)] = \mathsf{negl}(\lambda)\end{aligned}$$

holds, where $\mathcal{O}(\cdot,\cdot)$ is a verification oracle which receives a query (x,π) and returns $v \leftarrow \mathbf{Verify}(vsk, x, \pi)$.

Then, we give the definitions of zero-knowledge and witness indistinguishability for a DV-NIZK argument. We adopt a definition of zero-knowledge which was considered in [8]. Our definition of witness indistinguishability is a natural extension from one of a (standard) NIZK proof system. It is easy to see that our witness indistinguishability is implied by zero-knowledge.

Definition 14 (Zero-knowledge). *For a DV-NIZK argument $\Phi = (\mathbf{CRSGen}, \mathbf{Prove}, \mathbf{Verify}, \mathbf{SimCRS}, \mathbf{SimPrv})$ and a PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, we consider the following two experiments.*

```
$\mathsf{Exp}^{\mathsf{zk-real}}_{\Phi,\mathcal{A}}(\lambda)$ :
    $(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda)$
    $(x, w, \mathsf{st}_1) \leftarrow \mathcal{A}_1(crs, vsk)$
    $\pi \leftarrow \mathbf{Prove}(crs, x, w)$
    $b' \leftarrow \mathcal{A}_2(\pi, \mathsf{st}_1)$
    Return $b'$
```

```
$\mathsf{Exp}^{\mathsf{zk-sim}}_{\Phi,\mathcal{A}}(\lambda)$ :
    $(crs, vsk, tk) \leftarrow \mathbf{SimCRS}(1^\lambda)$
    $(x, w, \mathsf{st}_1) \leftarrow \mathcal{A}_1(crs, vsk)$
    $\pi \leftarrow \mathbf{SimPrv}(tk, x)$
    $b' \leftarrow \mathcal{A}_2(\pi, \mathsf{st}_1)$
    Return $b'$
```

In both of the experiments, it is required that $x \in \mathcal{L}$ and w be a witness for $x \in \mathcal{L}$. We say that Φ is zero-knowledge if for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{zk}}_{\Phi,\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{zk-real}}_{\Phi,\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{zk-sim}}_{\Phi,\mathcal{A}}(\lambda) = 1]| = \mathsf{negl}(\lambda)$ *holds.*

Definition 15 (Witness indistinguishability). *We say that a DV-NIZK argument $\Phi = (\mathbf{CRSGen}, \mathbf{Prove}, \mathbf{Verify}, \mathbf{SimCRS}, \mathbf{SimPrv})$ satisfies witness indistinguishability if for any PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$,*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{wi}}_{\Phi,\mathcal{A}}(\lambda) := 2 \cdot \Big| \Pr[(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda);\\ (x, w_0, w_1, \mathsf{st}_1) \leftarrow \mathcal{A}_1(crs, vsk); b \leftarrow \{0,1\};\\ \pi \leftarrow \mathbf{Prove}(crs, x, w_b); b' \leftarrow \mathcal{A}_2(\pi, \mathsf{st}_1) : b = b'] - \frac{1}{2}\Big| = \mathsf{negl}(\lambda),\end{aligned}$$

where $(x, w_0), (x, w_1) \in \mathcal{R}$ holds.

3 Construction of One-Time Simulation Sound DV-NIZK

In this section, we provide a construction of one-time simulation sound DV-NIZK. First, in Sect. 3.1, we describe our construction. Then, in Sect. 3.2, we give a security proof for our construction.

3.1 Description

In this section, we formally describe our construction of one-time simulation sound DV-NIZK argument for an NP language $\mathcal{L}'$. Let $\Sigma = (\mathbf{SKG}, \mathbf{Sign}, \mathbf{SVer})$ be a signature scheme, $\Omega = (\mathbf{CKG}, \mathbf{Commit})$ a commitment scheme, and $\Pi = (\mathbf{CRSGen}, \mathbf{Prove}, \mathbf{Verify}, \mathbf{SimCRS}, \mathbf{SimPrv})$ a (standard) DV-NIZK argument for $\mathcal{L}$, where

$$\mathcal{L} := \Big\{ (x', ck, vk, c) \,|\, \exists\, (w', r) \,\text{s.t.}\, ((x', w') \in \mathcal{R}') \vee (c = \mathbf{Commit}(ck, vk; r)) \Big\}.$$

Then, we construct our one-time simulation sound DV-NIZK argument $\Phi' = (\mathbf{CRSGen}', \mathbf{Prove}', \mathbf{Verify}', \mathbf{SimCRS}', \mathbf{SimPrv}')$ for $\mathcal{L}'$ as described in Fig.1.

3.2 Security Proof

In this section, we show that our scheme Φ' satisfies one-time simulation soundness (Theorem 2) and zero-knowledge (Theorem 3).

Theorem 2. *If Φ satisfies (standard) soundness, Ω satisfies statistical binding, and Σ satisfies strong one-time unforgeability, then Φ' satisfies one-time simulation soundness.*

Proof of Theorem 2. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a PPT adversary that attacks the one-time simulation soundness of Φ'. The detailed description of one-time simulation soundness for Φ' is as follows.

1. The challenger generates $ck \leftarrow \mathbf{CKG}(1^\lambda)$, $(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda)$, and $(sigk^*, vk^*) \leftarrow \mathbf{SKG}(1^\lambda)$. Then, it samples $r \leftarrow \mathcal{R}_\Pi$ and computes $c^* \leftarrow \mathbf{Commit}(ck, vk^*; r)$. Finally, it sets $crs' := (crs, ck, c^*)$ and $tk := (vk^*, sigk^*, r)$, and runs $\mathcal{A}_1(crs')$. When $\mathcal{A}_1$ makes a verification query $(\widetilde{x}, \widetilde{\pi})$, the challenger returns $v \leftarrow \mathbf{Verify}(crs, vsk, \widetilde{x}, \widetilde{\pi})$ to $\mathcal{A}_1$.
2. When $\mathcal{A}_1$ outputs $(\hat{x}', \mathsf{st}_1)$ and terminates, the challenger sets $\hat{x} := (\hat{x}', ck, vk^*, c^*)$ and $\hat{w} := (\perp, r)$, and computes $\hat{\pi} \leftarrow \mathbf{Prove}(crs, \hat{x}, \hat{w})$ and $\hat{\sigma} \leftarrow \mathbf{Sign}(sigk^*, (\hat{x}', \hat{\pi}))$. Then, it sets $\hat{\pi}' := (vk^*, \hat{\pi}, \hat{\sigma})$ and runs $\mathcal{A}_2(\hat{\pi}', \mathsf{st}_1)$. When $\mathcal{A}_2$ makes a verification query $(\widetilde{x}, \widetilde{\pi})$, the challenger returns $v \leftarrow \mathbf{Verify}(crs, vsk, \widetilde{x}, \widetilde{\pi})$ to $\mathcal{A}_2$.
3. $\mathcal{A}_2$ outputs a pair of a statement and a proof $(x', \pi' = (vk, \pi, \sigma))$ and terminates.

Here, in the above experiment, we let $\mathbf{Win}$ be the event that $((x', \pi') \neq (\hat{x}', \hat{\pi}')) \wedge (x' \notin \mathcal{L}') \wedge (\mathbf{Verify}(crs', vsk, x', \pi') = 1)$ holds. We have the inequality $\mathsf{Adv}^{\mathsf{ot-ss}}_{\Phi', \mathcal{A}}(\lambda) = \Pr[\mathbf{Win}] = \Pr[\mathbf{Win} \wedge vk \neq vk^*] + \Pr[\mathbf{Win} \wedge vk = vk^*]$.

In the following, we show that there exist a PPT adversary $\mathcal{B}$ against the soundness of Φ such that $\mathsf{Adv}^{\mathsf{sound}}_{\Phi, \mathcal{B}}(\lambda) = \Pr[\mathbf{Win} \wedge vk \neq vk^*]$ (Lemma 1) and a PPT adversary $\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2)$ against the strong one-time unforgeability of Σ such that $\mathsf{Adv}^{\mathsf{unf}}_{\Sigma, \mathcal{C}}(\lambda) = \Pr[\mathbf{Win} \wedge vk = vk^*]$ (Lemma 2).

```
$\mathbf{CRSGen}'(1^\lambda)$ :
    $ck \leftarrow \mathbf{CKG}(1^\lambda)$
    $(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda)$
    $r \leftarrow \mathcal{R}_\Omega$
    $c \leftarrow \mathbf{Commit}(ck, 0^{|vk|}; r)$
    $crs' := (crs, ck, c)$
    Return $(crs', vsk)$
```

```
$\mathbf{Prove}'(crs', x', w')$ :
    $(vk, sigk) \leftarrow \mathbf{SKG}(1^\lambda)$
    $x := (x', ck, vk, c)$
    $w := (w', \perp)$
    $\pi \leftarrow \mathbf{Prove}(crs, x, w)$
    $\sigma \leftarrow \mathbf{Sign}(sigk, (x', \pi))$
    Return $\pi' := (vk, \pi, \sigma)$
```

```
$\mathbf{Verify}'(crs', vsk, x', \pi')$ :
    If $\mathbf{SVer}(vk, (x', \pi), \sigma) = 1$
    $\wedge \mathbf{Verify}(crs, vsk, (x', ck, vk, c), \pi) = 1$ then
        Return 1
    else Return 0
```

```
$\mathbf{SimCRS}'(1^\lambda)$ :
    $ck \leftarrow \mathbf{CKG}(1^\lambda)$
    $(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda)$
    $(sigk^*, vk^*) \leftarrow \mathbf{SKG}(1^\lambda)$
    $r \leftarrow \mathcal{R}_\Pi$
    $c^* \leftarrow \mathbf{Commit}(ck, vk^*; r)$
    $crs' := (crs, ck, c^*)$
    $tk := (vk^*, sigk^*, r)$
    Return $(crs', tk, vsk)$
```

```
$\mathbf{SimPrv}'(crs', tk, x')$ :
    $x := (x', ck, vk^*, c^*)$
    $w := (\perp, r)$
    $\pi^* \leftarrow \mathbf{Prove}(crs, x, w)$
    $\sigma^* \leftarrow \mathbf{Sign}(sigk^*, (x', \pi^*))$
    Return $\pi' := (vk^*, \pi^*, \sigma^*)$
```

Fig. 1. Construction of one-time simulation sound DV-NIZK argument Φ'.

Lemma 1. *There exists a PPT adversary $\mathcal{B}$ such that* $\mathsf{Adv}^{\mathsf{sound}}_{\Phi,\mathcal{B}}(\lambda) = \Pr[\mathbf{Win} \wedge vk \neq vk^*]$.

Proof of Lemma 1. We construct a PPT adversary $\mathcal{B}$ that attacks the soundness of Φ so that $\mathsf{Adv}^{\mathsf{sound}}_{\Phi,\mathcal{B}}(\lambda) = \Pr[\mathbf{Win} \wedge vk \neq vk^*]$, using the adversary $\mathcal{A}$ as follows.

$\mathcal{B}(crs)$ **:** First, $\mathcal{B}$ generates $ck \leftarrow \mathbf{CKG}(1^\lambda)$, $(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda)$, and $(sigk^*, vk^*) \leftarrow \mathbf{SKG}(1^\lambda)$. Then, it samples $r \leftarrow \mathcal{R}_\Pi$ and computes $c^* \leftarrow \mathbf{Commit}(ck, vk^*; r)$. Next, it sets $crs' := (crs, ck, c)$ and $tk' := (vk^*, sigk^*, r)$, and runs $\mathcal{A}_1(crs')$. When $\mathcal{A}_1$ makes a verification query $(\widetilde{x}', (\widetilde{vk}, \widetilde{\pi}, \widetilde{\sigma}))$, $\mathcal{B}$ computes $s \leftarrow \mathbf{SVer}(\widetilde{vk}, (\widetilde{x}', \widetilde{\pi}), \widetilde{\sigma})$, makes a verification query $((\widetilde{x}', ck, \widetilde{vk}, c), \widetilde{\pi})$, and gets the result v. If $s = v = 1$ holds, $\mathcal{B}$ returns 1 to $\mathcal{A}_1$. Otherwise, $\mathcal{B}$ returns 0 to $\mathcal{A}_1$.
When $\mathcal{A}_1$ outputs a pair of a statement and state information $(\hat{x}', \mathsf{st}_1)$ and terminates, $\mathcal{B}$ sets $\hat{x} := (\hat{x}', vk, c)$ and $\hat{w} := (\perp, r)$. Next, $\mathcal{B}$ computes $\hat{\pi} \leftarrow \mathbf{Prove}(crs, \hat{x}, \hat{w})$ and $\hat{\sigma} \leftarrow \mathbf{Sign}(sigk^*, (\hat{\pi}, \hat{x}'))$. Then, $\mathcal{B}$ sets $\hat{\pi}' := (\widetilde{vk}, \hat{\pi}, \hat{\sigma})$ and runs $\mathcal{A}_2(\hat{\pi}', \mathsf{st}_1)$. When $\mathcal{A}_2$ makes a verification query $((\widetilde{x}', ck, \widetilde{vk}, c), \widetilde{\pi})$, $\mathcal{B}$ answers in the same way as above. When $\mathcal{A}_2$ outputs a pair of a statement and a proof $(x', (vk, \pi, \sigma))$ and terminates, $\mathcal{B}$ sets $x := (x', ck, vk, c)$, returns (x, π) to its experiment, and terminates.

We can see that $\mathcal{B}$ perfectly simulates an experiment of one-time simulation soundness for $\mathcal{A}$. Here, we assume that $vk \neq vk^*$ holds. Firstly, if the event **Win** occurs, $\textbf{Verify}'(crs', vsk, x', \pi') = 1$ holds, which means that $\textbf{Verify}(crs, vsk, (x', ck, vk, c), \pi) = 1$ holds.

Secondly, $x' \notin L'$ holds now. Moreover, due to the fact that $vk \neq vk^*$ and the statistical binding of Ω hold, we can see $\textbf{Commit}(ck, vk; r) \neq c$. Hence, we have $x \notin L$.

From the above argument, if **Win** occurs and $vk \neq vk^*$ holds, we can see that $\mathcal{B}$ can make a pair of a statement and a proof (x, π) breaking the soundness of Φ. Thus, $\mathsf{Adv}^{\mathsf{sound}}_{\Phi,\mathcal{B}}(\lambda) = \Pr[\textbf{Win} \wedge vk \neq vk^*]$ holds. □ **(Lemma 1)**

Lemma 2. *There exists a PPT adversary* $\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2)$ *such that* $\mathsf{Adv}^{\mathsf{unf}}_{\Sigma,\mathcal{C}}(\lambda) = \Pr[\textbf{Win} \wedge vk = vk^*]$.

Proof of Lemma 2. We construct a PPT adversary $\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2)$ that attacks the strong one-time unforgeability of Σ so that $\mathsf{Adv}^{\mathsf{unf}}_{\Sigma,\mathcal{C}}(\lambda) = \Pr[\textbf{Win} \wedge vk = vk^*]$, using the adversary $\mathcal{A}$ as follows.

$\mathcal{C}_1(vk^*)$**:** First, $\mathcal{C}_1$ generates $ck \leftarrow \textbf{CKG}(1^\lambda)$ and $(crs, vsk) \leftarrow \textbf{CRSGen}(1^\lambda)$. Next, $\mathcal{C}_1$ samples $r \leftarrow \mathcal{R}_\Pi$ and computes $c \leftarrow \textbf{Commit}(ck, vk^*; r)$. Then, $\mathcal{C}_1$ sets $crs' := (crs, ck, c)$ and runs $\mathcal{A}_1(crs')$. When $\mathcal{A}_1$ makes a verification query of $(\widetilde{x}', \widetilde{\pi}')$, $\mathcal{C}_1$ returns $v \leftarrow \textbf{Verify}'(crs', vsk, \widetilde{x}', \widetilde{\pi}')$ to $\mathcal{A}_1$.
When $\mathcal{A}_1$ outputs a pair of a statement and state information $(\hat{x}', \mathsf{st}_1)$ and terminates, $\mathcal{C}_1$ sets $\hat{x} := (\hat{x}', ck, vk^*, c)$ and $\hat{w} := (\perp, r)$, and computes $\hat{\pi} \leftarrow \textbf{Prove}(crs, \hat{x}, \hat{w})$. Then, $\mathcal{C}_1$ sets $\hat{m} := (\hat{x}', \hat{\pi})$ and st_1' as all the information known to $\mathcal{C}_1$, returns $(\hat{m}, \mathsf{st}_1)$ to its experiment, and terminates.

$\mathcal{C}_2(\hat{\sigma}, \mathsf{st}_1)$**:** First, $\mathcal{C}_2$ sets $\hat{\pi}' := (vk^*, \hat{\sigma}, \hat{\pi})$ and runs $\mathcal{A}_2(\hat{\pi}', \mathsf{st}_1)$. When $\mathcal{A}_2$ outputs a pair of a challenge statement and a proof $(x', (vk, \pi, \sigma))$, and terminates, $\mathcal{C}_2$ sets $m' := (x', \pi)$, returns (σ, m') to its experiment, and terminates.

We can see that $\mathcal{C}$ perfectly simulates an experiment of one-time simulation soundness for $\mathcal{A}$. Here, we assume that $vk = vk^*$ holds. If the event **Win** occurs, $\textbf{Verify}(crs', vsk, x', \pi') = 1$ holds, which means that $\textbf{SVer}(vk^*, m', \sigma) = 1$ holds. Moreover, $(x', \pi') \neq (\hat{x}', \hat{\pi}')$ holds now, which implies $(m', \sigma) \neq (\hat{m}, \hat{\sigma})$. From the above argument, if **Win** occurs and $vk = vk^*$ holds, we can see that $\mathcal{C}$ can make a pair of a statement and a proof (x', π) breaking the strong one-time unforgeability of Σ. Thus, $\mathsf{Adv}^{\mathsf{unf}}_{\Sigma,\mathcal{C}}(\lambda) = \Pr[\textbf{Win} \wedge vk = vk^*]$ holds. □ **(Lemma 2)**

Putting everything together, we obtain $\mathsf{Adv}^{\mathsf{ot-ss}}_{\Phi',\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{sound}}_{\Phi,\mathcal{B}}(\lambda) + \mathsf{Adv}^{\mathsf{unf}}_{\Sigma,\mathcal{C}}(\lambda)$. Since Φ satisfies (standard) soundness and Σ satisfies strong one-time unforgeability, for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{ot-ss}}_{\Phi',\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$ holds. Therefore, Φ' satisfies one-time simulation soundness. □ **(Theorem 2)**

Theorem 3. *If* Φ *satisfies witness indistinguishability and* Ω *satisfies computationally hiding, then* Φ' *satisfies zero-knowledge.*

Proof of Theorem 3. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a PPT adversary that attacks the zero-knowledge of Φ'. We introduce the following experiments $\{\mathsf{Exp}_i\}_{i=0}^{2}$.

Exp_0: Exp_0 is exactly the same as $\mathsf{Exp}^{\mathsf{zk-real}}_{\Phi',\mathcal{A}}(\lambda)$. The detailed description is as follows.

1. Exp_0 generates $ck \leftarrow \mathbf{CKG}(1^\lambda)$ and $(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda)$. Then, Exp_0 samples $r \leftarrow \mathcal{R}_\Omega$ and computes $c \leftarrow \mathbf{Commit}(0^{|vk|}; r)$. Next, Exp_0 sets $crs' := (crs, ck, c)$ and runs $\mathcal{A}_1(crs', vsk)$.
2. When $\mathcal{A}_1$ outputs a tuple (x', w', st_1) and terminates, Exp_0 generates $(vk, sigk) \leftarrow \mathbf{SKG}(1^\lambda)$ and sets $x := (x', ck, vk, c)$ and $w := (w', \perp)$. Then, Exp_0 computes $\pi \leftarrow \mathbf{Prove}(crs, x, w)$ and $\sigma \leftarrow \mathbf{Sign}(sigk, (x', \pi))$, and returns $\pi' := (vk, \pi, \sigma)$ to $\mathcal{A}_2$.
3. When $\mathcal{A}_2$ outputs a bit $b' \in \{0, 1\}$ and terminates, Exp_0 outputs b'.

Exp_1 : Exp_1 is identical to Exp_0 except that Exp_1 generates another $(sigk^*, vk^*) \leftarrow \mathbf{SKG}(1^\lambda)$ and computes $c \leftarrow \mathbf{Commit}(vk^*; r)$ instead of $c \leftarrow \mathbf{Commit}(0^{|vk|}; r)$.

Exp_2 : Exp_2 is identical to Exp_1 except that Exp_2 sets $w := (\perp, r)$ and uses this w to make a proof π. Note that Exp_2 is exactly the same as $\mathsf{Exp}^{\mathsf{zk-sim}}_{\Phi',\mathcal{A}}(\lambda)$.

We let $p_i := \Pr[\mathsf{Exp}_i(\lambda) = 1]$ for all $i \in [0, 2]$. Then, we have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{zk}}_{\Phi',\mathcal{A}}(\lambda) &= |\Pr[\mathsf{Exp}^{\mathsf{zk-real}}_{\Phi',\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{zk-sim}}_{\Phi',\mathcal{A}}(\lambda) = 1]| \\ &= |p_0 - p_2| \leq \sum_{i=0}^{1} |p_i - p_{i+1}|.\end{aligned}$$

It remains to show how each $|p_i - p_{i+1}|$ is upper-bounded. To this end, in the following, we show that there exist an adversary $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$ against the computationally hiding of Ω such that $|p_0 - p_1| = \mathsf{Adv}^{\mathsf{hide}}_{\Omega,\mathcal{D}}(\lambda)$ (Lemma 3) and an adversary $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ against the witness indistinguishability of Φ such that $|p_1 - p_2| = \mathsf{Adv}^{\mathsf{wi}}_{\Phi,\mathcal{E}}(\lambda)$ (Lemma 4).

Lemma 3. *There exists a PPT adversary $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$ such that $|p_0 - p_1| = \mathsf{Adv}^{\mathsf{hide}}_{\Omega,\mathcal{D}}(\lambda)$.*

Proof of Lemma 3. We construct a PPT adversary $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$ that attacks the hiding property of Ω so that $|p_0 - p_1| = \mathsf{Adv}^{\mathsf{hide}}_{\Omega,\mathcal{D}}(\lambda)$, using the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ as follows.

$\mathcal{D}_1(ck)$: First, $\mathcal{D}_1$ generates $(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda)$ and $(sigk^*, vk^*) \leftarrow \mathbf{SKG}(1^\lambda)$. Then, $\mathcal{D}_1$ sets $m_0 := 0^{|vk^*|}$, $m_1 := vk^*$, and st_1 as all of the information known to $\mathcal{D}_1$, returns $(m_0, m_1, \mathsf{st}_1)$ to its experiment, and terminates.

$\mathcal{D}_2(c)$: First, $\mathcal{D}_2$ sets $crs' := (crs, c)$ and runs $\mathcal{A}_1(crs')$. When $\mathcal{A}_1$ outputs a tuple (x', w', st'_1), $\mathcal{D}_2$ sets $x := (x', ck, vk^*, c)$ and $w := (w', \perp)$. Then, $\mathcal{D}_2$ computes $\pi \leftarrow \mathbf{Prove}(crs, x, w)$ and $\sigma \leftarrow \mathbf{Sign}(sigk^*, (x', \pi))$, sets $\pi' := (vk^*, \pi, \sigma)$, and runs $\mathcal{A}_2(\pi', \mathsf{st}_1)$. When $\mathcal{A}_2$ outputs a bit $b' \in \{0, 1\}$ and terminates, $\mathcal{D}_2$ returns b' to its experiment and terminates.

We let b be the challenge bit for $\mathcal{D}$ in its experiment. When $b = 0$, we can see that $\mathcal{D}$ perfectly simulates Exp_0 for $\mathcal{A}$. This ensures that when $b = 0$, the probability that $\mathcal{D}$ outputs 1 is exactly the same as the probability that $\mathcal{A}$ outputs $b' = 1$ in Exp_0. On the other hand, when $b = 1$, we can see that $\mathcal{D}$ perfectly simulates Exp_1 for $\mathcal{A}$. This ensures that when $b = 1$, the probability that $\mathcal{D}$ outputs 1 holds is exactly the same as the probability that $\mathcal{A}$ outputs $b' = 1$ in Exp_1. Therefore, we have $\mathsf{Adv}^{\mathsf{hide}}_{\Omega,\mathcal{D}}(\lambda) = |\Pr[b' = 1|b = 0] - \Pr[b' = 1|b = 1]| = |p_0 - p_1|$. □ **(Lemma 3)**

Lemma 4. *There exists a PPT adversary* $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ *such that* $|p_1 - p_2| = \mathsf{Adv}^{\mathsf{wi}}_{\Phi,\mathcal{E}}(\lambda)$.

Proof of Lemma 4. We construct a PPT adversary $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ that attacks the witness indistinguishability of Φ so that $|p_1 - p_2| = \mathsf{Adv}^{\mathsf{wi}}_{\Phi,\mathcal{E}}(\lambda)$, using the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ as follows.

$\mathcal{E}_1(crs)$**:** First, $\mathcal{E}_1$ generates $ck \leftarrow \mathbf{CKG}(1^\lambda)$ and $(sigk^*, vk^*) \leftarrow \mathbf{SKG}(1^\lambda)$. Then, $\mathcal{E}_1$ samples $r \leftarrow \mathcal{R}_\Omega$ and computes $c \leftarrow \mathbf{Commit}(ck, vk^*; r)$. Next, $\mathcal{E}_1$ sets $crs' := (crs, c)$ and runs $\mathcal{A}_1(crs')$. When $\mathcal{A}_1$ outputs (x', w', st_1) and terminates, $\mathcal{E}_1$ sets $x := (x', ck, vk^*, c)$, $w_0 := (w', \perp)$, and $w_1 := (\perp, r)$. Then, $\mathcal{E}_1$ returns (x, w_0, w_1) and st'_1 including all of the information known to $\mathcal{E}_1$ to its experiment, and terminates.

$\mathcal{E}_2(\pi, \mathsf{st}'_1)$**:** First, $\mathcal{E}_2$ computes $\sigma \leftarrow \mathbf{Sign}(sigk, (x', \pi))$. Then, $\mathcal{E}_2$ sets $\pi' := (\pi, \sigma, vk^*)$ and runs $\mathcal{A}_2(\pi', \mathsf{st}_1)$. When $\mathcal{A}_2$ outputs a bit $b' \in \{0, 1\}$ and terminates, $\mathcal{E}_2$ returns b' to its experiment and terminates.

We let b be the challenge bit for $\mathcal{E}$ in its experiment. When $b = 0$, we can see that $\mathcal{E}$ perfectly simulates Exp_1 for $\mathcal{A}$. This ensures that when $b = 0$, the probability that $\mathcal{E}$ outputs 1 holds is exactly the same as the probability that $\mathcal{A}$ outputs 1 in Exp_1. On the other hand, when $b = 1$, $\mathcal{E}$ perfectly simulates Exp_2 for $\mathcal{A}$. This ensures that when $b = 0$, the probability that $\mathcal{E}$ outputs 1 is exactly the same as the probability that $\mathcal{A}$ outputs 1 in Exp_2. Therefore, we have $\mathsf{Adv}^{\mathsf{wi}}_{\Phi,\mathcal{E}}(\lambda) = |\Pr[b' = 1|b = 0] - \Pr[b' = 1|b = 1]| = |p_1 - p_2|$. □ **(Lemma 4)**

Putting everything together, we obtain $\mathsf{Adv}^{\mathsf{zk}}_{\Phi',\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{hide}}_{\Omega,\mathcal{D}}(\lambda) + \mathsf{Adv}^{\mathsf{wi}}_{\Phi,\mathcal{E}}(\lambda)$. Since Ω satisfies computationally hiding and Φ satisfies witness indistinguishability, for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{zk}}_{\Phi',\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$ holds. Therefore, Φ' satisfies zero-knowledge. □ **(Theorem 3)**

4 Construction of RNC−CCA Secure RNCE

In this section, we show that our generic construction of RNC−CCA secure RNCE with the plaintext space $\{0, 1\}$. Firstly, in Sect. 4.1, we describe our generic construction. Then, in Sect. 4.2, we give a security proof for it.

```
$\mathbf{KG}'(1^\lambda)$ :
  $\alpha \leftarrow \{0,1\}$
  $(pk_0, sk_0) \leftarrow \mathbf{KG}(1^\lambda)$
  $(pk_1, sk_1) \leftarrow \mathbf{KG}(1^\lambda)$
  $(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda)$
  $pk := (pk_0, pk_1, crs)$
  $sk := (\alpha, sk_\alpha, vsk)$
  Return $(pk, sk)$
```

```
$\mathbf{Enc}'(pk, m)$ :
  $(r_0, r_1) \leftarrow (\mathcal{R}_\Pi)^2$
  $c_0 \leftarrow \mathbf{Enc}(pk_0, m; r_0)$
  $c_1 \leftarrow \mathbf{Enc}(pk_1, m; r_1)$
  $x := (pk_0, pk_1, c_0, c_1)$
  $w := (m, r_0, r_1)$
  $\pi \leftarrow \mathbf{Prove}(crs, x, w)$
  Return $c := (c_0, c_1, \pi)$
```

```
$\mathbf{Dec}'(pk, sk, c)$ :
  $x := (pk_0, pk_1, c_0, c_1)$
  If $\mathbf{Verify}(crs, vsk, x, \pi) = 1$
  then
    $m \leftarrow \mathbf{Dec}(pk_\alpha, sk_\alpha, c_\alpha)$
    Return $m$
  else Return $\perp$
```

```
$\mathbf{FKG}'(1^\lambda)$ :
  $\alpha \leftarrow \{0,1\}$
  $(pk_0, sk_0) \leftarrow \mathbf{KG}(1^\lambda)$
  $(pk_1, sk_1) \leftarrow \mathbf{KG}(1^\lambda)$
  $(crs, vsk, tk)$
          $\leftarrow \mathbf{SimCRS}(1^\lambda)$
  $pk := (pk_0, pk_1, crs)$
  $td := (\alpha, sk_0, sk_1, vsk, tk)$
  Return $(pk, td)$
```

```
$\mathbf{Fake}'(pk, td)$ :
  $c_\alpha \leftarrow \mathbf{Enc}(pk_\alpha, 0)$
  $c_{1\oplus\alpha} \leftarrow \mathbf{Enc}(pk_{1\oplus\alpha}, 1)$
  $x := (pk_0, pk_1, c_0, c_1)$
  $\pi \leftarrow \mathbf{SimPrv}(tk, x)$
  Return $\widetilde{c} := (c_0, c_1, \pi)$
```

```
$\mathbf{Open}'(pk, td, \widetilde{c}, m)$ :
  $\widetilde{sk} := (\alpha \oplus m,$
            $sk_{\alpha\oplus m}, vsk)$
  Return $\widetilde{sk}$
```

```
$\mathbf{FDec}'(pk, td, c)$ :
  $x := (pk_0, pk_1, c_0, c_1)$
  If $\mathbf{Verify}(crs, vsk, x, \pi) = 1$
  then
    $m \leftarrow \mathbf{Dec}(pk_0, sk_0, c_0)$
    Return $m$
  else Return $\perp$
```

Fig. 2. Construction of RNC−CCA secure RNCE Π'.

4.1 Description

In this section, we formally describe our generic construction of RNC-CCA secure RNCE with the plaintext space $\{0,1\}$. Let $\Pi = (\mathbf{KG}, \mathbf{Enc}, \mathbf{Dec})$ be a PKE scheme with the plaintext space $\{0,1\}$ and $\mathcal{R}_\pi$ a randomness space for the encryption algorithm **Enc**. Let $\Phi = (\mathbf{CRSGen}, \mathbf{Prove}, \mathbf{Verify}, \mathbf{SimCRS}, \mathbf{SimPrv})$ be a DV-NIZK argument for

$$\mathcal{L}_{eq} := \Big\{(pk_0, pk_1, c_0, c_1) \mid \exists (m, r_0, r_1) \in \{0,1\} \times (\mathcal{R}_\Pi)^2 \text{ s.t. } (c_0 = \mathbf{Enc}(pk_0, m; r_0)) \wedge (c_1 = \mathbf{Enc}(pk_1, m; r_1))\Big\}.$$

Then, we use them to construct our RNCE scheme $\Pi' = (\mathbf{KG}', \mathbf{Enc}', \mathbf{Dec}', \mathbf{FKG}', \mathbf{Fake}', \mathbf{Open}', \mathbf{FDec}')$ with the plaintext space $\{0,1\}$ as described in Fig. 2. We note that the correctness of our RNCE scheme holds due to the correctness of Π and Φ.

How to expand the plaintext space of our generic construction. In the above, we only give the construction whose plaintext space is $\{0,1\}$. However, we can expand the plaintext space by using our single-bit construction in a parallel way except for the generation of a proof of a DV-NIZK argument. More concretely, if we encrypt an ℓ-bit plaintext $m = m_1 \| \cdots \| m_\ell$, the procedure is as follows.

Firstly, we generate a public key $pk = ((pk_0^i, pk_1^i)_{i\in[\ell]}, crs)$ and a secret key $sk = (\alpha_i, sk_{\alpha_i}^i, vsk)_{i\in[\ell]}$, where $\alpha_1, \cdots, \alpha_\ell \leftarrow \{0,1\}$, $(pk_v^i, sk_v^i) \leftarrow \mathbf{KG}(1^\lambda)$ for all $(i, v) \in [\ell] \times \{0,1\}$, and *crs* (resp., *vsk*) denotes a CRS (resp., a secret verification key) of a DV-NIZK argument. Next, we compute a ciphertext $c = ((c_0^i)_{i\in[\ell]}, (c_1^i)_{i\in[\ell]}, \pi)$, where $c_v^i \leftarrow \mathbf{Enc}(pk_v^i, m_i)$ for all $(i, v) \in [\ell] \times \{0,1\}$ and π is a proof proving that, for each $i \in [\ell]$, the ciphertexts c_0^i and c_1^i encrypt the same plaintext $m_i \in \{0,1\}$. Similarly, for the other procedures, we execute one-bit version algorithms in a parallel way for all $i \in [\ell]$ except for the procedure of a DV-NIZK argument. See the full version of the paper for the details.

4.2 Security Proof

In this section, we show the following theorem.

Theorem 4. *If Π is an* IND−CPA *secure PKE scheme and Φ satisfies one-time simulation soundness and zero-knowledge, then Π' is* RNC−CCA *secure.*

Proof of Theorem 4. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ be a PPT adversary that attacks the RNC−CCA security of Π'. We introduce the following experiments $\{\mathsf{Exp}_i\}_{i=0}^5$.

Exp_0 : Exp_0 is exactly the same as $\mathsf{Exp}_{\Pi',\mathcal{A}}^{\mathsf{rnc-real}}(\lambda)$. The detailed description is as follows.

1. First, Exp_0 samples $\alpha \leftarrow \{0,1\}$ and computes $(pk_0, sk_0) \leftarrow \mathbf{KG}(1^\lambda)$, $(pk_1, sk_1) \leftarrow \mathbf{KG}(1^\lambda)$, and $(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda)$. Next, Exp_0 sets $pk := (pk_0, pk_1, crs)$ and $sk := (\alpha, sk_\alpha, vsk)$, and runs $\mathcal{A}_1(pk)$. When $\mathcal{A}_1$ makes a decryption query $c = (c_0, c_1, \pi)$, Exp_0 checks whether $\mathbf{Verify}(crs, vsk, (pk_0, pk_1, c_0, c_1), \pi) = 1$ or not. If this condition holds, Exp_0 computes $m \leftarrow \mathbf{Dec}(pk_\alpha, sk_\alpha, c_\alpha)$ and returns m to $\mathcal{A}_1$. Otherwise, Exp_0 returns $\perp$ to $\mathcal{A}_1$.
2. When $\mathcal{A}_1$ outputs (m^*, st_1) and terminates, Exp_0 computes the challenge ciphertext c^* as follows. First, Exp_0 samples $(r_0^*, r_1^*) \leftarrow (\mathcal{R}_\Pi)^2$ and computes $c_0^* \leftarrow \mathbf{Enc}(pk_0, m^*; r_0^*)$, $c_1^* \leftarrow \mathbf{Enc}(pk_1, m^*; r_1^*)$, and $\pi^* \leftarrow \mathbf{Prove}(crs, (pk_0, pk_1, c_0^*, c_1^*), (m^*, r_0^*, r_1^*))$. Next, Exp_0 sets $c^* = (c_0^*, c_1^*, \pi^*)$ and runs $\mathcal{A}_2(c^*, \mathsf{st}_1)$. When $\mathcal{A}_2$ makes a decryption query c, Exp_0 answers in the same way as above.
3. When $\mathcal{A}_2$ outputs state information st_2 and terminates, Exp_0 runs $\mathcal{A}_3(sk, \mathsf{st}_2)$.
4. When $\mathcal{A}_3$ outputs a bit b' and terminates, Exp_0 outputs b'.

Exp_1 : Exp_1 is identical to Exp_0 except for the following change. When computing the challenge ciphertext c^*, the common reference string *crs* is generated by executing $(crs, vsk, tk) \leftarrow \mathbf{SimCRS}(1^\lambda)$ instead of $(crs, vsk) \leftarrow \mathbf{CRSGen}(1^\lambda)$. Moreover, Exp_1 generates a simulated proof $\pi^* \leftarrow \mathbf{SimPrv}(tk, (pk_0, pk_1, c_0^*, c_1^*))$ instead of $\pi^* \leftarrow \mathbf{Prove}(crs, (pk_0, pk_1, c_0^*, c_1^*), (m^*, r_0^*, r_1^*))$.

Exp_2 : Exp_2 is identical to Exp_1 except that when computing the challenge ciphertext c^*, Exp_2 computes $c_{1\oplus\alpha}^* \leftarrow \mathbf{Enc}(pk_{1\oplus\alpha}, 1 \oplus m^*; r_{1\oplus\alpha}^*)$ instead of $c_{1\oplus\alpha}^* \leftarrow \mathbf{Enc}(pk_{1\oplus\alpha}, m^*; r_{1\oplus\alpha}^*)$.

Exp_3 : Exp_3 is identical to Exp_2 except that when responding to a decryption query $c = (c_0, c_1, \pi)$, if $\mathbf{Verify}(crs, vsk, (pk_0, pk_1, c_0, c_1), \pi) = 1$, Exp_3 answers $m \leftarrow \mathbf{Dec}(pk_0, sk_0, c_0)$ instead of $m \leftarrow \mathbf{Dec}(pk_\alpha, sk_\alpha, c_\alpha)$. Note that the decryption procedure in Exp_3 is exactly the same as $\mathbf{FDec}'(pk, td, c)$.

Exp_4 : Exp_4 is identical to Exp_3 except that $\alpha \oplus m^*$ is used instead of α. That is, when computing the challenge ciphertext c^*, Exp_4 computes c_0^* and c_1^* by $c^*_{\alpha\oplus m^*} \leftarrow \mathbf{Enc}(pk_{\alpha\oplus m^*}, m^*)$ and $c^*_{\alpha\oplus(1\oplus m^*)} \leftarrow \mathbf{Enc}(pk_{\alpha\oplus(1\oplus m^*)}, 1 \oplus m^*)$. Moreover, Exp_4 gives the secret key $sk = (\alpha \oplus m^*, sk_{\alpha\oplus m^*}, vsk)$ to $\mathcal{A}_3$ instead of $sk = (\alpha, sk_\alpha)$.

Exp_5 : Exp_5 is exactly the same as $\mathsf{Exp}^{\mathsf{rnc-sim}}_{\Pi',\mathcal{A}}(\lambda)$.

We let $p_i := \Pr[\mathsf{Exp}_i(\lambda) = 1]$ for all $i \in [0, 5]$. Then, we have $\mathsf{Adv}^{\mathsf{rnc-cca}}_{\Pi',\mathcal{A}}(\lambda) = |\Pr[\mathsf{Exp}^{\mathsf{rnc-real}}_{\Pi',\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{rnc-sim}}_{\Pi',\mathcal{A}}(\lambda) = 1]| = |p_0 - p_5| \le \sum_{i=0}^{4} |p_i - p_{i+1}|$. It remains to show how each $|p_i - p_{i+1}|$ is upper-bounded. To this end, we will show the following lemmata.

Lemma 5. *There exists a PPT adversary $\mathcal{E} = (\mathcal{E}_1, \mathcal{E}_2)$ against the zero-knowledge of Φ such that $|p_0 - p_1| = \mathsf{Adv}^{\mathsf{zk}}_{\Phi,\mathcal{E}}(\lambda)$.*

Lemma 6. *There exists a PPT adversary $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2)$ against the* IND−CPA *security of Π such that $|p_1 - p_2| = \mathsf{Adv}^{\mathsf{ind-cpa}}_{\Pi,\mathcal{F}}(\lambda)$.*

Lemma 7. *There exists a PPT adversary $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2)$ against the one-time simulation soundness of Φ such that $|p_2 - p_3| \le \mathsf{Adv}^{\mathsf{ot-ss}}_{\Phi,\mathcal{G}}(\lambda)$.*

Lemma 8. $|p_3 - p_4| = 0$ *holds.*

Lemma 9. $|p_4 - p_5| = 0$ *holds.*

As mentioned in Sect. 1.2, compared to the previous work [10], the most technically obscure part is Lemma 7 using the one-time simulation soundness of a DV-NIZK argument, and thus we show only it here due to the space limitation. We will show the rest lemmata formally in the full version of the paper.

Proof of Lemma 7. For $i \in \{2, 3\}$, we let $\mathbf{Bad}_i$ be the event that $\mathcal{A}_2$ makes a decryption query $c = (c_0, c_1, \pi)$ satisfying $(\mathbf{Dec}(pk_0, sk_0, c_0) \ne \mathbf{Dec}(pk_1, sk_1, c_1)) \wedge (\mathbf{Verify}(crs, vsk, (pk_0, pk_1, c_0, c_1), \pi) = 1)$ in Exp_i. (We call such a decryption query a *bad decryption query*.) Exp_2 proceeds identically to Exp_3 unless $\mathbf{Bad}_2$ happens. Therefore, the inequality $|p_2 - p_3| \le \Pr[\mathbf{Bad}_2] = \Pr[\mathbf{Bad}_3]$ holds. In the following, we show that one can construct a PPT adversary $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2)$ that attacks the one-time simulation soundness of Φ so that $\Pr[\mathbf{Bad}_2] = \mathsf{Adv}^{\mathsf{ot-ss}}_{\Phi,\mathcal{G}}(\lambda)$, using the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$.

$\mathcal{G}_1(crs)$ **:** First, $\mathcal{G}_1$ samples $\alpha \leftarrow \{0, 1\}$ and computes $(pk_0, sk_0) \leftarrow \mathbf{KG}(1^\lambda)$ and $(pk_1, sk_1) \leftarrow \mathbf{KG}(1^\lambda)$. Next, $\mathcal{G}_1$ sets $pk := (pk_0, pk_1, crs)$ and runs $\mathcal{A}_1(pk)$. When $\mathcal{A}_1$ makes a decryption query $c = (c_0, c_1, \pi)$, $\mathcal{G}_1$ makes a verification query $((pk_0, pk_1, c_0, c_1), \pi)$ to its experiment. Upon receiving a verification

result $v \in \{0,1\}$, $\mathcal{G}_1$ checks whether $v = 1$ or not. If this is the case, then $\mathcal{G}_1$ computes $m \leftarrow \mathbf{Dec}(pk_\alpha, sk_\alpha, c_\alpha)$ and returns m to $\mathcal{A}_1$. Otherwise, $\mathcal{G}_1$ returns $\perp$ to $\mathcal{A}_1$.
When $\mathcal{A}_1$ outputs the challenge plaintext m^* and state information st_1, and terminates, $\mathcal{G}_1$ computes $c^*_\alpha \leftarrow \mathbf{Enc}(pk_\alpha, m^*)$ and $c^*_{1\oplus\alpha} \leftarrow \mathbf{Enc}(pk_{1\oplus\alpha}, 1 \oplus m^*)$. Finally, $\mathcal{G}_1$ sets st'_1 as all the information known to it, returns $((pk_0, pk_1, c^*_0, c^*_1), \mathsf{st}'_1)$ to its experiment, and terminates.

$\mathcal{G}_2(\pi^*, \mathsf{st}'_1)$ **:** First, $\mathcal{G}_2$ sets $c^* := (c^*_0, c^*_1, \pi^*)$ and runs $\mathcal{A}_2(c^*, \mathsf{st}_1)$. When $\mathcal{A}_2$ makes a decryption query c, $\mathcal{G}_2$ parses $c := (c_0, c_1, \pi)$. Then, $\mathcal{G}_2$ makes a verification query $((pk_0, pk_1, c_0, c_1), \pi)$ to its experiment. If the verification result is 0, then $\mathcal{G}_2$ returns $\perp$ to $\mathcal{A}_2$. If the verification result is 1, then $\mathcal{G}_2$ checks whether $\mathbf{Dec}(pk_0, sk_0, c_0) \neq \mathbf{Dec}(pk_1, sk_1, c_1)$ or not. If this is the case, $\mathcal{G}_2$ returns $((pk_0, pk_1, c_0, c_1), \pi)$ to its experiment and terminates. Otherwise, $\mathcal{G}_2$ computes $m \leftarrow \mathbf{Dec}(pk_\alpha, sk_\alpha, c_\alpha)$ and returns m to $\mathcal{A}_2$. When $\mathcal{A}_2$ outputs state information st_2 and terminates, $\mathcal{G}_2$ gives up and terminates.

From the above construction of $\mathcal{G}$, it is easy to see that $\mathcal{G}$ perfectly simulates the experiment Exp_2 for $\mathcal{A}$. Here, the success condition of $\mathcal{G}$ is to output a pair of a statement and a proof (x, π) satisfying $((x^*, \pi^*) \neq (x, \pi)) \wedge (\mathbf{Verify}(crs, vsk, x, \pi) = 1) \wedge (x \notin \mathcal{L}_{eq})$, where $x^* = (pk_0, pk_1, c^*_0, c^*_1)$ and $x = (pk_0, pk_1, c_0, c_1)$. If $\mathcal{A}_2$ makes a bad decryption query $c = (c_0, c_1, \pi)$, then $\mathbf{Dec}(pk_0, sk_0, c_0) \neq \mathbf{Dec}(pk_1, sk_1, c_1)$ and $\mathbf{Verify}(crs, vsk, x, \pi) = 1$. Thus, we can see that $x \notin \mathcal{L}_{eq}$ holds now due to the correctness of Π.

Moreover, due to the condition of decryption queries by $\mathcal{A}_2$, we have $(c^*_0, c^*_1, \pi^*) = c^* \neq c = (c_0, c_1, \pi)$. That is, we have $(x^*, \pi^*) \neq (x, \pi)$. Thus, when $\mathcal{A}_2$ makes a bad decryption query c, $\mathcal{G}$ achieves its success condition by returning (x, π) to its experiment. We note that $\mathcal{G}$ can detect that the event $\mathbf{Bad}_2$ occurs because $\mathcal{G}$ has both of the secret keys sk_0 and sk_1, and can make a verification query (x, π) to its experiment. From the above arguments, the probability that $\mathcal{A}_2$ makes a bad decryption query is exactly the same as the probability that $\mathcal{G}$ breaks the one-time simulation soundness of Φ. Therefore, we have $\Pr[\mathbf{Bad}_2] = \mathsf{Adv}^{\mathsf{ot-ss}}_{\Phi,\mathcal{G}}(\lambda)$, which in turn implies $|p_2 - p_3| \leq \mathsf{Adv}^{\mathsf{ot-ss}}_{\Phi,\mathcal{G}}(\lambda)$. $\square$ **(Lemma 7)**

Putting everything together, we obtain $\mathsf{Adv}^{\mathsf{rnc-cca}}_{\Pi',\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{zk}}_{\Phi,\mathcal{E}}(\lambda) + \mathsf{Adv}^{\mathsf{ind-cpa}}_{\Pi,\mathcal{F}}(\lambda) + \mathsf{Adv}^{\mathsf{ot-ss}}_{\Phi,\mathcal{G}}(\lambda)$. Since Π is IND−CPA secure and Φ satisfies one-time simulation soundness and zero-knowledge, for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{rnc-cca}}_{\Pi',\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$ holds. Therefore, Π' satisfies RNC−CCA security. $\square$ **(Theorem 4)**

Acknowledgement. A part of this work was supported by NTT Secure Platform Laboratories, JST OPERA JPMJOP1612, JST CREST JPMJCR14D6, JSPS KAKENHI JP16H01705, JP17H01695, JP20J14338.

References

1. Alekhnovich, M.: More on average case vs approximation complexity. In: 44th FOCS, pp. 298–307 (2003)

2. Bellare, M., Dowsley, R., Waters, B., Yilek, S.: Standard security does not imply security against selective-opening. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 645–662. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_38
3. Bellare, M., Hofheinz, D., Yilek, S.: Possibility and impossibility results for encryption and commitment secure under selective opening. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 1–35. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01001-9_1
4. Bellare, M., Yilek, S.: Encryption schemes secure under selective opening attack. Cryptology ePrint Archive, Report 2009/101 (2009)
5. Canetti, R., Halevi, S., Katz, J.: Adaptively-secure, non-interactive public-key encryption. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 150–168. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30576-7_9
6. Couteau, G., Hofheinz, D.: Designated-verifier pseudorandom generators, and their applications. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11477, pp. 562–592. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17656-3_20
7. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (extended abstract). In: 23rd ACM STOC, pp. 542–552 (2020)
8. Elkind, E., Sahai, A.: A unified methodology for constructing public-key encryption schemes secure against adaptive chosen-ciphertext attack. Cryptology ePrint Archive, Report 2002/042 (2002)
9. Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. **28**(2), 270–299 (1984)
10. Hara, K., Kitagawa, F., Matsuda, T., Hanaoka, G., Tanaka, K.: Simulation-based receiver selective opening CCA secure PKE from standard computational assumptions. In: Catalano, D., De Prisco, R. (eds.) SCN 2018. LNCS, vol. 11035, pp. 140–159. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98113-0_8
11. Hara, K., Kitagawa, F., Matsuda, T., Hanaoka, G., Tanaka, K.: Simulation-based receiver selective opening CCA secure PKE from standard computational assumptions. Theor. Comput. Sci. **795**, 570–597 (2019)
12. Huang, Z., Lai, J., Chen, W., Au, M.H., Peng, Z., Li, J.: Simulation-based selective opening security for receivers under chosen-ciphertext attacks. Des. Codes Cryptogr. **87**(6), 1345–1371 (2019)
13. Haralambiev, K., Jager, T., Kiltz, E., Shoup, V.: Simple and efficient public-key encryption from computational Diffie-Hellman in the standard model. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 1–18. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_1
14. Hazay, C., Patra, A., Warinschi, B.: Selective opening security for receivers. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 443–469. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_19
15. Jia, D., Lu, X., Li, B.: Receiver selective opening security from indistinguishability obfuscation. In: Dunkelman, O., Sanadhya, S.K. (eds.) INDOCRYPT 2016. LNCS, vol. 10095, pp. 393–410. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-49890-4_22
16. Jia, D., Lu, X., Li, B.: Constructions secure against receiver selective opening and chosen ciphertext attacks. In: Handschuh, H. (ed.) CT-RSA 2017. LNCS, vol. 10159, pp. 417–431. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-52153-4_24

17. Katsumata, S., Nishimaki, R., Yamada, S., Yamakawa, T.: Designated verifier/prover and preprocessing NIZKs from Diffie-Hellman assumptions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11477, pp. 622–651. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17656-3_22
18. Kitagawa, F., Matsuda, T.: CPA-to-CCA transformation for KDM security. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019. LNCS, vol. 11892, pp. 118–148. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36033-7_5
19. Lindell, Y.: A simpler construction of CCA2-secure public-key encryption under general assumptions. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 241–254. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_15
20. Lombardi, A., Quach, W., Rothblum, R.D., Wichs, D., Wu, D.J.: New constructions of reusable designated-verifier NIZKs. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11694, pp. 670–700. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26954-8_22
21. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd ACM STOC, pp. 427–437 (1990)
22. Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (Plain) learning with errors. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11692, pp. 89–114. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_4
23. Quach, W., Rothblum, R.D., Wichs, D.: Reusable designated-verifier NIZKs for all NP from CDH. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11477, pp. 593–621. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17656-3_21
24. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: 37th ACM STOC, pp. 84–93 (2009)
25. Yu, Yu., Zhang, J.: Cryptography with auxiliary input and trapdoor from constant-noise LPN. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 214–243. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_9