Keyed-Fully Homomorphic Encryption without Indistinguishability Obfuscation[*]

Shingo Sato[†] Keita Emura[‡] Atsushi Takayasu[§]

September 20, 2023

Abstract

(Fully) homomorphic encryption ((F)HE) allows users to publicly evaluate circuits on encrypted data. Although publicly homomorphic evaluation property has various applications, (F)HE cannot achieve security against chosen ciphertext attacks (CCA2) due to its nature. To achieve both the CCA2 security and homomorphic evaluation property, Emura et al. (PKC 2013) introduced keyed-homomorphic public key encryption (KH-PKE) and formalized its security denoted by $\mathsf{KH\text{-}CCA}$ security. KH-PKE has a homomorphic evaluation key that enables users to perform homomorphic operations. Intuitively, KH-PKE achieves the CCA2 security unless adversaries have a homomorphic evaluation key. Although Lai et al. (PKC 2016) proposed the first keyed-fully homomorphic encryption (keyed-FHE) scheme, its security relies on the indistinguishability obfuscation (iO), and this scheme satisfies only a weak variant of $\mathsf{KH\text{-}CCA}$ security. Here, we propose a generic construction of a $\mathsf{KH\text{-}CCA}$ secure keyed-FHE scheme from an FHE scheme secure against non-adaptive chosen ciphertext attack (CCA1) and a strong dual-system simulation-sound non-interactive zero-knowledge (strong DSS-NIZK) argument system by using the Naor-Yung paradigm. We show that there are existing strong DSS-NIZK systems and IND-CCA1 secure FHE schemes that are suitable for our generic construction. This shows that there exists a keyed-FHE scheme from simpler primitives than iO.

1 Introduction

1.1 Background

Homomorphic encryption (HE) allows users to convert encryptions of messages $\mathsf{m}_1, \dots, \mathsf{m}_\ell$ into an encryption of $\mathsf{C}(\mathsf{m}_1, \dots, \mathsf{m}_\ell)$ publicly for some circuit C. In particular, *fully homomorphic encryption (FHE)* can be used to handle arbitrary circuits. The publicly homomorphic evaluation property is applied to various applications. For example, suppose (evaluated) encryptions of private data are stored in a remote server, delegating computations on the encrypted data to the server without revealing the private data is possible. Thus, users leverage the results of computations on other devices without compromising data privacy.

Since Gentry proposed the first FHE scheme [26], the research area has gained widespread attention and many schemes have been proposed (e.g., FHE schemes [6, 8–12, 27, 44], identity-based FHE (IBFHE) schemes [17, 27], and attribute-based FHE schemes [7, 27]), where most schemes are secure under the learning with errors (LWE) assumption.

[*] An extended abstract appeared at ACNS 2022. This is the full version.

[†] Yokohama National University, Japan. sato-shingo-zk@ynu.ac.jp

[‡] Kanazawa University, Japan. k-emura@se.kanazawa-u.ac.jp

[§] The University of Tokyo, Japan. takayasu-a@g.ecc.u-tokyo.ac.jp

Although the public evaluation property is useful, one downside is that HE schemes are vulnerable against adaptive chosen ciphertext attacks (CCA). (In this paper, we use IND-CCA2 or IND-CCA, IND-CCA1, and IND-CPA as indistinguishability against adaptive chosen ciphertext attacks, non-adaptive chosen ciphertext (i.e., lunchtime) attacks, and chosen-plaintext attacks, respectively). Therefore, IND-CCA1 secure FHE schemes have been proposed. For example, Canetti et al. [12] proposed a generic construction of IND-CCA1 secure FHE from the LWE assumption or a zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK) [4, 5] and IND-CPA secure FHE. However, IND-CCA1 security can be inadequate for FHE since Loftus et al. [36] showed that an IND-CCA1 secure FHE scheme is vulnerable against ciphertext validity attacks.

To achieve both CCA2-like security and homomorphic evaluation property, Emura et al. [22, 23] introduced *keyed-homomorphic public-key encryption (KH-PKE)*. Contrary to traditional HE, the homomorphic evaluation property of KH-PKE is not public. Specifically, KH-PKE has a homomorphic evaluation key. Thus, only users with the homomorphic evaluation key can perform homomorphic operations. Due to its nature, KH-PKE can achieve CCA2-like security.[1] Suppose adversaries do not have the homomorphic evaluation key, then, KH-PKE satisfies the IND-CCA2 security. Moreover, KH-PKE satisfies a stronger security than that of HE even if adversaries receive a homomorphic evaluation key. Suppose adversaries receive the homomorphic evaluation key before the challenge query, then the strongest security that KH-PKE can satisfy is the IND-CCA1 security as the case of HE. In contrast, KH-PKE can satisfy a stronger security than the IND-CCA1 security after the challenge query since adversaries continue making decryption queries until they receive the homomorphic evaluation key. Moreover, KH-PKE is secure against key recovery and ciphertext validity attacks [21].

Emura et al. [23] proposed the security notion of KH-PKE, called KH-CCA security, and gave the KH-PKE schemes under the decisional Diffie-Hellman (DDH) assumption or the decisional composite residuosity (DCR) assumption. Though their security proofs contain bugs, they have been corrected in [22]. Libert et al. [35] proposed the first KH-PKE schemes KH-CCA secure using the Decision Linear (DLIN) assumption or the symmetric external Diffie-Hellman (SXDH) assumption. Jutla and Roy [32] proposed a KH-PKE scheme based on SXDH assumption. All KH-PKE schemes support either multiplicative or additive homomorphisms. As another direction, Maeda and Nuida [38] proposed a two-level KH-PKE scheme based on the SXDH assumption, which supports one multiplication and any number of additions. Shinoki and Nuida demonstrated the condition when more than two ciphertexts can be evaluated simultaneously [43].

Lai et al. [33] proposed the first *keyed-fully homomorphic encryption (keyed-FHE)*[2]scheme, which is secure under lattice assumptions and the indistinguishability obfuscation (iO) [1]. However, known candidates of iO [1] remain arguable.Therefore, constructing keyed-FHE schemes without iO has to be an interesting open problem. We remark that the keyed-FHE scheme of [33] satisfies only a weaker security than KH-CCA security. In the security game considered in [33], it is supposed that an adversary receives a homomorphic evaluation key before the challenge phase. In this case, the adversary is prohibited to access the decryption oracle.

1.2 Contribution

In this work, we propose a generic construction of keyed-FHE without iO. Our construction is based on the Naor-Yung paradigm [40, 41] to achieve KH-CCA security. The building blocks of our construction are IND-CCA1 secure FHE and a strong dual-system unbounded simulation-sound

[1] Although Desmedt et al. [20] proposed a HE scheme with a designated evaluation called controlled HE, no CCA security was considered unlike the KH-PKE.

[2] In this paper, we consider keyed-FHE in a public key setting.

NIZK (strong DSS-NIZK) [32]. There exist IND-CCA1 secure FHE schemes [12]. As a result, we can obtain a keyed-FHE instantiation constructed from simpler primitives than iO and its security is based on a knowledge assumption (this is concretely discussed in Section 5). We remark that we have to construct a desired strong DSS-NIZK system since there is no existing DSS-NIZK for IND-CCA1 FHE ciphertexts.

For this purpose, we propose a generic construction of strong DSS-NIZK for IND-CCA1 FHE ciphertexts from a smooth projective hash proof system (PHPS) and an unbounded simulation-sound NIZK system. There are statistically secure smooth PHPS [3] and unbounded simulation-sound NIZK schemes [13,29,34] whose security depends on lattice assumptions or the security of the commitment schemes used in [13, 29]. We remark that for instantiating our generic construction of strong DSS-NIZK for IND-CCA1 FHE ciphertexts, it is required that the IND-CCA1 secure FHE schemes are publicly verifiable (but there exists such a scheme [12]) though our keyed-FHE construction does not require this property for the underlying FHE schemes.

To sum up, we obtain the first keyed-FHE scheme without iO, and its security is based on a knowledge assumption[3]. Furthermore, another advantage of our result is that our keyed-FHE scheme satisfies stronger security (i.e., KH-CCA security) than the existing one [33].

1.3 Technical Overview

We give a brief overview of our results. Since Lai et al. [33] constructed the keyed-FHE scheme using iO, the most convincing way to achieve the goal is to remove the iO from the construction. However, completing the task seems technically difficult because the correctness and security of most existing KH-PKE schemes [22, 23, 35, 38, 43] extremely depend on the properties of cryptographic primitives based on the DDH, DCR, or pairing-based assumptions, and it is unclear to construct these primitives based on the computational assumptions used in the existing FHE schemes. Thus, we focus on the Jutla and Roy's KH-PKE scheme [32] by providing a strong variant of NIZK suitable for constructing a keyed-FHE scheme. Their construction used an ElGamal encryption scheme and a stronger version of the dual-system unbounded simulation-sound NIZK (DSS-NIZK) for the Diffie-Hellman language. Due to the nature of one-time simulation-sound NIZK for the Diffie-Hellman language, their construction satisfies KH-CCA security as noted in [32]. Therefore, the remaining task to prove the KH-CCA security is how to simulate the homomorphic key reveal oracle (RevHK) and how to prove the IND-CCA1 security even after the RevHK query. Here, the properties of strong DSS-NIZK resolve the problems. The homomorphic evaluation key of the KH-PKE scheme is a trapdoor of the strong DSS-NIZK, which is a secret value used in the zero-knowledge simulator of (strong) DSS-NIZK. In particular, the trapdoor leakage resilience of strong DSS-NIZK ensure IND-CCA1 security even after the trapdoor is revealed to an adversary. This is because the proof generated by a (DSS-)NIZK system contains secret information such as a message, and there is a possibility that an adversary obtains such secret information after the RevHK oracle access. Hence, that property of strong DSS-NIZK is necessary to guarantee the KH-CCA security. To satisfy the required properties, Jutla and Roy constructed the strong DSS-NIZK scheme for the Diffie-Hellman language using quasi-adaptive NIZK for the same language [31] and an hash proof system (HPS) [19] that is smooth projective and universal$_2$.

Using a similar approach, we construct the keyed-FHE without iO by replacing (a variant of) the ElGamal encryption scheme with FHE schemes. For this purpose, we have to overcome some issues.

[3] Note that even if an IND-CPA secure FHE scheme under (a variant of) the approximate GCD assumption (e.g., [14, 18, 44]) is employed to construct an IND-CCA1 secure FHE scheme, our generic construction gives no keyed-FHE scheme based solely on that assumption because there is no existing HPS for approximate GCD-based ciphertexts.

First, the Jutla and Roy's KH-PKE scheme used strong DSS-NIZK for the Diffie-Hellman language that is not suitable for FHE. Therefore, we construct strong DSS-NIZK for another language that handles FHE ciphertexts. Here, we observe whether we can construct strong DSS-NIZK for FHE ciphertexts following a similar approach as Jutla and Roy. They used quasi-adaptive NIZK for the Diffie-Hellman language and an HPS [19] that is smooth projective and universal$_2$. In this step, an issue occurs since there is no known lattice-based universal$_2$ HPS. We construct the desired strong DSS-NIZK for FHE ciphertexts by replacing the universal$_2$ HPS of Jutla-Roy's DSS-NIZK with unbounded simulation-sound NIZK and modifying slightly their construction. Second, the Jutla and Roy's KH-PKE scheme satisfies $\mathsf{KH\text{-}CCA}$ security based on simulation-sound NIZK for the Diffie-Hellman language. That is, just replacing the ElGamal encryption scheme with FHE schemes does not work well, since their scheme is constructed in a semi-generic way. Here, we resolve this issue by employing the Naor-Yung paradigm [40, 41], so that our keyed-FHE scheme works correctly and achieves $\mathsf{KH\text{-}CCA}$ security. These modifications enable us to construct a keyed-FHE scheme without iO.

Therefore, this completes a brief overview of our generic keyed-FHE scheme.

1.4 Differences from the Proceedings Version

In the current version, we modify the proceedings version [42], as follows: The evaluation algorithm of our keyed-FHE scheme explicitly rerandomizes evaluated ciphertexts of the underlying FHE schemes by adding encryptions of 0, while this rerandomization procedure was not described in the proceedings version. Furthermore, the detailed proofs of Theorems 1 and 2 (the security proofs of our keyed-FHE scheme and DSS-NIZK system) are described in Sections 3.2.1 and 4.2, respectively. Meanwhile, in the proceedings version, we omitted these security proofs, due to the page-limitation. In addition, we describe a keyed-FHE scheme constructed from $\mathsf{IND\text{-}CPA}$ secure FHE, zk-SNARK, and strong DSS-NIZK, in A, while this was also omitted in the proceedings version, because of the page-limitation.

Here, we give a detailed explanation about the necessity of the rerandomization procedure in our scheme. In the proceedings version, we implicitly assumed that the evaluation algorithms (denoted by $\mathsf{Eval}_{F,1}$ and $\mathsf{Eval}_{F,2}$) of the underlying FHE of our keyed-FHE scheme were probabilistic. If $\mathsf{Eval}_{F,1}$ or $\mathsf{Eval}_{F,2}$ is deterministic, the security game Game_2 in the proof of Theorem 1 is distinguishable from the previous security game. Concretely, for the first and the second components (denoted by $\widehat{\mathsf{ct}}_1$ and $\widehat{\mathsf{ct}}_2$) of an evaluated ciphertext, an adversary can distinguish those games by comparing $\widehat{\mathsf{ct}}_1$ and $\widehat{\mathsf{ct}}_2$ received from the evaluation oracle, with these components computed by itself. Thus, we had to assume that both $\mathsf{Eval}_{F,1}$ and $\mathsf{Eval}_{F,2}$ were probabilistic. However, even though $\mathsf{Eval}_{F,1}$ or $\mathsf{Eval}_{F,2}$ is deterministic, it is possible to rerandomize $\widehat{\mathsf{ct}}_1$ and $\widehat{\mathsf{ct}}_2$ by using $\mathsf{Eval}_{F,1}$ and $\mathsf{Eval}_{F,2}$.

In order to clarify the procedure of our keyed-FHE scheme, we explicitly write that the evaluation algorithm of the keyed-FHE scheme (in Section 3) in the current version rerandomizes $\widehat{\mathsf{ct}}_1$ and $\widehat{\mathsf{ct}}_2$ even if $\mathsf{Eval}_{F,1}$ or $\mathsf{Eval}_{F,2}$ is deterministic.

1.5 Organization

The rest of this paper is organized as follows: In Section 2, we describe the notation used in this paper and definitions of (DSS-)NIZK, PHPS, and (keyed-)FHE. In Section 3, we propose our generic construction of keyed-FHE and prove its security. In Section 4, we present a generic construction of DSS-NIZKs from a smooth PHPS and an unbounded-simulation sound NIZK. In Section 5, we show that there exist existing primitives suitable for constructing keyed-FHE schemes without iO.

2 Preliminaries

In this section, we describe the notation used in this paper and the definitions of several cryptographic primitives.

Notation. For a positive integer n, let $[n] := \{1, 2, \ldots, n\}$. For n values $x_1, x_2, \ldots, x_n$ and a subset $I \subseteq [n]$ of indices, let $\{x_i\}_{i \in I}$ be a set of values whose indices are included in I, and let $(x_i)_{i \in I}$ be a sequence of values whose indices are included in I. Probabilistic polynomial-time is abbreviated as PPT. If a function $f : \mathbb{N} \to \mathbb{R}$ fulfills $f(\lambda) = o(\lambda^{-c})$ for every constant $c > 0$ and sufficiently large λ, then we say that f is negligible in λ and denoted by $f(\lambda) = \mathsf{negl}(\lambda)$. A probability is overwhelming if it is $1 - \mathsf{negl}(\lambda)$. For a probabilistic algorithm A, $y \leftarrow \mathsf{A}(x; r)$ means that A takes as input x and a picked randomness r, and it outputs y. For algorithms A and $\mathsf{B_A}$, $(y; z) \leftarrow (\mathsf{A} \| \mathsf{B_A})(x)$ denotes that the execution of A on input x is followed by that of $\mathsf{B_A}$ on the same input x including randomness, and $(y; z)$ is the concatenation of the outputs y and z of the two algorithms A and $\mathsf{B_A}$, respectively.

2.1 Non-Interactive Zero-Knowledge Arguments

Definition 1. *Let $\mathcal{L}(R) = \{x \mid \exists w \text{ s.t. } (x, w) \in R\}$ be the language defined by a relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$. A non-interactive zero-knowledge argument (NIZK) system for a relation R consists of three polynomial-time algorithms* $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$*:*

- $\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda)$*: The randomized algorithm* Gen *called a generator takes as input a security parameter* 1^λ*, and it outputs a common reference string (CRS)* crs*.*
- $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$*: The randomized algorithm* P *called a prover takes as input a CRS* crs*, a statement x, and a witness w, and it outputs a proof π.*
- $1/0 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi)$*: The deterministic algorithm* V *called a verifier takes as input a CRS* crs*, a statement x, and a proof π, and it outputs 1 or 0.*

We describe properties of traditional NIZKs.

Definition 2. *It is required that a NIZK system* $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ *satisfies* completeness, soundness, *and* zero-knowledge*:*

Correctness. *For every $(x, w) \in R$, it holds that*

$$\Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda); \\ \pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)\end{array} : \mathsf{V}(\mathsf{crs}, x, \pi) = 1\right] \geq 1 - \mathsf{negl}(\lambda).$$

Soundness. *For any PPT algorithm* A*, it holds that*

$$\Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda); \\ (x, \pi) \leftarrow \mathsf{A}(\mathsf{crs})\end{array} : \begin{array}{l}\mathsf{V}(\mathsf{crs}, x, \pi) = 1 \\ \wedge x \notin \mathcal{L}(R)\end{array}\right] \leq \mathsf{negl}(\lambda).$$

(Computational) Zero-Knowledge. *There exists a PPT simulator* $\mathsf{Sim} = (\mathsf{Sim}_0, \mathsf{Sim}_1)$ *such that for every PPT algorithm* A*, it holds that*

$$\left|\Pr[\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda) : 1 \leftarrow \mathsf{A}^{\mathsf{P}(\mathsf{crs}, \cdot, \cdot)}(\mathsf{crs})] - \Pr[(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Sim}_0(1^\lambda) : 1 \leftarrow \mathsf{A}^{\mathsf{Sim}^*(\cdot, \cdot)}(\mathsf{crs})]\right| \leq \mathsf{negl}(\lambda),$$

where the Sim^* *oracle on input (x, w) returns $\perp$ if $(x, w) \notin R$, and otherwise, returns* $\pi \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \mathsf{td}, x)$*, where the PPT algorithm* Sim_1 *takes as input* crs*, a trapdoor* td*, and a statement x, and outputs a simulated proof π.*

Following [32], we describe several properties of NIZKs which are required for constructing strong DSS-NIZK.

Definition 3. *Let* $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$ *be a NIZK system for a relation* R *which has the zero-knowledge simulator* $\mathsf{Sim} = (\mathsf{Sim}_0, \mathsf{Sim}_1)$.

Unbounded Simulation-Soundness. *For any PPT adversary* A, *it holds that*

$$\Pr\left[\begin{array}{l}(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Sim}_0(1^\lambda); \mathcal{Q} \leftarrow \emptyset; \\ (x^*, \pi^*) \leftarrow \mathsf{A}^{\mathsf{Sim}_1(\mathsf{crs}, \mathsf{td}, \cdot)}(\mathsf{crs})\end{array} : \begin{array}{l}(x^*, \pi^*) \notin \mathcal{Q} \wedge \\ x^* \notin \mathcal{L}(R) \wedge \\ \mathsf{V}(\mathsf{crs}, x^*, \pi^*) = 1\end{array}\right] \leq \mathsf{negl}(\lambda),$$

where the Sim_1 *oracle on input* x *returns* $\pi \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \mathsf{td}, x)$ *and sets* $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(x, \pi)\}$.

Composable Zero-Knowledge. *For any PPT adversaries* A_1 *and* A_2, *it holds that*

$$\left|\Pr\left[\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda) : 1 \leftarrow \mathsf{A}_1(\mathsf{crs})\right] - \Pr\left[(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Sim}_0(1^\lambda) : 1 \leftarrow \mathsf{A}_1(\mathsf{crs})\right]\right| \leq \mathsf{negl}(\lambda), \textit{ and}$$

$$\Big|\Pr[(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Sim}_0(1^\lambda) : 1 \leftarrow \mathsf{A}_2^{\mathsf{P}(\mathsf{crs}, \cdot, \cdot)}(\mathsf{crs}, \mathsf{td})] - \Pr[(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Sim}_0(1^\lambda) : 1 \leftarrow \mathsf{A}_2^{\mathsf{Sim}^*(\cdot, \cdot)}(\mathsf{crs}, \mathsf{td})]\Big| \leq \mathsf{negl}(\lambda),$$

where the Sim^* *oracle on input* (x, w) *returns* $\perp$ *if* $(x, w) \notin R$, *and otherwise, returns* $\pi \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \mathsf{td}, x)$.

Notice that the adversary is allowed to query x such that $x \notin \mathcal{L}(R)$ in the definition of unbounded simulation-soundness.

2.2 Dual-System Simulation-Sound NIZK

We describe the definition of dual-system (unbounded) simulation-sound NIZK (DSS-NIZK) by following [32].

Definition 4. *Let* $\mathcal{L}(R) = \{x \mid \exists w \text{ s.t. } (x, w) \in R\}$ *be the language defined by a relation* $R \subseteq \{0,1\}^* \times \{0,1\}^*$. *A DSS-NIZK system for a relation* R *consists of polynomial-time algorithms in three worlds called real world, partial-simulation world, and one-time full simulation world, as follows:*

Real World. *A DSS-NIZK in real world consists of three polynomial-time algorithms* $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$*:*

- $\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda)$*: The randomized algorithm* Gen *called a generator takes as input a security parameter* 1^λ, *and it outputs a common reference string (CRS)* crs.
- $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w, \mathsf{lbl})$*: The randomized algorithm* P *called a prover takes as input a CRS* crs, *a statement* x, *a witness* w, *and a label* $\mathsf{lbl} \in \{0,1\}^*$, *and it outputs a proof* π.
- $1/0 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi, \mathsf{lbl})$*: The deterministic algorithm* V *called a verifier takes as input a CRS* crs, *a statement* x, *a proof* π, *and a label* $\mathsf{lbl} \in \{0,1\}^*$, *and it outputs* 1 *or* 0.

Partial-Simulation World. *A DSS-NIZK in partial-simulation world consists of three polynomial-time algorithms* $(\mathsf{sfGen}, \mathsf{sfSim}, \mathsf{pV})$*:*

- $(\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_v) \leftarrow \mathsf{sfGen}(1^\lambda)$*: The randomized algorithm* sfGen, *called a semi-functional generator, takes as input a security parameter* 1^λ, *and it outputs a semi-functional CRS* crs, *and two trapdoors* td_s *and* td_v.

- $\pi \leftarrow \mathsf{sfSim}(\mathsf{crs}, \mathsf{td}_s, x, \beta, \mathsf{lbl})$*: The randomized algorithm* sfSim*, called a semi-functional simulator, takes as input a CRS* crs*, a trapdoor* td_s*, a statement* x*, a membership-bit* $\beta \in \{0,1\}$*, and a label* $\mathsf{lbl} \in \{0,1\}^*$*, and it outputs a proof* π*.*

- $1/0 \leftarrow \mathsf{pV}(\mathsf{crs}, \mathsf{td}_v, x, \pi, \mathsf{lbl})$*: The deterministic algorithm* pV*, called a private verifier, takes as input a CRS* crs*, a trapdoor* td_v*, a statement* x*, a proof* π*, and a label* $\mathsf{lbl} \in \{0,1\}^*$*, and it outputs* 1 *or* 0*.*

One-time Full Simulation World. *A DSS-NIZK in one-time full simulation world consists of three polynomial-time algorithms* $(\mathsf{otfGen}, \mathsf{otfSim}, \mathsf{sfV})$*:*

- $(\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_{s,1}, \mathsf{td}_v) \leftarrow \mathsf{otfGen}(1^\lambda)$*: The randomized algorithm* otfGen*, called a one-time full generator, takes as input a security parameter* 1^λ*, and it outputs a CRS* crs *and three trapdoors* td_s*,* $\mathsf{td}_{s,1}$*, and* td_v*.*

- $\pi \leftarrow \mathsf{otfSim}(\mathsf{crs}, \mathsf{td}_{s,1}, x, \mathsf{lbl})$*: The randomized algorithm* otfSim*, called a one-time full simulator, takes as input a CRS* crs*, a trapdoor* $\mathsf{td}_{s,1}$*, a statement* x*, and a label* $\mathsf{lbl} \in \{0,1\}^*$*, and it outputs a proof* π*.*

- $1/0 \leftarrow \mathsf{sfV}(\mathsf{crs}, \mathsf{td}_v, x, \pi, \mathsf{lbl})$*: The deterministic algorithm* sfV*, called a semi-functional verifier, takes as input a CRS* crs*, a trapdoor* td_v*, a statement* x*, a proof* π*, and a label* $\mathsf{lbl} \in \{0,1\}^*$*, and it outputs* 1 *or* 0*.*

We remark that the witness relation parameter ρ is introduced in [32] because it considers quasi-adaptive NIZK. We omit the parameter in this paper.

Definition 5. *It is required that a DSS-NIZK system for a relation* R *satisfies* completeness, partial zero-knowledge, unbounded partial simulation-soundness, *and* one-time full zero-knowledge*:*

Completeness. *For every* $(x, w) \in R$ *and every* $\mathsf{lbl} \in \{0,1\}^*$*, it holds that*

$$\Pr\left[\begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda); \\ \pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w, \mathsf{lbl}) \end{array} : \mathsf{V}(\mathsf{crs}, x, \pi, \mathsf{lbl}) = 1\right] \geq 1 - \mathsf{negl}(\lambda).$$

(Composable) Partial Zero-Knowledge. *For any PPT algorithms* A_0 *and* A_1*, it holds that*

$$\left|\Pr\left[\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda) : 1 \leftarrow \mathsf{A}_0(\mathsf{crs})\right] - \Pr\left[(\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_v) \leftarrow \mathsf{sfGen}(1^\lambda) : 1 \leftarrow \mathsf{A}_0(\mathsf{crs})\right]\right| \leq \mathsf{negl}(\lambda), \textit{ and}$$

$$\left|\Pr\left[\begin{array}{l} (\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_v) \\ \qquad \leftarrow \mathsf{sfGen}(1^\lambda) \end{array} : 1 \leftarrow \mathsf{A}_1^{\mathsf{O.PZK}_0}(\mathsf{crs})\right] - \Pr\left[\begin{array}{l} (\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_v) \\ \qquad \leftarrow \mathsf{sfGen}(1^\lambda) \end{array} : 1 \leftarrow \mathsf{A}_1^{\mathsf{O.PZK}_1}(\mathsf{crs})\right]\right| \leq \mathsf{negl}(\lambda),$$

where, let $\mathsf{O.PZK}_0 = (\mathsf{P}(\mathsf{crs}, \cdot, \cdot, \cdot), \mathsf{sfSim}(\mathsf{crs}, \mathsf{td}_s, \cdot, \cdot, \cdot), \mathsf{V}(\mathsf{crs}, \cdot, \cdot, \cdot))$ *and* $\mathsf{O.PZK}_1 := (\mathsf{sfSim}^*(\mathsf{crs}, \mathsf{td}_s, \cdot, \cdot, \cdot), \mathsf{sfSim}(\mathsf{crs}, \mathsf{td}_s, \cdot, \cdot, \cdot), \mathsf{pV}(\mathsf{crs}, \mathsf{td}_v, \cdot, \cdot, \cdot))$ *denote tuples of oracles which* A *can access, the* $\mathsf{sfSim}^*(\mathsf{crs}, \mathsf{td}_s, x, w, \mathsf{lbl})$ *oracle returns* $\mathsf{sfSim}(\mathsf{crs}, \mathsf{td}_s, x, \beta = 1, \mathsf{lbl})$*, and the challenger aborts if either* (x, w, lbl) *such that* $(x, w) \notin R$ *is queried to the first oracle (*sfSim^* *or* P*), or the second oracle* sfSim *receives a query* (x, β, lbl) *such that* $\beta = 0$ *or* $x \notin \mathcal{L}(R)$*.*

Unbounded Partial Simulation-Soundness. *For any PPT algorithm* A*, it holds that*

$$\Pr\left[\begin{array}{l} (\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_v) \leftarrow \mathsf{sfGen}(1^\lambda); \\ (x, \pi, \mathsf{lbl}) \leftarrow \mathsf{A}^{\mathsf{sfSim}(\mathsf{crs}, \mathsf{td}_s, \cdot, \cdot, \cdot), \mathsf{pV}(\mathsf{crs}, \mathsf{td}_v, \cdot, \cdot, \cdot)}(\mathsf{crs}) \end{array} : \mathsf{A} \textit{ wins}\right] \leq \mathsf{negl}(\lambda),$$

where the winning event $[\mathsf{A}$ *wins*$]$ *of* A *is defined as*

$$[\mathsf{A} \textit{ wins}] := \left[\begin{array}{l} ((x \notin \mathcal{L}(R) \vee \mathsf{V}(\mathsf{crs}, x, \pi, \mathsf{lbl}) = 0) \wedge \\ \mathsf{pV}(\mathsf{crs}, \mathsf{td}_v, x, \pi, \mathsf{lbl}) = 1 \end{array}\right].$$

One-time Full Zero-Knowledge. *For any PPT algorithm* $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$*, it holds that*

$$\left| \Pr\left[\begin{array}{l} (\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_v) \leftarrow \mathsf{sfGen}(\lambda); \\ (x^*, \beta^*, \mathsf{lbl}^*, \mathsf{st}) \leftarrow \mathsf{A}_0^{\mathsf{O.OTZK}_0}(\mathsf{crs}); \\ \pi^* \leftarrow \mathsf{sfSim}(\mathsf{crs}, \mathsf{td}_s, x^*, \beta^*, \mathsf{lbl}^*) \\ b \leftarrow \mathsf{A}_1^{\mathsf{O.OTZK}_0}(\pi^*, \mathsf{st}); \end{array} : b = 1\right] \right.$$
$$\left. - \Pr\left[\begin{array}{l} (\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_{s,1}, \mathsf{td}_v) \leftarrow \mathsf{otfGen}(\lambda); \\ (x^*, \beta^*, \mathsf{lbl}^*, \mathsf{st}) \leftarrow \mathsf{A}_0^{\mathsf{O.OTZK}_1}(\mathsf{crs}); \\ \pi^* \leftarrow \mathsf{otfSim}(\mathsf{crs}, \mathsf{td}_{s,1}, x^*, \mathsf{lbl}^*); \\ b \leftarrow \mathsf{A}_1^{\mathsf{O.OTZK}_1}(\pi^*, \mathsf{st}) \end{array} : b = 1\right] \right| \leq \mathsf{negl}(\lambda),$$

where st *is internal state-information, let* $\mathsf{O.OTZK}_0 = (\mathsf{sfSim}(\mathsf{crs}, \mathsf{td}_s, \cdot, \cdot, \cdot), \mathsf{pV}(\mathsf{crs}, \mathsf{td}_v, \cdot, \cdot, \cdot))$ *and* $\mathsf{O.OTZK}_1 = (\mathsf{sfSim}(\mathsf{crs}, \mathsf{td}_s, \cdot, \cdot, \cdot), \mathsf{sfV}(\mathsf{crs}, \mathsf{td}_v, \cdot, \cdot, \cdot))$ *denote tuples of oracles which* A *can access, and the challenger aborts if one of the following conditions holds:*

- *The generated* (x^*, β^*) *is not correct for the language* $\mathcal{L}(R)$*, where* (x, β) *is correct for a language* $\mathcal{L}(R)$ *(or* β *is correct for* x*) if* $x \in \mathcal{L}(R) \wedge \beta = 1$*, or* $x \notin \mathcal{L}(R) \wedge \beta = 0$*. Otherwise,* (x, β) *is not correct for* $\mathcal{L}(R)$ *(or* β *is not correct for* x*).*
- (x, β, lbl) *such that the membership-bit* β *is not correct for* $\mathcal{L}(R)$ *is queried to the first oracle* sfSim^**.*
- *The generated* $(x^*, \pi^*, \mathsf{lbl}^*)$ *is queried to* sfV/pV*.*

Here, for a DSS-NIZK system Π_{DN}, let $\mathsf{Adv}^{\mathrm{pzk}}_{\Pi_{\mathsf{DN}}}(\lambda)$ be the maximum probability that any PPT adversary breaks the partial zero-knowledge of Π_{DN}, let $\mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}}}(\lambda)$ be the maximum probability that any PPT adversary breaks the unbounded partial simulation-soundness of Π_{DN}, and let $\mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}}}(\lambda)$ be the maximum probability that any PPT adversary breaks the one-time full zero-knowledge of Π_{DN}. Jutla and Roy proved the following facts [32]:

Proposition 1 ([32, Lemma 4] (true simulation-soundness))**.** *Let* Π_{DN} *denote a DSS-NIZK system for a relation* R*. Then for any PPT adversary* A*, it holds that*

$$\Pr\left[\begin{array}{l} (\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_v) \leftarrow \mathsf{sfGen}(1^\lambda); \\ (x, \pi, \mathsf{lbl}) \leftarrow \mathsf{A}^{\mathsf{sfSim}(\mathsf{crs}, \mathsf{td}_s, \cdot, \cdot, \cdot)}(\mathsf{crs}) \end{array} : \mathsf{A}\ \mathit{wins}\right] \leq \mathsf{Adv}^{\mathrm{pzk}}_{\Pi_{\mathsf{DN}}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}}}(\lambda),$$

where the winning event $[\mathsf{A}\ \mathit{wins}]$ *of* A *is defined as* $[\mathsf{A}\ \mathit{wins}] := [\mathsf{V}(\mathsf{crs}, x, \pi, \mathsf{lbl}) = 1 \wedge x \notin \mathcal{L}(R)]$*, and the challenger aborts if* A *issues a query* (y, β, lbl) *such that* $y \notin \mathcal{L}(R)$ *or* $\beta = 0$*, to the* sfSim^* *oracle.*

Proposition 2 ([32, Lemma 12] (simulation-soundness of semi-functional verifier))**.** *Let* Π_{DN} *denote a DSS-NIZK system for a relation* R*. Then, for any PPT algorithm* $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$*, it holds that*

$$\Pr\left[\begin{array}{l} (\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_{s,1}, \mathsf{td}_v) \leftarrow \mathsf{otfGen}(1^\lambda); \\ (x^*, \mathsf{lbl}^*, \beta^*, \mathsf{st}) \leftarrow \mathsf{A}_0^{\mathsf{O.OTZK}_1}(\mathsf{crs}); \\ \pi^* \leftarrow \mathsf{otfSim}(\mathsf{crs}, \mathsf{td}_{s,1}, x^*, \mathsf{lbl}^*); \\ (x, \mathsf{lbl}, \pi) \leftarrow \mathsf{A}_1^{\mathsf{O.OTZK}_1}(\pi^*, \mathsf{st}) \end{array} : \mathsf{A}\ \mathit{wins}\right] \leq \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}}}(\lambda),$$

where the winning event $[\mathsf{A}\ \mathit{wins}]$ *of* A *is defined as* $[\mathsf{A}\ \mathit{wins}] := [\mathsf{sfV}(\mathsf{crs}, \mathsf{td}_v, x, \pi, \mathsf{lbl}) = 1 \wedge x \notin \mathcal{L}(R)]$*,* $\mathsf{O.OTZK}_1$ *is defined in the same way as Definition 5, and the challenger aborts if at least one of the following conditions hold:*

- *For* (x, β, lbl) *queried to the* sfSim^* *oracle,* (x, β) *is not correct for* $\mathcal{L}(R)$*.*
- β^* *is not the correct membership-bit of* $\mathcal{L}(R)$*.*
- $(x^*, \mathsf{lbl}^*, \pi^*)$ *is queried to* sfV*.*
- *The output of* A *is the same as* $(x^*, \mathsf{lbl}^*, \pi^*)$*.*

Furthermore, a stronger notion of DSS-NIZK, which is introduced in [32], is defined. We call reveal event when td_s is revealed to adversaries where $(\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_v) \leftarrow \mathsf{sfGen}(1^\lambda)$ or $(\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_{s,1}, \mathsf{td}_v) \leftarrow \mathsf{otfGen}(1^\lambda)$ [32].

Definition 6 (Strong DSS-NIZK [32])**.** *A DSS-NIZK system with partial simulation trapdoor reveal oracle is a strong DSS-NIZK system with the following changes to the DSS-NIZK definition:*

- *The first part of the* composable partial zero-knowledge *continues to hold.*
- *The second part of the* composable partial zero-knowledge *holds under the additional restriction that the adversary cannot invoke the third oracle (i.e.,* V *or* pV *oracle) after the reveal event.*
- *The* unbounded partial simulation-soundness *continues to hold.*
- *The trapdoors* td_s *and* $\mathsf{td}_{s,1}$ *generated by* otfGen *are same and statistically indistinguishable from* td_s *generated by* sfGen*.*
- *The* one-time full zero-knowledge *holds under the additional restriction that* $(x^*, \beta^*, \mathsf{lbl}^*)$ *is such that* $x^* \in \mathcal{L}(R)$ *and* $\beta^* = 1$ *and the second oracle (i.e.,* pV *or* sfV *oracle) is not invoked after the reveal event.*
- *The* simulation-soundness *of* sfV *(Proposition 2) holds under the additional restriction that* sfV *oracle is not invoked after the reveal event. Notice that there is no restriction that* $(x^*, \beta^*, \mathsf{lbl}^*)$ *is such that* $x^* \in \mathcal{L}(R)$ *and* $\beta^* = 1$*.*

The adversaries against the above properties are given access to the partial simulation trapdoor reveal oracle which, on input a request, returns td_s *for* $(\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_v) \leftarrow \mathsf{sfGen}(1^\lambda)$ *or* $(\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_{s,1}, \mathsf{td}_v) \leftarrow \mathsf{otfGen}(1^\lambda)$*.*

2.3 Smooth Projective Hash Proof System

Following [19], we describe the definition of (smooth) projective hash proof systems (PHPSs), as follows:

Definition 7 (Projective Hash Family [19])**.** *Let* X *and* Π *be finite sets. Let* $H = \{H_k\}_{k \in K}$ *be a collection of functions indexed by* K *so that* $H_k : X \to \Pi$ *is a hash function for every* $k \in K$*. Then,* (H, K, X, Π) *is called a* hash family*. Let* L *be a non-empty proper subset of* X*. Let* S *be a finite set, and* $\alpha : K \to S$ *be a function.* $\mathbf{H} = (H, K, X, \Pi, L, S, \alpha)$ *is called a* projective hash family *(PHF) if for every* $k \in K$*, the action of* H_k *on* L *is determined by* $\alpha(k)$*.*

Definition 8 ((Smooth) PHPS [19])**.** *For a PHF* $\mathbf{H} = (H, K, X, \Pi, L, S, \alpha)$ *(where languages are defined by a relation* $R \subseteq \{0,1\}^* \times \{0,1\}^*$*), let* $\hat{H}$ *be a public evaluation function which takes the projection key* $\alpha(k)$*, a statement* $x \in L = \{x \mid \exists w \text{ s.t. } (x, w) \in R\}$*, and a witness* w *such that* $(x, w) \in R$*, and it computes* $H_k(x)$*. The PHF* $\mathbf{H}$ *constitutes a* projective hash proof system *(PHPS) if* α*,* H_k*, and* $\hat{H}$ *are efficiently computable.*

Furthermore, a PHPS constituted by a PHF $\mathbf{H} = (H, K, X, \Pi, L, S, \alpha)$ *is called a labeled PHPS if the public evaluation function takes an additional input* $\mathsf{lbl} \in \{0,1\}^*$ *which is called a label. A labeled PHPS is* ϵ-smooth *if the statistical distance between* $U(\mathbf{H}) = (x, \alpha(k), \pi')$ *and* $V(\mathbf{H}) = (x, \alpha(k), H_k(x, \mathsf{lbl}))$ *is at most* ϵ *for all* $k \in K$*, all* $x \in X \backslash L$*, all* $\mathsf{lbl} \in \{0,1\}^*$*, and all* $\pi' \in \Pi$*.*

2.4 (Keyed-)Fully Homomorphic Encryption

Fully Homomorphic Encryption (FHE). We first describe the syntax and a security definition of FHE by following [12].

Definition 9. *For a security parameter* λ*, let* $\mathcal{M} = \mathcal{M}(\lambda)$ *be a message space. An FHE scheme consists of four polynomial-time algorithms* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$*:*

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$*: The randomized algorithm* KGen *takes as input a security parameter* 1^λ*, and it outputs a public key* pk *and a secret key* sk*.*
- $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$*: The randomized algorithm* Enc *takes as input a public key* pk *and a message* $\mathsf{m} \in \mathcal{M}$*, and it outputs a ciphertext* ct*.*
- $\mathsf{m}/\bot \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$*: The deterministic algorithm* Dec *takes as input a secret key* sk *and a ciphertext* ct*, and it outputs a message* $\mathsf{m} \in \mathcal{M}$ *or a rejection symbol* $\bot$*.*
- $\widehat{\mathsf{ct}} \leftarrow \mathsf{Eval}(\mathsf{C}, (\mathsf{ct}^{(1)}, \mathsf{ct}^{(2)}, \ldots, \mathsf{ct}^{(\ell)}))$*: The deterministic or randomized algorithm* Eval *takes as input a circuit* $\mathsf{C} : \mathcal{M}^\ell \rightarrow \mathcal{M}$ *and a tuple of ciphertexts* $(\mathsf{ct}^{(1)}, \mathsf{ct}^{(2)}, \ldots, \mathsf{ct}^{(\ell)})$*, and it outputs a new ciphertext* $\widehat{\mathsf{ct}}$*.*

We require that an FHE scheme meets both correctness and compactness.

Definition 10 (Correctness)**.** *An FHE scheme* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *satisfies* correctness *if the following conditions hold:*

- *For every* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$ *and every* $\mathsf{m} \in \mathcal{M}$*, it holds that* $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct}) = \mathsf{m}$ *with overwhelming probability, where* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$*.*
- *For every* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$*, every circuit* C*, and every* $(\mathsf{m}^{(1)}, \ldots, \mathsf{m}^{(\ell)}) \in \mathcal{M}^\ell$*, it holds that* $\mathsf{Dec}(\mathsf{sk}, \widehat{\mathsf{ct}}) = \mathsf{C}(\mathsf{m}^{(1)}, \ldots, \mathsf{m}^{(\ell)})$ *with overwhelming probability, where* $\widehat{\mathsf{ct}} \leftarrow \mathsf{Eval}(\mathsf{C}, (\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}))$ *and for every* $i \in [\ell]$*,* $\mathsf{ct}^{(i)} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}^{(i)})$*.*

Definition 11 (Compactness)**.** *An FHE scheme satisfies* compactness *if there exists a polynomial* poly *such that the output-size of* $\mathsf{Eval}(\cdot, \cdot)$ *is at most* $\mathsf{poly}(\lambda)$ *for every security parameter* λ*.*

The IND-CCA1 security of FHE is defined as follows.

Definition 12 (IND-CCA1 security)**.** *An FHE scheme* $\Pi_{\mathsf{FHE}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *is* IND-CCA1 *secure if for any PPT adversary* $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ *against* Π_{FHE}*, the advantage*

$$\mathsf{Adv}^{\text{ind-cca1}}_{\Pi_{\mathsf{FHE}},\mathsf{A}}(\lambda) := \left| \Pr\left[\begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda); \\ (\mathsf{m}_0, \mathsf{m}_1, \mathsf{st}) \leftarrow \mathsf{A}_0^{\mathsf{Dec}(\mathsf{sk}, \cdot)}(\mathsf{pk}); \\ b \xleftarrow{\$} \{0,1\}; \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b); \\ b' \leftarrow \mathsf{A}_1(\mathsf{ct}^*, \mathsf{st}) \end{array} : b = b' \right] - \frac{1}{2} \right|$$

is negligible in λ*, where* st *is internal state information.*

In addition, $\mathsf{IND\text{-}CPA}$ security is defined in the same way as $\mathsf{IND\text{-}CCA1}$ security except that the adversary is not given access to the decryption oracle Dec.

Keyed-Fully Homomorhpic Encryption (Keyed-FHE). Following the definition of KH-PKE in [22], we describe the definition of keyed-fully homomorphic encryption (keyed-FHE) given by Lai et al. [33]

Definition 13. *For a security parameter λ, let $\mathcal{M} = \mathcal{M}(\lambda)$ be a message space. A keyed-FHE scheme consists of four polynomial-time algorithms* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$*:*

- $(\mathsf{pk}, \mathsf{sk}_d, \mathsf{sk}_h) \leftarrow \mathsf{KGen}(1^\lambda)$*: The randomized algorithm* KGen *takes as input a security parameter 1^λ, and it outputs a public key* pk*, a decryption key* sk_d*, and a homomorphic evaluation key* sk_h*.*

- $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$*: The randomized algorithm* Enc *takes as input a public key* pk *and a message* $\mathsf{m} \in \mathcal{M}$*, and it outputs a ciphertext* ct*.*

- $\mathsf{m}/\bot \leftarrow \mathsf{Dec}(\mathsf{sk}_d, \mathsf{ct})$*: The deterministic algorithm* Dec *takes as input a decryption key* sk_d *and a ciphertext* ct*, and it outputs a message $\mathsf{m} \in \mathcal{M}$ or a rejection symbol $\bot$.*

- $\widehat{\mathsf{ct}}/\bot \leftarrow \mathsf{Eval}(\mathsf{sk}_h, \mathsf{C}, (\mathsf{ct}^{(1)}, \mathsf{ct}^{(2)}, \ldots, \mathsf{ct}^{(\ell)}))$*: The deterministic or randomized algorithm* Eval *takes as input a homomorphic evaluation key* sk_h*, a circuit $\mathsf{C} : \mathcal{M}^\ell \to \mathcal{M}$, and a tuple of ciphertexts $(\mathsf{ct}^{(1)}, \mathsf{ct}^{(2)}, \ldots, \mathsf{ct}^{(\ell)})$, and it outputs a new ciphertext $\widehat{\mathsf{ct}}$ or a rejection symbol $\bot$.*

We require that a keyed-FHE scheme satisfies both correctness and compactness.

Definition 14 (Correctness)**.** *A keyed-FHE scheme* $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *satisfies* correctness *if the following conditions hold:*

- *For every $(\mathsf{pk}, \mathsf{sk}_d, \mathsf{sk}_h) \leftarrow \mathsf{KGen}(1^\lambda)$ and every $\mathsf{m} \in \mathcal{M}$, it holds that $\mathsf{Dec}(\mathsf{sk}_d, \mathsf{ct}) = \mathsf{m}$ with overwhelming probability, where $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$.*

- *For every $(\mathsf{pk}, \mathsf{sk}_d, \mathsf{sk}_h) \leftarrow \mathsf{KGen}(1^\lambda)$, every circuit $\mathsf{C} : \mathcal{M}^\ell \to \mathcal{M}$, and every $(\mathsf{m}^{(1)}, \ldots, \mathsf{m}^{(\ell)}) \in \mathcal{M}^\ell$, it holds that $\mathsf{Dec}(\mathsf{sk}_d, \widehat{\mathsf{ct}}) = \mathsf{C}(\mathsf{m}^{(1)}, \ldots, \mathsf{m}^{(\ell)})$ with overwhelming probability, where $\mathsf{ct} \leftarrow \mathsf{Eval}(\mathsf{sk}_h, \mathsf{C}, (\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}))$ and for every $i \in [\ell]$, $\mathsf{ct}^{(i)} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}^{(i)})$.*

Definition 15 (Compactness)**.** *A keyed-FHE scheme satisfies* compactness *if there exists a polynomial* poly *such that the output-size of* $\mathsf{Eval}(\mathsf{sk}_h, \cdot, \cdot, \cdot)$ *is at most* $\mathsf{poly}(\lambda)$ *for every security parameter λ.*

As a security notion of keyed-FHE, we describe the definition of $\mathsf{KH\text{-}CCA}$ security [22]. This is the same as the security considered in [33] except that the adversary is allowed to access the decryption oracle until the homomorphic evaluation key is revealed.

Definition 16 ($\mathsf{KH\text{-}CCA}$ security)**.** *A keyed-FHE scheme $\Pi_{\mathsf{KFHE}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is* $\mathsf{KH\text{-}CCA}$ *secure if for any PPT adversary $\mathsf{A} = (\mathsf{A}_0, \mathsf{A}_1)$ against Π_{KFHE}, the advantage*

$$\mathsf{Adv}^{\text{kh-cca}}_{\Pi_{\mathsf{KFHE}},\mathsf{A}}(\lambda) := \left| \Pr\left[\begin{array}{l} (\mathsf{pk}, \mathsf{sk}_d, \mathsf{sk}_h) \leftarrow \mathsf{KGen}(1^\lambda); \\ (\mathsf{m}_0, \mathsf{m}_1, \mathsf{st}) \leftarrow \mathsf{A}_0^{\mathsf{O.KFHE}}(\mathsf{pk}); \\ b \xleftarrow{\$} \{0,1\}; \mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m}_b); \\ b' \leftarrow \mathsf{A}_1^{\mathsf{O.KFHE}}(\mathsf{ct}^*, \mathsf{st}) \end{array} : b = b' \right] - \frac{1}{2} \right|$$

is negligible in λ, where st *is state information, and let $\mathcal{D}$ be a list which is set as $\mathcal{D} \leftarrow \{\mathsf{ct}^*\}$ in* ***Challenge*** *phase, and* $\mathsf{O.KFHE}$ *consists of three oracles* $(\mathsf{RevHK}(), \mathsf{Eval}(\mathsf{sk}_h, \cdot, \cdot), \mathsf{Dec}(\mathsf{sk}_d, \cdot))$ *defined as follows:*

- *Homomorphic key reveal oracle* RevHK*: Given a request, the* RevHK *oracle returns* sk_h.

- *Evaluation oracle* Eval*: Given an* Eval *query* $(\mathsf{C}, (\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}))$*, the* Eval *oracle checks whether the* RevHK *oracle has been queried before. If so, it returns* $\perp$*. Otherwise, it returns* $\widehat{\mathsf{ct}}/\perp \leftarrow \mathsf{Eval}(\mathsf{sk}_h, \mathsf{C}, (\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}))$*. In addition, if* $\widehat{\mathsf{ct}} \neq \perp$ *and one of ciphertexts* $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}$ *is in* $\mathcal{D}$*, it sets* $\mathcal{D} \leftarrow \mathcal{D} \cup \{\widehat{\mathsf{ct}}\}$*.*

- *Decryption oracle* Dec*: This oracle is not available if* A *has accessed the* RevHK *oracle and obtained the challenge ciphertext* ct^**. Given a* Dec *query* ct*, the* Dec *oracle returns* $\mathsf{Dec}(\mathsf{sk}_d, \mathsf{ct})$ *if* $\mathsf{ct} \notin \mathcal{D}$*, and returns* $\perp$ *otherwise.*

3 Generic Construction of keyed-FHE

3.1 Our Construction

In this section, we propose a keyed-FHE scheme Π_{KFHE} and prove its security. Π_{KFHE} is constructed from two $\mathsf{IND}\text{-}\mathsf{CCA1}$ secure FHE schemes $\Pi_{\mathsf{FHE},1}, \Pi_{\mathsf{FHE},2}$ and a (strong) DSS-NIZK system Π_{DN}. We briefly explain an overview of the construction whose spirit is similar to Jutla and Roy's KH-PKE scheme [32] except that we use the Naor-Yung paradigm [40]. Let $(\mathsf{pk}_1, \mathsf{sk}_1)$ and $(\mathsf{pk}_2, \mathsf{sk}_2)$ denote two pairs of public/secret keys of $\Pi_{\mathsf{FHE},1}$ and $\Pi_{\mathsf{FHE},2}$. A public key $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2, \mathsf{crs})$ of Π_{KFHE} consists of two public keys $(\mathsf{pk}_1, \mathsf{pk}_2)$ of schemes $\Pi_{\mathsf{FHE},1}, \Pi_{\mathsf{FHE},2}$ and the CRS crs of Π_{DN}, while the secret key $\mathsf{sk}_d = \mathsf{sk}_1$ is the secret key of $\Pi_{\mathsf{FHE},1}$. The ciphertext $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2, \pi)$ consists of two FHE ciphertexts $(\mathsf{ct}_1, \mathsf{ct}_2)$ both of which are encryptions of m and π is a proof such that $(\mathsf{ct}_1, \mathsf{ct}_2)$ are encryptions of the same message. The decryption algorithm first checks the validity of π by using the real world verification algorithm V_N, then decrypt ct_1 by using $\mathsf{sk}_d = \mathsf{sk}_1$. To complete the overview, we show how to evaluate keyed-FHE ciphertexts $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}$ for a circuit C and obtain $\widehat{\mathsf{ct}}$. A point to note is that we should create a proof $\widehat{\pi}$ without the knowledge of the message $\mathsf{C}(\mathsf{m}^{(1)}, \ldots, \mathsf{m}^{(\ell)})$ of $\widehat{\mathsf{ct}}$. For this purpose, we use the DSS-NIZK system Π_{DN} in the *partial-simulation world* as the case of Jutla and Roy's KH-PKE scheme [32]. Then, we set the homomorphic evaluation key $\mathsf{sk}_h = \mathsf{td}_s$ as the trapdoor of Π_{DN}. Therefore, the *(composable)* partial zero-knowledge ensures that $\widehat{\pi}$ can be computed correctly by using the sfSim_N algorithm. Here, we note that the verification algorithm V_N can correctly verify the proof created by the sfSim_N algorithm owing to partial zero-knowledge.

To sum up, we use the following primitives: An FHE scheme $\Pi_{\mathsf{FHE},i} = (\mathsf{KGen}_{F,i}, \mathsf{Enc}_{F,i}, \mathsf{Dec}_{F,i}, \mathsf{Eval}_{F,i})$ with the message space $\mathcal{M}$ for $i \in \{1, 2\}$, and a DSS-NIZK system Π_{DN} in the partial-simulation world $(\mathsf{sfGen}_N, \mathsf{sfSim}_N, \mathsf{pV}_N)$[4] for a relation $R_N = \{(\mathsf{ct}_1, \mathsf{ct}_2), (\mathsf{m}, r_1, r_2) \mid \mathsf{ct}_1 = \mathsf{Enc}_{F,1}(\mathsf{pk}_1, \mathsf{m}; r_1) \wedge \mathsf{ct}_2 = \mathsf{Enc}_{F,2}(\mathsf{pk}_2, \mathsf{m}; r_2)\}$, where $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}_{F,1}(1^\lambda)$ and $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \mathsf{KGen}_{F,2}(1^\lambda)$.

Our scheme $\Pi_{\mathsf{KFHE}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is constructed as follows:

- $(\mathsf{pk}, \mathsf{sk}_d, \mathsf{sk}_h) \leftarrow \mathsf{KGen}(1^\lambda)$:

  1. $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}_{F,1}(1^\lambda)$, $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \mathsf{KGen}_{F,2}(1^\lambda)$.
  2. $(\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_v) \leftarrow \mathsf{sfGen}_N(1^\lambda)$.
  3. Output $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2, \mathsf{crs})$, $\mathsf{sk}_d = \mathsf{sk}_1$, and $\mathsf{sk}_h = \mathsf{td}_s$.

- $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$:

[4] A proof generated by the sfSim_N algorithm can be verified by the real world verification algorithm V_N owing to the partial zero-knowledge property. Thus, we use the V_N algorithm in our construction.

1. $\mathsf{ct}_1 \leftarrow \mathsf{Enc}_{F,1}(\mathsf{pk}_1, \mathsf{m}; r_1)$, $\mathsf{ct}_2 \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, \mathsf{m}; r_2)$.
2. $\pi \leftarrow \mathsf{P}_N(\mathsf{crs}, (\mathsf{ct}_1, \mathsf{ct}_2), (\mathsf{m}, r_1, r_2), \emptyset)$.
3. Output $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2, \pi)$.

- $\mathsf{m}/\bot \leftarrow \mathsf{Dec}(\mathsf{sk}_d, \mathsf{ct})$: Let $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2, \pi)$.

  1. If $\mathsf{V}_N(\mathsf{crs}, (\mathsf{ct}_1, \mathsf{ct}_2), \pi, \emptyset) = 1$, output $\mathsf{m} \leftarrow \mathsf{Dec}_{F,1}(\mathsf{sk}_1, \mathsf{ct}_1)$. Otherwise, output $\bot$.

- $\widehat{\mathsf{ct}}/\bot \leftarrow \mathsf{Eval}(\mathsf{sk}_h, \mathsf{C}, (\mathsf{ct}^{(1)}, \dots, \mathsf{ct}^{(\ell)}))$: Let $\mathsf{ct}^{(i)} = (\mathsf{ct}_1^{(i)}, \mathsf{ct}_2^{(i)}, \pi^{(i)})$ for $i \in [\ell]$.

  1. Output $\bot$ if $\mathsf{V}_N(\mathsf{crs}, (\mathsf{ct}_1^{(i)}, \mathsf{ct}_2^{(i)}), \pi^{(i)}, \emptyset) = 0$ for some $i \in [\ell]$.
  2. $\widehat{\mathsf{ct}}_1^{(0)} \leftarrow \mathsf{Enc}_{F,1}(\mathsf{pk}_1, 0)$, $\widehat{\mathsf{ct}}_2^{(0)} \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, 0)$, where 0 is the additive identity in $\mathcal{M}$.
  3. $\widehat{\mathsf{ct}}_1' \leftarrow \mathsf{Eval}_{F,1}(\mathsf{C}, (\mathsf{ct}_1^{(1)}, \dots, \mathsf{ct}_1^{(\ell)}))$, $\widehat{\mathsf{ct}}_2' \leftarrow \mathsf{Eval}_{F,2}(\mathsf{C}, (\mathsf{ct}_2^{(1)}, \dots, \mathsf{ct}_2^{(\ell)}))$.
  4. $\widehat{\mathsf{ct}}_1 \leftarrow \mathsf{Add}_{F,1}(\widehat{\mathsf{ct}}_1', \widehat{\mathsf{ct}}_1^{(0)})$, $\widehat{\mathsf{ct}}_2 \leftarrow \mathsf{Add}_{F,2}(\widehat{\mathsf{ct}}_2', \widehat{\mathsf{ct}}_2^{(0)})$, where the PPT algorithm $\mathsf{Add}_{F,i}$ ($i \in \{1,2\}$) evaluates the addition gate over $\mathcal{M}$ by using $\mathsf{Eval}_{F,i}$.
  5. $\widehat{\pi} \leftarrow \mathsf{sfSim}_N(\mathsf{crs}, \mathsf{td}_s, (\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2), 1, \emptyset)$.
  6. Output $\widehat{\mathsf{ct}} = (\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2, \widehat{\pi})$.

The correctness of Π_{KFHE} follows the correctness of $\Pi_{\mathsf{FHE},1}$ and $\Pi_{\mathsf{FHE},2}$, and the completeness of Π_{DN}. The first condition of the correctness holds since the completeness of Π_{DN} ensures that V_N outputs 1 and the correctness of $\Pi_{\mathsf{FHE},1}$ ensures that $\mathsf{Dec}_{F,1}$ correctly outputs m with overwhelming probability. Similarly, the second condition of the correctness also holds since the composable partial zero-knowledge of Π_{DN} ensures that V_N outputs 1 even if the proof $\widehat{\pi}$ is computed by the semi-functional simulator sfSim_N. In addition, the output-size of sfSim_N used in Eval is equal to that of P_N due to the partial zero-knowledge property, since it is possible to break this property if these output-sizes are different. Thus, the compactness of Π_{KFHE} follows the compactness of $\Pi_{\mathsf{FHE},1}$ and $\Pi_{\mathsf{FHE},2}$.

Remark 1. *Canetti et al. [12] showed that* IND-CCA1 *secure FHE can be constructed from* IND-CPA *secure FHE and zk-SNARK via the Naor-Yung transformation. Here, circuit* C *to be evaluated is a witness and thus the underlying NIZK system needs to be succinct. On the other hand, in our evaluation algorithm first ciphertexts are evaluated by the evaluation algorithm of the underlying* IND-CCA1 *secure FHE schemes, and then the underlying NIZK system proves that two ciphertexts* $\widehat{\mathsf{ct}}_1$ *and* $\widehat{\mathsf{ct}}_2$ *have the same plaintext using the trapdoor. So,* C *is not a witness here, and we do not have to directly employ zk-SNARK in our construction.*

3.2 Security Analysis

Theorem 1 (KH-CCA security of Π_{KFHE})**.** *If both* $\Pi_{\mathsf{FHE},1}$ *and* $\Pi_{\mathsf{FHE},2}$ *are* IND-CCA1 *secure, and* Π_{DN} *is a strong DSS-NIZK system, then the resulting keyed-FHE scheme* Π_{KFHE} *is* KH-CCA *secure.*

Overview of Proof of Theorem 1. Theorem 1 shows the security of our keyed-FHE scheme. For simplicity, we explain that our scheme satisfies KH-CCA security if the underlying NIZK system Π_{DN} meets the properties of strong DSS-NIZKs, and the underlying FHE schemes satisfy IND-CCA1 security. We give the intuitive explanation. To guarantee security against adaptive chosen ciphertext attacks before a homomorphic evaluation key (a trapdoor of Π_{DN}) is revealed by RevHK oracle access, the underlying DSS-NIZK system must satisfy (one-time) simulation-soundness so that we

Table 1: Summary of Games in the Proof of Theorem 1

Game	Components of ct^*		$\mathsf{C}(\mathsf{m}_1,\ldots,\mathsf{m}_\ell)$ computed for Dep. Eval	Verification of		Msg-Rec. of Dec
	ct_2^*	π^*		Indep. Eval	Dec	
Game_0	$\mathsf{Enc}_{F,2}(\mathsf{m}_b)$	P_N^*	Ordinary	V_N	V_N	$\mathsf{Dec}_{F,1}$
Game_1	$\mathsf{Enc}_{F,2}(\mathsf{m}_b)$	sfSim_N^*	Ordinary	pV_N	pV_N	$\mathsf{Dec}_{F,1}$
Game_2	$\mathsf{Enc}_{F,2}(\mathsf{m}_b)$	sfSim_N^*	Random	pV_N	pV_N	$\mathsf{Dec}_{F,1}$
Game_3	$\mathsf{Enc}_{F,2}(\mathsf{m}_b)$	otfSim_N^*	Random	sfV_N	sfV_N	$\mathsf{Dec}_{F,1}$
Game_4	$\mathsf{Enc}_{F,2}(0^{\lvert\mathsf{m}_b\rvert})$	otfSim_N^*	Random	sfV_N	sfV_N	$\mathsf{Dec}_{F,1}$
Game_5	$\mathsf{Enc}_{F,2}(0^{\lvert\mathsf{m}_b\rvert})$	otfSim_N^*	Random	sfV_N	sfV_N	$\mathsf{Dec}_{F,2}$

"$\mathsf{C}(\mathsf{m}_1,\ldots,\mathsf{m}_\ell)$ computed for Dep. Eval" denotes a message $\mathsf{C}(\mathsf{m}_1,\ldots,\mathsf{m}_\ell)$ for $\widehat{\mathsf{ct}}$ generated by the Eval oracle on input a dependent Eval query. "Ordinary" (resp. "Random") means that $\mathsf{C}(\mathsf{m}_1,\ldots,\mathsf{m}_\ell)$ is a message whose encryption is generated by the Eval algorithm on input encryptions queried by the adversary A (resp. encryptions of random messages). "Verification of Indep. Eval" denotes a verification algorithm in the Eval algorithm run by the Eval oracle on input an independent Eval query. "Verification of Dec" denotes a verification algorithm in the Dec algorithm run by the Dec oracle on input a Dec query. "Msg-Rec. of Dec" denotes an algorithm which recovers a message in the Dec algorithm run by Dec oracle on input a Dec query. For $i \in \{1,2\}$, let $\mathsf{Enc}_{F,i}(\cdot) = \mathsf{Enc}_{F,i}(\mathsf{pk}_i,\cdot)$ and $\mathsf{Dec}_{F,i}(\cdot) = \mathsf{Dec}_{F,i}(\mathsf{sk}_i,\cdot)$. Let $\mathsf{P}_N^* = \mathsf{P}_N(\mathsf{crs},(\mathsf{ct}_1^*,\mathsf{ct}_2^*),(\mathsf{m}_b,r_1^*,r_2^*),\emptyset)$, $\mathsf{sfSim}_N^* = \mathsf{sfSim}_N(\mathsf{crs},\mathsf{td}_s,(\mathsf{ct}_1^*,\mathsf{ct}_2^*),1,\emptyset)$, and $\mathsf{otfSim}_N^* = \mathsf{otfSim}_N(\mathsf{crs},\mathsf{td}_{s,1},(\mathsf{ct}_1^*,\mathsf{ct}_2^*),\emptyset)$.

can correctly return the non-malleable ciphertext generated in the challenge phase. The unbounded (partial) simulation-soundness is required for Π_{DN} in order to return non-malleable ciphertexts for evaluation queries. This is because this property ensures that A cannot generate valid evaluated ciphertexts whose $\mathsf{IND\text{-}CCA1}$ FHE ciphertexts are not in the language of the strong DSS-NIZK, by substituting evaluated ones. Moreover, our scheme needs the $\mathsf{partial\ zero\text{-}knowledge}$ and $\mathsf{one\text{-}time\ full\ zero\text{-}knowledge}$ properties of strong DSS-NIZKs, so that the challenge message can be hidden even if a simulation trapdoor of Π_{DN} is revealed. Since we assume that the underlying FHE schemes are $\mathsf{IND\text{-}CCA1}$ secure, we can simulate decryption queries until the challenge phase.

Remark 2. *Although we assume that the underlying FHE schemes are* $\mathsf{IND\text{-}CCA1}$ *secure in Theorem 1, we can prove* $\mathsf{KH\text{-}CCA}$ *security even when the underlying FHE schemes are* $\mathsf{IND\text{-}CPA}$ *secure. For this purpose, we follow Canetti et al. generic construction [12] and additionally use a zk-SNARK (this construction is concretely described in A). Nevertheless, we assume* $\mathsf{IND\text{-}CCA1}$ *security of the underlying FHE schemes since it enables us to obtain a much simpler proof.*

We explain the overview of the security proof, more concretely. For simplicity, we classify the queries to the Eval oracle into two types: *Dependent* Eval queries and *independent* Eval queries.

- Let a dependent Eval query be a query $(\mathsf{C},(\mathsf{ct}^{(1)},\ldots,\mathsf{ct}^{(\ell)}))$ issued to the Eval oracle, such that at least one of $\mathsf{ct}^{(1)},\ldots,\mathsf{ct}^{(\ell)}$ are in the set $\mathcal{D}$ of the derivatives of the challenge ciphertext.
- Let an independent Eval query be a query issued to the Eval oracle, such that all $\mathsf{ct}^{(1)},\ldots,\mathsf{ct}^{(\ell)}$ are not in $\mathcal{D}$.

In order to prove Theorem 1, we consider security games $\mathsf{Game}_0,\ldots,\mathsf{Game}_5$ (Table 1 shows the summary of these games).

- Game_0 is the ordinary $\mathsf{KH\text{-}CCA}$ security game.

Table 2: Outline of the Proof of Theorem 1

Game	Property
$\mathsf{Game}_0 \approx \mathsf{Game}_1$	partial zero-knowledge of Π_{DN}, true simulation-soundness of Π_{DN}
$\mathsf{Game}_1 \approx \mathsf{Game}_2$	one-time full zero-knowledge of Π_{DN}, unbounded partial simulation-soundness of Π_{DN}, IND-CCA1 security of $\Pi_{\mathsf{FHE},1}$ and $\Pi_{\mathsf{FHE},2}$
$\mathsf{Game}_2 \approx \mathsf{Game}_3$	one-time full zero-knowledge of Π_{DN}, simulation-soundness of sfV_N
$\mathsf{Game}_3 \approx \mathsf{Game}_4$	simulation-soundness of sfV_N, IND-CCA1 security of $\Pi_{\mathsf{FHE},2}$
$\mathsf{Game}_4 \approx \mathsf{Game}_5$	one-time full zero-knowledge of Π_{DN}, unbounded partial simulation-soundness of Π_{DN}
Game_5	IND-CCA1 security of $\Pi_{\mathsf{FHE},1}$

- Game_1 is the same as Game_0 except that the P algorithm and the V algorithm are replaced by the sfSim algorithm and the pV algorithm, respectively, when simulating **Challenge** phase and the Eval oracle. The indistinguishability between Game_0 and Game_1 mainly follows the partial zero-knowledge of Π_{DN}.

- Game_2 is the same as Game_1 except that the derivatives of the challenge ciphertexts are replaced by encryptions of random messages. The indistinguishability between Game_1 and Game_2 mainly follows the unbounded partial simulation-soundness of Π_{DN}.

- Game_3 is the same as Game_2 except that the sfSim algorithm and the pV algorithm are replaced by the otfSim algorithm and sfV algorithm, respectively, when simulating the challenge phase and the Eval oracle. The indistinguishability between Game_2 and Game_3 mainly follows the one-time full zero-knowledge of Π_{DN}.

- Game_4 is the same as Game_3 except that the challenge ciphertext generated by $\Pi_{\mathsf{FHE},2}$ is replaced by an encryption of $0^{|\mathsf{m}_b|}$. The indistinguishability between Game_3 and Game_4 mainly follows the IND-CCA1 security of $\Pi_{\mathsf{FHE},2}$.

- Game_5 is the same as Game_4 except that the decryption oracle uses the $\mathsf{Dec}_{F,2}$ algorithm instead of $\mathsf{Dec}_{F,1}$, when decrypting the given ciphertext. The indistinguishability between Game_4 and Game_5 mainly follows the one-time full zero-knowledge and unbounded partial simulation-soundness of Π_{DN}. In addition, the indistinguishability in Game_5 follows the IND-CCA1 security of $\Pi_{\mathsf{FHE},1}$.

The proof of the indistinguishability between Game_0 and Game_3 is similar to a part of the security proof of the Jutla and Roy's scheme [32] because this indistinguishability follows the properties of the underlying strong DSS-NIZK (see also Table 2). The remaining proofs are different from the security proof of [32], because our scheme employs the Naor-Yung paradigm while the Jutla and Roy's scheme uses a variant of ElGamal encryption.

Furthermore, we describe the important point of our security proof. The security proofs in Game_4 and Game_5 are similar to those of the Naor-Yung paradigm [40]. In these games, the challenge ciphertext is an invalid one due to reductions from the security of the underlying primitives. However, when an adversary issues the challenge ciphertext (or derivatives of the challenge ciphertext) to the Eval oracle, this oracle must return a valid ciphertext. In order to simulate the

Eval oracle correctly even in this case, the Eval oracle on input a dependent Eval query returns a random and valid ciphertext instead of an ordinary evaluated ciphertext, in Game_2. The partial zero-knowledge property and unbounded (partial) simulation soundness mainly ensure that Game_2 is indistinguishable from the previous games. Hence, it is possible to replace the ordinary challenge ciphertext by an invalid one in the security games after Game_2.

3.2.1 Proof of Theorem 1

We give the proof of Theorem 1 as follows. Let A be a PPT adversary against Π_{KFHE}. We recall that a *dependent* Eval query is a query $(\mathsf{C}, (\mathsf{ct}^{(1)}, \dots, \mathsf{ct}^{(\ell)}))$ issued to the Eval oracle, such that at least one of $\mathsf{ct}^{(1)}, \dots, \mathsf{ct}^{(\ell)}$ are in $\mathcal{D}$, and an *independent* Eval query is a query issued to the Eval oracle, such that all $\mathsf{ct}^{(1)}, \dots, \mathsf{ct}^{(\ell)}$ are not in $\mathcal{D}$.

By definition, we can immediately detect whether A's Eval queries are dependent or independent. Let Q_{dep} be the number of dependent Eval queries. Let $\mathsf{Adv}^{\mathrm{pzk}}_{\Pi_{\mathsf{DN}},\mathsf{B}_1}$, $\mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{B}_2}(\lambda)$, and $\mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{B}_3}$ be the maximum probabilities that any PPT adversaries B_1, B_2, and B_3 break the partial zero-knowledge in the second part, the unbounded partial simulation-soundness, and the one-time full zero-knowledge properties of Π_{DN}, respectively. Let reveal event be the event that the homomorphic evaluation key (resp. the partial simulation trapdoor) is revealed by accessing the reveal oracle in the KH-CCA security game (resp. a security game of strong DSS-NIZKs).

We consider security games $\mathsf{Game}_0, \mathsf{Game}_1, \dots, \mathsf{Game}_5$. Regarding the summary of these games, see Table 1. For $i \in \{0, 1, \dots, 5\}$, let W_i be the event that A outputs $b' \in \{0, 1\}$ such that $b = b'$ in Game_i.

$\underline{\mathsf{Game}_0}$: The same game as the ordinary KH-CCA game. Then, we have $\mathsf{Adv}^{\text{kh-cca}}_{\Pi_{\mathsf{KFHE}},\mathsf{A}}(\lambda) = |\Pr[W_0] - 1/2|$.

$\underline{\mathsf{Game}_1}$: The same game as Game_0 except that

- the Dec oracle uses the private verifier pV_N instead of the verifier V_N when running the Dec algorithm,
- for all independent Eval queries, the Eval oracle uses the private verifier pV_N instead of the verifier V_N when running the Eval algorithm, and
- in **Challenge** phase, the challenger generates a proof π^*_N by using the semi-functional simulator sfSim_N with the membership-bit $\beta = 1$, instead of the prover P_N.

Intuitively, the partial zero-knowledge property of Π_{DN} guarantees the indistinguishability between Game_0 and Game_1. Notice that the reduction algorithm against this property does not issue statements $(\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2) \notin \mathcal{L}(R_N)$ to the given prover oracle of the partial zero-knowledge game, due to Proposition 1, that is, the true simulation-soundness of Π_{DN}.

We define Fail as the event that A issues a (dependent or independent) Eval query $(\mathsf{C}, (\mathsf{ct}^{(1)}, \dots, \mathsf{ct}^{(\ell)}))$ such that $\mathsf{Dec}_{F,1}(\mathsf{sk}_1, \widehat{\mathsf{ct}}_1) \neq \mathsf{Dec}_{F,2}(\mathsf{sk}_2, \widehat{\mathsf{ct}}_2)$ and $\mathsf{V}_N(\mathsf{crs}, (\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2), \widehat{\pi}, \emptyset) = 1$, where $\mathsf{ct}^{(i)} = (\mathsf{ct}^{(i)}_1, \mathsf{ct}^{(i)}_2, \pi^{(i)})$ for $i \in [\ell]$, $\widehat{\mathsf{ct}}_j \leftarrow \mathsf{Eval}_{F,j}(\mathsf{C}, (\mathsf{ct}^{(1)}_j, \dots, \mathsf{ct}^{(\ell)}_j))$ for $j \in \{1, 2\}$, and $\widehat{\pi} \leftarrow \mathsf{sfSim}_N(\mathsf{crs}, \mathsf{td}_s, (\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2), 1, \emptyset)$. Then, we have

$$\begin{aligned}
|\Pr[W_0] - \Pr[W_1]| &= |\Pr[\mathsf{Fail} \wedge W_0] + \Pr[\neg\mathsf{Fail} \wedge W_0] - \Pr[\mathsf{Fail} \wedge W_1] - \Pr[\neg\mathsf{Fail} \wedge W_1]| \\
&= |\Pr[\neg\mathsf{Fail}] \cdot (\Pr[W_0 \mid \neg\mathsf{Fail}] - \Pr[W_1 \mid \neg\mathsf{Fail}]) \\
&\qquad + \Pr[\mathsf{Fail}] \cdot (\Pr[W_0 \mid \mathsf{Fail}] - \Pr[W_1 \mid \mathsf{Fail}])| \\
&\leq |\Pr[W_0 \mid \neg\mathsf{Fail}] - \Pr[W_1 \mid \neg\mathsf{Fail}]| + \Pr[\mathsf{Fail}].
\end{aligned}$$

We show Lemmas 1 and 2 to estimate the upper bound of the probability of distinguishing Game_1 from Game_0.

Lemma 1. *If the event* Fail *does not occur, then any PPT adversary* A *cannot distinguish the two games* Game_0 *and* Game_1*. In particular, there exists a PPT algorithm* D^{pzk} *against the* partial zero-knowledge *property of* Π_{DN}*, such that*

$$|\Pr[W_0 \mid \neg\mathsf{Fail}] - \Pr[W_1 \mid \neg\mathsf{Fail}]| \leq \mathsf{Adv}^{\mathrm{pzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}^{pzk}}(\lambda).$$

Proof. We construct a PPT algorithm D^{pzk} against the partial zero-knowledge property of Π_{DN}, as follows: At the beginning of the KH-CCA game, D^{pzk} takes as input the CRS crs of Π_{DN} and generates $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}_{F,1}(1^\lambda)$ and $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \mathsf{KGen}_{F,2}(1^\lambda)$. It gives $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2, \mathsf{crs})$ to A. The RevHK, Eval, and Dec oracles are simulated as follows:

- $\mathsf{RevHK}()$: Given a request, obtain the simulation trapdoor td_s by invoking the reveal oracle of the partial zero-knowledge game. Return $\mathsf{sk}_h = \mathsf{td}_s$.
- $\mathsf{Eval}(\mathsf{sk}_h, \cdot)$: Given $(\mathsf{C}, (\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}))$ (where $\mathsf{ct}^{(i)} = (\mathsf{ct}_1^{(i)}, \mathsf{ct}_2^{(i)}, \pi^{(i)})$ for every $i \in [\ell]$), do the following:
  1. If the RevHK oracle has been called, then return $\perp$.
  2. If $\mathsf{ct}^{(i)} \in \mathcal{D}$ for some $i \in [\ell]$, then verify $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}$ by using V_N algorithm. If $\mathsf{ct}^{(i)} \notin \mathcal{D}$ for all $i \in [\ell]$, then verify $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}$ by using the given verifier oracle V_N^{pzk}.
  3. Compute $\widehat{\mathsf{ct}}_1$ and $\widehat{\mathsf{ct}}_2$ in the same way as the Eval algorithm, if all $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}$ pass the verification above. Return $\perp$ otherwise
  4. Obtain $\widehat{\pi}$ by issuing $((\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2), 1, \emptyset)$ to the given semi-functional simulator oracle sfSim_N^{pzk}.
  5. Return $\widehat{\mathsf{ct}} = (\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2, \widehat{\pi})$.
  6. Set $\mathcal{D} \leftarrow \mathcal{D} \cup \{\widehat{\mathsf{ct}}\}$ if $\mathsf{ct}^{(i)} \in \mathcal{D}$ for some $i \in [\ell]$.
- $\mathsf{Dec}(\mathsf{sk}_d, \cdot)$: Given $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2, \pi)$, return $\perp$ if the RevHK oracle has been invoked, or $\mathsf{ct} \in \mathcal{D}$ holds. Return $\mathsf{m} \leftarrow \mathsf{Dec}_{F,1}(\mathsf{sk}_1, \mathsf{ct}_1)$ if the verifier oracle V_N^{pzk} on input $((\mathsf{ct}_1, \mathsf{ct}_2), \pi, \emptyset)$ returns 1, and return $\perp$ otherwise.

When A submits $(\mathsf{m}_0, \mathsf{m}_1)$, D^{pzk} samples $b \xleftarrow{\$} \{0,1\}$, computes $\mathsf{ct}_1^* \leftarrow \mathsf{Enc}_{F,1}(\mathsf{pk}_1, \mathsf{m}_b; r_1^*)$ and $\mathsf{ct}_2^* \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, \mathsf{m}_b; r_2^*)$, and obtains π^* by querying $((\mathsf{ct}_1^*, \mathsf{ct}_2^*), 1, \emptyset)$ to the given prover or semi-functional simulator oracle P_N^{pzk}. Then, D^{pzk} returns $\mathsf{ct}^* = (\mathsf{ct}_1^*, \mathsf{ct}_2^*, \pi^*)$ and sets $\mathcal{D} \leftarrow \{\mathsf{ct}^*\}$.

When A outputs $b' \in \{0,1\}$, D^{pzk} outputs 1 if $b = b'$, and outputs 0 otherwise.

If the algorithm D^{pzk} simulating the Eval oracle submits $((\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2), 1, \emptyset)$ such that $(\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2) \notin \mathcal{L}(R_N)$, to sfSim_N^{pzk} oracle, then D^{pzk} fails the simulation above. This event does not occur due to the condition $[\neg\mathsf{Fail}]$. In addition, although it is forbidden for D^{pzk} to access the given verifier oracle in the partial zero-knowledge game after the simulation trapdoor td_s is revealed, both of the oracles Eval and Dec do not have to verify given ciphertexts in the KH-CCA security game. Thus, D^{pzk} simulates the environment of A correctly even after A invokes the RevHK oracle. Hence, we have $|\Pr[W_0 \mid \neg\mathsf{Fail}] - \Pr[W_1 \mid \neg\mathsf{Fail}]| \leq \mathsf{Adv}^{\mathrm{pzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}^{pzk}}(\lambda)$. ■

Lemma 2. *If the event* Fail *occurs, then there exist two PPT algorithms* D *and* F *against the* partial zero-knowledge *property and the* unbounded partial simulation-soundness*, respectively, such that* $\Pr[\mathsf{Fail}] \leq \mathsf{Adv}^{\mathrm{pzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$.

Proof. In order to show Lemma 2, we construct a PPT algorithm F^{tss} against the $\mathsf{true\ simulation\text{-}soundness}$ (see Proposition 1) of Π_{DN}, as follows: Given the CRS crs of Π_{DN}, F^{tss} gives $(\mathsf{pk}_1, \mathsf{pk}_2, \mathsf{crs})$ to A by computing $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}_{F,1}(1^\lambda)$ and $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \mathsf{KGen}_{F,2}(1^\lambda)$. F^{tss} can simulate the RevHK, Eval, and Dec oracles by using the decryption key $\mathsf{sk}_d = \mathsf{sk}_1$ and the sfSim_N oracle of the $\mathsf{true\ simulation\text{-}soundness}$ game. Then, F^{tss} can check whether the event Fail occurs, since it has the secret keys sk_1 and sk_2. If Fail happens, F^{tss} outputs the evaluated ciphertext $(\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2, \widehat{\pi}, \emptyset)$ and halts. If Fail does not happen, and A halts, then F^{tss} aborts. The output of F^{tss} satisfies the winning condition of the $\mathsf{true\ simulation\text{-}soundness}$ game since the statement $(\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2)$ such that $\mathsf{Dec}_{F,1}(\mathsf{sk}_1, \widehat{\mathsf{ct}}_1) \neq \mathsf{Dec}_{F,2}(\mathsf{sk}_2, \widehat{\mathsf{ct}}_2)$ is not in $\mathcal{L}(R_N)$, but V_N accepts $(\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2, \widehat{\pi}, \emptyset)$. Hence, the probability $\Pr[\mathsf{Fail}]$ is negligible due to the $\mathsf{true\ simulation\text{-}soundness}$ of Π_{DN}. From Proposition 1,[5] this probability is at most $\mathsf{Adv}^{\mathrm{pzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$. ■

From Lemmas 1 and 2, it holds that $|\Pr[W_0] - \Pr[W_1]| \leq \mathsf{Adv}^{\mathrm{pzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}^{pzk}}(\lambda) + \mathsf{Adv}^{\mathrm{pzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$.

$\underline{\mathsf{Game}_2}$: The same game as Game_1 except that the Eval oracle on input a dependent Eval query computes a proof on random ciphertexts $\widehat{\mathsf{ct}}_1 \leftarrow \mathsf{Add}_{F,1}(\widehat{\mathsf{ct}}'_1, \mathsf{Enc}_{F,1}(\mathsf{pk}_1, \bar{\mathsf{m}}_1))$ and $\widehat{\mathsf{ct}}_2 \leftarrow \mathsf{Add}_{F,2}(\widehat{\mathsf{ct}}'_2, \mathsf{Enc}_{F,2}(\mathsf{pk}_2, \bar{\mathsf{m}}_2))$ instead of $\widehat{\mathsf{ct}}_1 \leftarrow \mathsf{Add}_{F,1}(\widehat{\mathsf{ct}}'_1, \mathsf{Enc}_{F,1}(\mathsf{pk}_1, 0))$ and $\widehat{\mathsf{ct}}_2 \leftarrow \mathsf{Add}_{F,2}(\widehat{\mathsf{ct}}'_2, \mathsf{Enc}_{F,2}(\mathsf{pk}_2, 0))$, where $\bar{\mathsf{m}}_1, \bar{\mathsf{m}}_2 \in \mathcal{M}$ are distinct values chosen uniformly at random.

Lemma 3 shows the indistinguishability between Game_1 and Game_2, and the proof of this lemma is given in Section 3.2.2.

Lemma 3. *Assuming that all statements $(\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2)$ generated by the* Eval *algorithm are language members of $\mathcal{L}(R_N)$, then any PPT adversary* A *cannot distinguish the two games* Game_1 *and* Game_2*. The probability of distinguishing the two games is at most*

$$O(Q_{dep}) \cdot \left(\mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}_1}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}_1}(\lambda) + \mathsf{Adv}^{\mathrm{ind\text{-}cca1}}_{\Pi_{\mathsf{FHE},1},\mathsf{D}_2}(\lambda) + \mathsf{Adv}^{\mathrm{ind\text{-}cca1}}_{\Pi_{\mathsf{FHE},2},\mathsf{D}'_2}(\lambda)\right).$$

$\underline{\mathsf{Game}_3}$: Let $(\mathsf{otfGen}_N, \mathsf{otfSim}_N, \mathsf{sfV}_N)$ be a DSS-NIZK system Π_{DN} in one-time full simulation world. The same game as Game_2 except that

- the one-time full simulation generator otfGen_N of Π_{DN} is used to generate crs_N, instead of the semi-functional generator sfGen_N,
- for Dec and independent Eval queries, the semi-functional verifier sfV_N is used to check given ciphertexts, instead of the private verifier pV_N, when running the Dec and Eval algorithms, respectively, and
- in **Challenge** phase, the proof of Π_{DN} is generated by using the one-time full simulator otfSim_N, instead of the semi-functional simulator sfSim_N.

Intuitively, the indistinguishability between Game_2 and Game_3 is guaranteed by the $\mathsf{one\text{-}time\ full\ zero\text{-}knowledge}$ property of Π_{DN}. In the same way as the proof of Lemma 2, the $\mathsf{simulation\text{-}soundness}$ of sfV_N (Proposition 2) ensures that the reduction algorithm against the $\mathsf{one\text{-}time\ full\ zero\text{-}knowledge}$ of Π_{DN} does not call the semi-functional simulator oracle with $((\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2), 1, \emptyset)$ such that $(\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2) \notin \mathcal{L}(R_N)$. The following lemma shows the indistinguishability between the two games:

[5] Even in the case of strong DSS-NIZKs, Propositions 1 and 2 hold.

Lemma 4. *If* Π_{N} *satisfies both* one-time full zero-knowledge *and* unbounded partial simulation-soundness*, then any PPT adversary* A *cannot distinguish the two games* Game_2 *and* Game_3*. In particular, there exist three PPT algorithms* D^{otzk}*,* D*, and* F *such that*

$$|\Pr[W_2] - \Pr[W_3]| \leq \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}^{otzk}}(\lambda) + \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda).$$

Proof. In order to show Lemma 4, we construct a PPT algorithm D^{otzk} against the one-time full zero-knowledge property of Π_{DN}. D^{otzk} is constructed in the same way as the PPT algorithm D^{pzk} in Lemma 1 except for the procedures of the Eval oracle and the **Challenge** phase, as follows:

- $\mathsf{Eval}(\mathsf{sk}_h, \cdot)$: Given $(\mathsf{C}, (\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}))$ (where $\mathsf{ct}^{(i)} = (\mathsf{ct}_1^{(i)}, \mathsf{ct}_2^{(i)}, \pi^{(i)})$ for every $i \in [\ell]$), simulate the Eval oracle, as follows:
  - If the RevHK oracle has been called, then return $\perp$.
  - If $\mathsf{ct}^{(i)} \in \mathcal{D}$ for some $i \in [\ell]$, do the following:
    1. Verify $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}$ by using $\mathsf{V}_N(\mathsf{crs}, \cdot, \cdot, \cdot)$.
    2. If all $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}$ pass the verification above, then
       - ∗ compute $\widehat{\mathsf{ct}}_1^{(0)} \leftarrow \mathsf{Enc}_{F,1}(\mathsf{pk}_1, \bar{\mathsf{m}}_1)$, $\widehat{\mathsf{ct}}_2^{(0)} \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, \bar{\mathsf{m}}_2)$, where $\bar{\mathsf{m}}_1, \bar{\mathsf{m}}_2 \xleftarrow{\$} \mathcal{M}$.
       - ∗ compute $\widehat{\mathsf{ct}}_1' \leftarrow \mathsf{Eval}_{F,1}(\mathsf{C}, (\mathsf{ct}_1^{(1)}, \ldots, \mathsf{ct}_1^{(\ell)}))$, $\widehat{\mathsf{ct}}_2' \leftarrow \mathsf{Eval}_{F,2}(\mathsf{C}, (\mathsf{ct}_2^{(1)}, \ldots, \mathsf{ct}_2^{(\ell)}))$.
       - ∗ compute $\widehat{\mathsf{ct}}_1 \leftarrow \mathsf{Add}_{F,1}(\widehat{\mathsf{ct}}_1', \widehat{\mathsf{ct}}_1^{(0)})$, $\widehat{\mathsf{ct}}_2 \leftarrow \mathsf{Add}_{F,2}(\widehat{\mathsf{ct}}_2', \widehat{\mathsf{ct}}_2^{(0)})$.
    3. Obtain $\widehat{\pi}$ by issuing $((\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2), 1, \emptyset)$ to the given semi-functional simulator oracle sfSim_N^{otzk}.
    4. Return $\widehat{\mathsf{ct}} = (\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2, \widehat{\pi})$ and set $\mathcal{D} \leftarrow \mathcal{D} \cup \{\widehat{\mathsf{ct}}\}$.
  - If $\mathsf{ct}^{(i)} \notin \mathcal{D}$ for all $i \in [\ell]$, do the following:
    1. Verify $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}$ by using the given private or semi-functional verifier oracle V_N^{otzk}.
    2. Compute $\widehat{\mathsf{ct}}_1$ and $\widehat{\mathsf{ct}}_2$ in the same way as the Eval algorithm, if all $\mathsf{ct}^{(1)}, \ldots, \mathsf{ct}^{(\ell)}$ pass the verification above. Return $\perp$ otherwise.
    3. Obtain $\widehat{\pi}$ by issuing $((\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2), 1, \emptyset)$ to the semi-functional simulator oracle sfSim_N^{otzk}.
    4. Return $\widehat{\mathsf{ct}} = (\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2, \widehat{\pi})$.

- **Challenge** phase: When A submits $(\mathsf{m}_0, \mathsf{m}_1)$, D^{otzk} chooses $b \xleftarrow{\$} \{0,1\}$, computes $\mathsf{ct}_1^* \leftarrow \mathsf{Enc}_{F,1}(\mathsf{pk}_1, \mathsf{m}_b; r_1^*)$ and $\mathsf{ct}_2^* \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, \mathsf{m}_b; r_2^*)$, and obtains π^* by querying $((\mathsf{ct}_1^*, \mathsf{ct}_2^*), 1, \emptyset)$ to the one-time full simulator oracle otfSim_N^{otzk}. Then D^{otzk} returns $\mathsf{ct}^* = (\mathsf{ct}_1^*, \mathsf{ct}_2^*, \pi^*)$ and sets $\mathcal{D} \leftarrow \{\mathsf{ct}^*\}$.

If for all Eval queries, D^{otzk} invokes the sfSim_N^{otzk} oracle with $((\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2), 1, \emptyset)$ such that $(\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2) \notin \mathcal{L}(R_N)$ and sfV_N accepts, then the simulation above fails. In the same way as the proof of Lemma 2, the probability that this event occurs is at most $\mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$ due to Proposition 2. In addition, if A wins in the KH-CCA security game, then D^{otzk} breaks the one-time full zero-knowledge property of Π_{DN}, in the straightforward way. Notice that in the same way as D^{pzk}, D^{otzk} correctly simulates the environment of A even after the reveal event of the KH-CCA security game.

The probability of distinguishing the two games Game_2 and Game_3 is at most $\mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}^{otzk}}(\lambda) + \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$. ■

$\underline{\mathsf{Game}_4}$: The same game as Game_3 except that in **Challenge** phase, a ciphertext $\mathsf{ct}_2^* \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, \mathsf{m}_b; r_2^*)$ is replaced by $\mathsf{ct}_2^* \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, 0^{|\mathsf{m}_b|}; r_2^*)$.

By using A, it is possible to construct a PPT algorithm D^{cca1} against the $\mathsf{IND\text{-}CCA1}$ security of $\Pi_{\mathsf{FHE},2}$, which distinguishes between Game_3 and Game_4, in the straightforward way. Regarding this reduction algorithm D^{cca1}, if A can issue a Dec query $(\mathsf{ct}_1, \mathsf{ct}_2, \pi)$ such that $(\mathsf{ct}_1, \mathsf{ct}_2) \notin \mathcal{L}(R_N)$ and sfV_N accepts this query, then D^{cca1} fails to simulate the environment of A. The probability that A issues such a query is at most $\mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$, by Proposition 2. Hence, $|\Pr[W_3] - \Pr[W_4]| \leq \mathsf{Adv}^{\text{ind-cca1}}_{\Pi_{\mathsf{FHE},2},\mathsf{D}^{cca1}}(\lambda) + \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$ holds.

$\underline{\mathsf{Game}_5}$: The same game as Game_4 except that the Dec oracle returns $\mathsf{m} \leftarrow \mathsf{Dec}_{F,2}(\mathsf{sk}_2, \mathsf{ct}_2)$ if $\mathsf{sfV}_N(\mathsf{crs}, \mathsf{td}_v, (\mathsf{ct}_1, \mathsf{ct}_2), \pi, \emptyset) = 1$ holds.

We have $|\Pr[W_5] - 1/2| \leq \mathsf{Adv}^{\text{ind-cca1}}_{\Pi_{\mathsf{FHE},1},\mathsf{D}}(\lambda)$ by constructing a PPT algorithm against the $\mathsf{IND\text{-}CCA1}$ security of $\Pi_{\mathsf{FHE},1}$ in the straightforward way. Furthermore, Lemma 5 below shows the indistinguishability of games Game_4 and Game_5. The proof of this lemma is given in Section 3.2.3.

Lemma 5. *If Π_{DN} meets both of properties* one-time full zero-knowledge *and* unbounded partial simulation-soundness*, then the probability of distinguishing between Game_4 and Game_5 is negligible in λ. This probability is at most*

$$2 \cdot \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}^{otzk}}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}^{upss}}(\lambda).$$

From the discussion above, we obtain

$$\begin{aligned}\mathsf{Adv}^{\text{kh-cca}}_{\Pi_{\mathsf{KFHE}},\mathsf{A}}(\lambda) \ \leq\ & 2 \cdot \mathsf{Adv}^{\mathrm{pzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}^{pzk}}(\lambda) + O(Q_{dep}) \cdot \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}^{ss}}(\lambda) + O(Q_{dep}) \cdot \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}^{otzk}}(\lambda) \\ & + O(Q_{dep}) \cdot \mathsf{Adv}^{\text{ind-cca1}}_{\Pi_{\mathsf{FHE},1},\mathsf{D}^{cca1}_1}(\lambda) + O(Q_{dep}) \cdot \mathsf{Adv}^{\text{ind-cca1}}_{\Pi_{\mathsf{FHE},2},\mathsf{D}^{cca1}_2}(\lambda)\end{aligned}$$

and complete the proof. □

3.2.2 Proof of Lemma 3

For $j \in \{0, 1, \ldots, Q_{dep}\}$, we consider security games $\mathsf{Game}_{1,j}$, as follows: $\mathsf{Game}_{1,0}$ is the same game as Game_1. For $j \in \{0, 1, \ldots, Q_{dep} - 1\}$, let $\mathsf{Game}_{1,j+1}$ be the same game as $\mathsf{Game}_{1,j}$ except that for the $(Q_{dep} - j)$-th dependent Eval query, the Eval oracle chooses distinct $\bar{\mathsf{m}}_{j,1}, \bar{\mathsf{m}}_{j,2} \xleftarrow{\$} \mathcal{M}$ and computes $\widehat{\mathsf{ct}}^{(0)}_1 \leftarrow \mathsf{Enc}_{F,1}(\mathsf{pk}_1, \bar{\mathsf{m}}_{j,1})$, $\widehat{\mathsf{ct}}^{(0)}_2 \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, \bar{\mathsf{m}}_{j,2})$ instead of $\widehat{\mathsf{ct}}^{(0)}_1 \leftarrow \mathsf{Enc}_{F,1}(\mathsf{pk}_1, 0)$, $\widehat{\mathsf{ct}}^{(0)}_2 \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, 0)$, when generating $\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2$. Notice that $\mathsf{Game}_{1,Q_{dep}}$ is identical to Game_2.

We show that A cannot distinguish two games $\mathsf{Game}_{1,j}$ and $\mathsf{Game}_{1,j+1}$, computationally ($j \in \{0, 1, \ldots, Q_{dep} - 1\}$). To achieve this, we consider a sequence of security games $\mathsf{Game}'_0, \ldots, \mathsf{Game}'_6$.

$\underline{\mathsf{Game}'_0}$: The same game as $\mathsf{Game}_{1,j}$.

$\underline{\mathsf{Game}'_1}$: The same game as Game'_0 except that

- otfGen_N is used to generate a CRS and trapdoors of Π_{DN},
- for Dec and independent Eval queries, pV_N is replaced by sfV_N when running the Dec and Eval algorithms, respectively, and
- for the $(Q_{dep} - j)$-th dependent Eval query, the Eval oracle generates a proof of Π_{DN} by using otfSim_N instead of sfSim_N.

A PPT algorithm D^{otzk}_j breaking the one-time full zero-knowledge property of Π_{DN} can be constructed in the same way as the PPT algorithm D^{otzk} in the proof of Lemma 4.

If D_j^{otzk} issues membership-bits β which are not correct, to the given semi-functional simulator oracle sfSim_N^{otzk} and one-time simulator oracle otfSim_N^{otzk}, then it fails the simulation. During the simulation of the Eval oracle, D_j^{otzk} issues language members $(\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2)$ of $\mathcal{L}(R_N)$, due to the $\mathsf{correctness}$ of Π_{FHE} and Proposition 2, namely the $\mathsf{simulation}$-$\mathsf{soundness}$ of the semi-functional verifier. In **Challenge** phase, it also submits a language member, due to the $\mathsf{correctness}$ of Π_{FHE}. Hence, membership-bits β issued to sfSim_N^{otzk} are correct. In the same way as the proof of Lemma 2, the probability that D_j^{otzk} fails the simulation of the oracles is at most $\mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$.

Therefore, the probability of distinguishing between Game'_0 and Game'_1 is at most $\mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}_j^{otzk}}(\lambda) + \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$.

$\underline{\mathsf{Game}'_2}$: The same game as Game'_1 except that given the $(Q_{dep} - j)$-th dependent Eval query, the Eval oracle computes $\widehat{\mathsf{ct}}_2^{(0)} \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, \bar{\mathsf{m}}_{j,2})$ instead of $\widehat{\mathsf{ct}}_2^{(0)} \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, 0)$ when generating $\widehat{\mathsf{ct}}_2$, where $\bar{\mathsf{m}}_{j,2} \xleftarrow{\$} \mathcal{M}$.

The indistinguishability between Game'_1 and Game'_2 follows the IND-$\mathsf{CCA1}$ security of $\Pi_{\mathsf{FHE},2}$. In this reduction, if A issues a Dec query $(\mathsf{ct}_1, \mathsf{ct}_2, \pi)$ such that $(\mathsf{ct}_1, \mathsf{ct}_2) \notin \mathcal{L}(R_N)$ and the semi-functional verifier sfV_N accepts this query, then it can distinguish the two games. Due to Proposition 2, the probability that this event occurs is at most $\mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$.

The probability of distinguishing the two games is at most probability $\mathsf{Adv}^{\mathrm{ind\text{-}cca1}}_{\Pi_{\mathsf{FHE},2},\mathsf{D}^{cca1}}(\lambda) + \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$.

$\underline{\mathsf{Game}'_3}$: The same game as Game'_2 except that the Dec oracle returns $\mathsf{m} \leftarrow \mathsf{Dec}_{F,2}(\mathsf{sk}_2, \mathsf{ct}_2)$ if $\mathsf{sfV}_N(\mathsf{crs}, \mathsf{td}_v, (\mathsf{ct}_1, \mathsf{ct}_2), \pi, \emptyset) = 1$ holds.

The indistinguishability between Game'_2 and Game'_3 is proven in the same way as the proof of Lemma 5. Then, the probability of distinguishing the two games is at most $2 \cdot \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$.

$\underline{\mathsf{Game}'_4}$: The same game as Game'_3 except that given the $(Q_{dep} - j)$-th dependent Eval query, the Eval oracle computes $\widehat{\mathsf{ct}}_1^{(0)} \leftarrow \mathsf{Enc}_{F,1}(\mathsf{pk}_1, \bar{\mathsf{m}}_{j,1})$ instead of $\widehat{\mathsf{ct}}_1^{(0)} \leftarrow \mathsf{Enc}_{F,1}(\mathsf{pk}_1, 0)$, when generating $\widehat{\mathsf{ct}}_1$, where $\bar{\mathsf{m}}_{j,1} \xleftarrow{\$} \mathcal{M}$.

It is possible to construct a PPT algorithm which breaks IND-$\mathsf{CCA1}$ security in the straightforward way since it can simulate the environment of A by generating secret keys by itself. Thus, the IND-$\mathsf{CCA1}$ security of $\Pi_{\mathsf{FHE},1}$ guarantees the indistinguishability of the two games, and the $\mathsf{simulation}$-$\mathsf{soundness}$ of sfV_N guarantees the correctness of the simulation by the reduction algorithm against the IND-$\mathsf{CCA1}$ security. Thus, the probability of distinguishing the two games is at most $\mathsf{Adv}^{\mathrm{ind\text{-}cca1}}_{\Pi_{\mathsf{FHE},1},\mathsf{D}^{cca1}}(\lambda) + \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$.

We consider security games Game'_5 and Game'_6 which are similar to the above security games except for how to generate ciphertexts $\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2$ for the $(Q_{dep} - j)$-th dependent Eval query. Namely, the security games are defined as follows:

- Let Game'_5 be the same game as Game'_4 except that the Dec oracle returns $\mathsf{m} \leftarrow \mathsf{Dec}_{F,1}(\mathsf{sk}_1, \mathsf{ct}_1)$ instead of $\mathsf{m} \leftarrow \mathsf{Dec}_{F,2}(\mathsf{sk}_2, \mathsf{ct}_2)$, when running the Dec algorithm.

- Let Game'_6 be the same game as Game'_5 except that
  - sfGen_N is used to generate a CRS and trapdoors of Π_{DN},
  - for Dec and independent Eval queries, sfV_N is replaced by pV_N when running the Dec and Eval algorithms, respectively, and

- for the $(Q_{dep} - j)$-th dependent Eval query, the Eval oracle generates a proof of Π_{DN} by using sfSim_N instead of otfSim_N.

From the proofs above,

- the indistinguishability between Game'_4 and Game'_5 is proved in the same way as the proof of the indistinguishability between Game'_3 and Game'_2, and
- Game'_6 is indistinguishable from Game'_5 with at most probability $\mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$, due to the one-time full zero-knowledge and the unbounded partial simulation-soundness of Π_{DN}.

In addition, Game'_6 is identical to $\mathsf{Game}_{1,j+1}$.

From the discussion above, the probability of distinguishing between $\mathsf{Game}_{1,j+1}$ and $\mathsf{Game}_{1,j}$ is at most $10 \cdot \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}_1}(\lambda) + 8 \cdot \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}_1}(\lambda) + \mathsf{Adv}^{\mathrm{ind\text{-}cca1}}_{\Pi_{\mathsf{FHE},1},\mathsf{D}_2}(\lambda) + \mathsf{Adv}^{\mathrm{ind\text{-}cca1}}_{\Pi_{\mathsf{FHE},2},\mathsf{D}'_2}(\lambda)$. Therefore, A distinguishes the two games Game_1 and Game_2 with at most probability $O(Q_{dep}) \cdot (\mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}_1}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}_1}(\lambda) + \mathsf{Adv}^{\mathrm{ind\text{-}cca1}}_{\Pi_{\mathsf{FHE},1},\mathsf{D}_2}(\lambda) + \mathsf{Adv}^{\mathrm{ind\text{-}cca1}}_{\Pi_{\mathsf{FHE},2},\mathsf{D}'_2}(\lambda))$, and the proof is completed. □

3.2.3 Proof of Lemma 5

Let Bad be the event that A submits a decryption query $(\mathsf{ct}_1, \mathsf{ct}_2, \pi)$ such that $\mathsf{sfV}_N(\mathsf{crs}, (\mathsf{ct}_1, \mathsf{ct}_2), \pi, \emptyset) = 1$ and $\mathsf{Dec}_{F,1}(\mathsf{sk}_1, \mathsf{ct}_1) \neq \mathsf{Dec}_{F,2}(\mathsf{sk}_2, \mathsf{ct}_2)$. For $i \in [5]$, let Bad_i be the event that Bad occurs in Game_i.

Unless Bad occurs, Game_4 and Game_5 are identical. Thus, we have

$$\begin{aligned}|\Pr[W_4] - \Pr[W_5]| &\leq \Pr[\mathsf{Bad}_4] \\ &\leq |\Pr[\mathsf{Bad}_4] - \Pr[\mathsf{Bad}_3]| + |\Pr[\mathsf{Bad}_3] - \Pr[\mathsf{Bad}_2]| + \Pr[\mathsf{Bad}_2].\end{aligned}$$

$\Pr[\mathsf{Bad}_4] = \Pr[\mathsf{Bad}_3]$ holds because the difference between Game_3 and Game_4 does not affect whether a Dec query meets the condition of the Bad event, or not.

The indistinguishability between Bad_3 and Bad_2 follows the one-time full zero-knowledge property of Π_{DN}. It is possible to construct a PPT algorithm $\mathsf{D}^{otzk}_{\mathsf{Bad}}$ which breaks the security of Π_{DN}. This one is the same as D^{otzk} in the proof of Theorem 1, except that it aborts if Bad occurs. Thus, we have $|\Pr[\mathsf{Bad}_3] - \Pr[\mathsf{Bad}_2]| \leq \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}^{otzk}_{\mathsf{Bad}}}(\lambda) + \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}}(\lambda) + \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}}(\lambda)$.

Finally, we show that $\Pr[\mathsf{Bad}_2]$ is negligible. We can construct a PPT algorithm $\mathsf{F}^{upss}_{\mathsf{Bad}}$ against the unbounded partial simulation-soundness of Π_{DN}, as follows: By using the given oracles, it can simulate the environment of A in Game_2. If A submits a Dec query such that $\mathsf{Dec}_{F,1}(\mathsf{sk}_1, \mathsf{ct}_1) \neq \mathsf{Dec}_{F,2}(\mathsf{sk}_2, \mathsf{ct}_2)$ and the given private verifier oracle pV^{upss}_N accepts, then $\mathsf{F}^{upss}_{\mathsf{Bad}}$ outputs $((\mathsf{ct}_1, \mathsf{ct}_2), \pi, \emptyset)$ and halts. This output of $\mathsf{F}^{upss}_{\mathsf{Bad}}$ fulfills the winning condition in the partial unbounded simulation-soundness game since in the Bad event, the private verifier of Π_{DN} accepts $((\mathsf{ct}_1, \mathsf{ct}_2), \pi, \emptyset)$, and $(\mathsf{ct}_1, \mathsf{ct}_2)$ is not in the language $\mathcal{L}(R_N)$. Hence, the probability that Bad_2 occurs is at most $\mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}^{upss}_{\mathsf{Bad}}}(\lambda)$.

From the discussion above, we obtain

$$|\Pr[W_4] - \Pr[W_5]| \leq 2 \cdot \mathsf{Adv}^{\mathrm{otzk}}_{\Pi_{\mathsf{DN}},\mathsf{D}^{otzk}}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathrm{upss}}_{\Pi_{\mathsf{DN}},\mathsf{F}^{upss}}(\lambda),$$

and the proof is completed. □

4 Strong DSS-NIZK from Smooth PHPS and Unbounded Simulation-Sound NIZK

In this section, we construct a strong DSS-NIZK system for NP, constructed from a smooth PHPS and an unbounded simulation-sound NIZK.

4.1 Our Construction

Difference from the strong DSS-NIZK of [32]. Although our construction is similar to the generic construction [32] of strong DSS-NIZKs for linear subspaces, weaker properties of the underlying PHPS is sufficient in ours. As mentioned in Section 1.2, the previous construction assumes the underlying PHPS to be universal$_2$ and uses a true simulation-sound quasi-adaptive NIZK while we assume that the underlying PHPS does not have to satisfy universal$_2$, and the underlying NIZK satisfies the unbounded simulation-soundness (Definition 3).

Additional algorithms. In order to construct our strong DSS-NIZK system, we prepare additional polynomial-time algorithms E_1, E_2, E_3, $\mathcal{G}$, and $\mathcal{E}_{\mathcal{G}}$, which are defined as follows:

- $\psi \leftarrow E_1()$: E_1 samples auxiliary information ψ of a relation R, which can be regarded as witness of R.
- $1/0 \leftarrow E_2(\psi, x)$: E_2 given ψ and a statement x outputs 1 if x is in the language $\mathcal{L}(R)$, and 0 otherwise.
- $\pi \leftarrow E_3()$: E_3 samples a uniformly random value from Π.
- $(x_H; w_H) \leftarrow (\mathcal{G} \| \mathcal{E}_{\mathcal{G}})(x, \mathsf{lbl}; w)$ means that $\mathcal{G}$ given $(x, \mathsf{lbl}) \in X \times \{0,1\}^*$ outputs $x_H \in X_H$ (then, we write $x_H \leftarrow \mathcal{G}(x, \mathsf{lbl})$), and $\mathcal{E}_{\mathcal{G}}$ given w outputs a witness w_H by using the internal information of $\mathcal{G}(x, \mathsf{lbl})$. $(\mathcal{G} \| \mathcal{E}_{\mathcal{G}})(x, \mathsf{lbl}; w)$ outputs $(x_H; w_H)$ such that x_H is in the language L_H of Π_{PHPS} (and (x_H, w_H) is in the relation R_H of Π_{PHPS}) if x is in $\mathcal{L}(R)$, but x_H is not in L_H (and $(x_H, w_H) \notin R_H$) otherwise.

Furthermore, there is a gap between the two languages $\mathcal{L}(R)$ and L_H (e.g., $\mathcal{L}(R) \subset L_H$) in general. This may be a problem to construct $\mathcal{G}$. Thus, we assume that a statement x is publicly verifiable for a language L_X such that $\mathcal{L}(R) = L_H \cap L_X$, because this assumption is valid when instantiating our keyed-FHE scheme.

Validity of additional algorithms. The E_1, E_2, and E_3 algorithms are also assumed in the DSS-NIZK of [32]. Hence, we explain that assuming the remaining algorithms $\mathcal{G}, \mathcal{E}_{\mathcal{G}}$ and the public verifiability for L_X is reasonable.

In the case of constructing a strong DSS-NIZK for IND-CCA1 FHE ciphertexts, we review the language of the PHPS of [3], which can be simply defined as $L_H = \{\mathsf{ct} \mid \exists w, \mathsf{Enc}_{\mathsf{pk}}(0; w) = \mathsf{ct}\}$, where $\mathsf{Enc}_{\mathsf{pk}}(\cdot)$ is an encryption algorithm of public key encryption. In addition, we suppose that this public key encryption scheme for L_H is an IND-CCA1 secure FHE scheme from IND-CPA secure FHE schemes and a zk-SNARK [12]. Let L_X be the language for the zk-SNARK used in this IND-CCA1 secure FHE scheme [12].

First, assuming the public verifiability for L_X is reasonable because the FHE scheme [12] is based on the Naor-Yung paradigm, and it is clear that the ciphertexts are publicly verifiable for L_X.

Next, we show that assuming the $\mathcal{G}$ algorithm is reasonable. $\mathcal{G}$ checks whether two FHE ciphertexts are in L_X. If so, $\mathcal{G}$ transforms this pair into a statement in L_H by using the technique of the "onion encryption" of [37] [6]. Notice that just computing $\mathsf{ct}_1 - \mathsf{ct}_2$ is possible as the statement, in the case where the FHE scheme is some lattice-based one such as [8–11, 15, 26, 28]. Otherwise, it samples $x_H \notin L_H$ and outputs this. Hence, if two ciphertexts are in $\mathcal{L}(R)$, then this pair is also in L_H. Otherwise, it is not in L_H due to the public verifiability of the $\mathsf{IND\text{-}CCA1}$ secure FHE scheme. Hence, $\mathcal{G}$ fulfills the required property. Accordingly, there exits an algorithm $\mathcal{E}_\mathcal{G}$ generating the corresponding witness (concretely, the randomness used in the onion encryption) by using the $\mathcal{G}$ algorithm. Hence, there exist $\mathcal{G}$ and $\mathcal{E}_\mathcal{G}$.

Our strong DSS-NIZK system. In order to construct our strong DSS-NIZK system Π_{DN}, we employ the following primitives:

- An ϵ-smooth labeled PHPS Π_{PHPS} with a public evaluation function $\hat{H}$, which is constituted by a PHF $\mathbf{H} = (H, K, X_H, L_H, \Pi, S, \alpha)$.
- A NIZK system $\Pi_{\mathsf{N}} = (\mathsf{Gen}_N, \mathsf{P}_N, \mathsf{V}_N)$ for an augmented relation $R_N = \{((x, x_H, \pi_H, \mathsf{lbl}), (w, w_H)) \mid (x, w) \in R \wedge \pi_H = \hat{H}(\alpha(k), (x_H, x \| \mathsf{lbl}), w_H)\}$, with a PPT simulator $(\mathsf{Sim}_{N,0}, \mathsf{Sim}_{N,1})$ (where $R \subseteq X \times W$ is the relation of Π_{DN}).
- The above algorithms E_1, E_2, E_3, $\mathcal{G}$, and $\mathcal{E}_\mathcal{G}$.

Our DSS-NIZK system Π_{DN} for a relation R is described as follows:

Real World consists of

- $\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda)$: Sample $k \overset{\$}{\leftarrow} K$ and compute $\mathsf{crs}_N \leftarrow \mathsf{Gen}_N(\lambda)$. Output $\mathsf{crs} = (\alpha(k), \mathsf{crs}_N)$.
- $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w, \mathsf{lbl})$: Compute $(x_H; w_H) \leftarrow (\mathcal{G} \| \mathcal{E}_\mathcal{G})(x, \mathsf{lbl}; w)$, $\pi_H \leftarrow \hat{H}(\alpha(k), (x_H, x \| \mathsf{lbl}), w_H)$ and $\pi_N \leftarrow \mathsf{P}_N(\mathsf{crs}_N, (x, x_H, \pi_H, \mathsf{lbl}), (w, w_H))$. Output $\pi = (x_H, \pi_H, \pi_N)$
- $1/0 \leftarrow \mathsf{V}(\mathsf{crs}, x, \pi, \mathsf{lbl})$: Output 1 if $\mathsf{V}_N(\mathsf{crs}_N, (x, x_H, \pi_H, \mathsf{lbl}), \pi_N) = 1$. Output 0 otherwise.

Partial Simulation World consists of

- $(\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_v) \leftarrow \mathsf{sfGen}(1^\lambda)$: Sample $\psi \leftarrow E_1()$ and $k \overset{\$}{\leftarrow} K$, and compute $(\mathsf{crs}_N, \mathsf{td}_N) \leftarrow \mathsf{Sim}_{N,0}(1^\lambda)$. Output $\mathsf{crs} = (\alpha(k), \mathsf{crs}_N)$, $\mathsf{td}_s = (k, \mathsf{td}_N)$, and $\mathsf{td}_v = (\psi, k)$.
- $\pi \leftarrow \mathsf{sfSim}(\mathsf{crs}, \mathsf{td}_s, x, \beta, \mathsf{lbl})$:
  - If $\beta = 1$, then compute $x_H \leftarrow \mathcal{G}(x, \mathsf{lbl})$, $\pi_H \leftarrow H_k(x_H, x \| \mathsf{lbl})$ and $\pi_N \leftarrow \mathsf{Sim}_{N,1}(\mathsf{crs}_N, \mathsf{td}_N, (x, x_H, \pi_H, \mathsf{lbl}))$.
  - If $\beta = 0$, then sample a uniformly random $\pi_H \leftarrow E_3()$ and compute $x_H \leftarrow \mathcal{G}(x, \mathsf{lbl})$ and $\pi_N \leftarrow \mathsf{Sim}_{N,1}(\mathsf{crs}_N, \mathsf{td}_N, (x, x_H, \pi_H, \mathsf{lbl}))$.

  Output $\pi = (x_H, \pi_H, \pi_N)$.
- $1/0 \leftarrow \mathsf{pV}(\mathsf{crs}, \mathsf{td}_v, x, \pi, \mathsf{lbl})$: Output 1 if $E_2(\psi, x) = 1$ (namely, it holds that $x \in \mathcal{L}(R_N)$), $H_k(x_H, x \| \mathsf{lbl}) = \pi_H$, and $\mathsf{V}_N(\mathsf{crs}_N, (x, x_H, \pi_H, \mathsf{lbl}), \pi_N) = 1$. Output 0 otherwise.

One-time Full Simulation World consists of

- $(\mathsf{crs}, \mathsf{td}_s, \mathsf{td}_{s,1}, \mathsf{td}_v) \leftarrow \mathsf{otfGen}(1^\lambda)$: Sample $k \overset{\$}{\leftarrow} K$ and compute $(\mathsf{crs}_N, \mathsf{td}_N) \leftarrow \mathsf{Sim}_{N,0}(1^\lambda)$. Output $\mathsf{crs} = (\alpha(k), \mathsf{crs}_N)$, $\mathsf{td}_s = \mathsf{td}_{s,1} = (k, \mathsf{td}_N)$, and $\mathsf{td}_v = k$.

[6] Concretely, two FHE ciphertexts $\mathsf{Enc}(\mathsf{pk}_1, \mathsf{m}_1)$ and $\mathsf{Enc}(\mathsf{pk}_2, \mathsf{m}_2)$ can be transformed into a ciphertext $\mathsf{Enc}(\mathsf{pk}_1, \mathsf{Enc}(\mathsf{pk}_2, \mathsf{m}_1 - \mathsf{m}_2))$. If for two FHE ciphertexts $\mathsf{Enc}(\mathsf{pk}_1, \mathsf{m}_1; r_1)$ and $\mathsf{Enc}(\mathsf{pk}_2, \mathsf{m}_2; r_2)$, (m, r_1, r_2) where $\mathsf{m} = \mathsf{m}_1 = \mathsf{m}_2$ is a witness of the Naor-Yung language, then $\mathsf{Enc}(\mathsf{pk}_1, \mathsf{Enc}(\mathsf{pk}_2, \mathsf{m}_1 - \mathsf{m}_2))$ is a statement in L_H.

- $\pi \leftarrow \mathsf{otfSim}(\mathsf{crs}, \mathsf{td}_{s,1}, x, \mathsf{lbl})$: Compute $x_H \leftarrow \mathcal{G}(x, \mathsf{lbl})$, $\pi_H \leftarrow H_k(x_H, x\|\mathsf{lbl})$, and $\pi_N \leftarrow \mathsf{Sim}_{N,1}(\mathsf{crs}_N, \mathsf{td}_N, (x, x_H, \pi_H, \mathsf{lbl}))$. Output $\pi = (x_H, \pi_H, \pi_N)$.
- $1/0 \leftarrow \mathsf{sfV}(\mathsf{crs}, \mathsf{td}_v, x, \pi, \mathsf{lbl})$: Output 1 if it holds that $H_k(x_H, x\|\mathsf{lbl}) = \pi_H$ and $\mathsf{V}_N(\mathsf{crs}_N, (x, x_H, \pi_H, \mathsf{lbl}), \pi_N) = 1$. Output 0 otherwise.

4.2 Security Analysis

Theorem 2 shows the properties of Π_{DN}.

Theorem 2. *If the labeled PHPS Π_{PHPS} is ϵ-*smooth*, and the NIZK Π_{N} satisfies* unbounded simulation-soundness*, then the resulting NIZK system Π_{DN} is a strong DSS-NIZK system.*

The overview of the proof for this theorem is as follows: We prove Theorem 2 By showing Lemmas 6, 7, 8, and 9. The completeness of Π_{DN} is ensured by that of the underlying NIZK and the definition of the underlying labeled PHPS (see Lemma 6). The partial zero-knowledge and unbounded partial simulation-soundness of Π_{DN} (Lemmas 7 and 8, respectively) can be proven in the same way as the proof of [32, Theorem 21]. Thus, we omit to describe the proofs of Lemmas 7 and 8.

Lemma 6 (Completeness of Π_{DN})**.** *If the PHF* **H** *constitutes the labeled PHPS Π_{PHPS} and the NIZK system Π_{N} satisfies* compleneteness*, then the proposed strong DSS-NIZK system Π_{DN} satisfies* Completeness*.*

Proof. We show that Π_{DN} satisfies completeness. If the public evaluation function $\hat{H}$ of the underlying PHPS correctly computes $\pi_H = \hat{H}(\alpha(k), (x_H, x\|\mathsf{lbl}), w_H)$, and x is in $\mathcal{L}(R)$, then the prover algorithm P_N computes a correct proof π_N since $(x, x_H, \pi_H, \mathsf{lbl})$ is in the language $\mathcal{L}(R_N)$. Thus, the completeness of Π_{DN} follows the completeness of the underlying NIZK. ■

Lemma 7 (Partial zero-knowledge of Π_{DN})**.** *If the NIZK system Π_{N} satisfies* composable zero-knowledge *property and* unbounded simulation-soundenss*, then the resulting strong DSS-NIZk system Π_{DN} satisfies* partial zero-knowledge *property.*

Lemma 8 (Unbounded partial simulation-soundness of Π_{DN})**.** *If the failure probability of the E_2 algorithm for the relation R is negligible in λ, then the resulting strong DSS-NIZK system Π_{DN} satisfies* unbounded partial simulation-soundness*.*

Finally, we give an explanation of the proof of one-time full zero-knowledge because the difference between the proofs of [32] and Theorem 2 is this proof, namely the proof of Lemma 9. In the one-time full zero-knowledge game, an adversary is allowed to submit $(x^*, \beta^*, \mathsf{lbl}^*)$ such that $x^* \notin \mathcal{L}(R)$ in order to get a proof π^* generated by sfSim or otfSim. The difference between pV and sfV is the verification of $x \in \mathcal{L}(R)$ with E_2. Thus, the outputs of pV and sfV may be different if the adversary issues (x, π, lbl) to the given verifier oracle, such that $x \notin \mathcal{L}(R)$, $(x, \pi, \mathsf{lbl}) \neq (x^*, \pi^*, \mathsf{lbl}^*)$, and the verifier oracle accepts. In the proof of [32], it is proven that this event does not occur due to the universal$_2$ property of Π_{PHPS} and a special property of the underlying NIZK. In our proof, the event occurs with negligible probability, due to the unbounded simulation-soundness of Definition 3. This is because $((x^*, x_H^*, \pi_H^*, \mathsf{lbl}^*), \pi^*)$ is included in the list $\mathcal{Q}$ of the unbounded simulation-soundness game of Π_{N}, and issuing the query above $(x, \pi = (x_H, \pi_H, \pi_N), \mathsf{lbl})$ corresponds to the adversary's winning condition in Definition 3 (i.e., $(x, x_H, \pi_H, \mathsf{lbl}) \notin \mathcal{L}(R_N)$, $((x, x_H, \pi_H, \mathsf{lbl}), \pi_N) \notin \mathcal{Q}$, and $\mathsf{V}_N(\mathsf{crs}_N, (x, x_H, \pi_H, \mathsf{lbl}), \pi_N) = 1$). Therefore, Π_{PHPS} does not need to satisfy universal$_2$ property, and Π_{N} must fulfill the unbounded simulation-soundness.

Lemma 9 (One-time full zero-knowledge of Π_{DN})**.** *If the labeled PHPS* Π_{PHPS} *is* ϵ*-smooth and the NIZK system* Π_{N} *satisfies* unbounded simulation-soundness*, then the resulting strong DSS-NIZK system* Π_{DN} *satisfies* one-time full zero-knowledge *property.*

Proof. We prove that Π_{DN} satisfies one-time full zero-knowledge. We consider a sequence of security games. Game_0 is identical to the one-time full zero-knowledge game in the partial simulation world. Let Game_1 be the same game as Game_0 except that the proof of $(x^*, \beta^*, \mathsf{lbl}^*)$ is generated by otfSim instead of sfSim. If β^* is not correct for x^*, then the challenger aborts in both of the two games. Thus, we assume that β^* is correct for x^*. In the case $\beta^* = 1$, sfSim and otfSim are identical. In the case $\beta^* = 0$, π_H generated by sfSim is uniformly at random while π_H generated by otfSim is $H_k(x_H, x\|\mathsf{lbl})$. Due to the ϵ-smoothness of Π_{PHPS}, the statistical distance between the distributions of the two proofs is at most ϵ. Hence, the adversary distinguishes between Game_0 and Game_1 with at most probability ϵ.

Game_2 is the same game as Game_1 except that the private verifier oracle pV is replaced by the semi-functional verifier oracle sfV. Let N be the number of queries issued to the private or semi-functional verifier oracle. Let $\mathsf{Game}_{1,0}$ and $\mathsf{Game}_{1,N}$ be the same games as Game_1 and Game_2, respectively. For each $i \in \{0, 1, \ldots, N-1\}$, we consider a security game $\mathsf{Game}_{1,i+1}$ in which the verifier oracle returns the output of sfV for the $(N-i)$-th query issued to the verifier oracle, and it returns that of pV for the j-th query ($j \in \{1, \ldots, N-i-1\}$). We prove that A cannot distinguish between $\mathsf{Game}_{1,i}$ and $\mathsf{Game}_{1,i+1}$ due to the unbounded simulation-soundness of the underlying NIZK. To prove this fact, it is sufficient to consider the event in which A issues the $(N-i)$-th query $(x, \pi = (x_H, \pi_H, \pi_N), \mathsf{lbl})$ to the given verifier oracle, such that $(x, \pi, \mathsf{lbl}) \neq (x^*, \pi^*, \mathsf{lbl}^*)$, $x \notin \mathcal{L}(R)$, and the sfV oracle accepts (i.e., $H_k(x_H, x\|\mathsf{lbl}) = \pi_H$ and $\mathsf{V}_N(\mathsf{crs}_N, (x, x_H, \pi_H, \mathsf{lbl}), \pi_N) = 1$ hold). The reasons for this are as follows:

- In the case $(x, \pi, \mathsf{lbl}) = (x^*, \pi^*, \mathsf{lbl}^*)$, both of the two games are aborted.
- In the case $x \in \mathcal{L}(R)$, both of the verifier oracles in $\mathsf{Game}_{1,i}$ and $\mathsf{Game}_{1,i+1}$ return the same output since the difference between pV and sfV is only the verification by E_2.
- In the case $H_k(x_H, x\|\mathsf{lbl}) \neq \pi_H$, both pV and sfV return 0.
- In the case $\mathsf{V}_N(\mathsf{crs}_N, (x, x_H, \pi_H, \mathsf{lbl}), \pi_N) = 0$, both pV and sfV return 0.

Hence, $\mathsf{Game}_{1,i}$ and $\mathsf{Game}_{1,i+1}$ are identical unless A submits that query. Then, we show that it is possible to break the unbounded simulation-soundness of Π_{N} if A issues the $(N-i)$-th query above. Let $\mathcal{Q}$ be the list of queries and responses in the unbounded simulation-soundness game of Π_{N}. Notice that, in the reduction from the property of Π_{N}, $((x^*, x_H^*, \pi_H^*, \mathsf{lbl}^*), \pi_N^*)$ is included in $\mathcal{Q}$. If A issues the query meeting the additional condition $((x, x_H, \pi_H, \mathsf{lbl}), \pi_N) \notin \mathcal{Q}$, then this query clearly fulfills the winning condition of the simulation-soundness game of Π_{N}. Thus, we consider the additional condition $((x, x_H, \pi_H, \mathsf{lbl}), \pi_N) \in \mathcal{Q}$. If $((x, x_H, \pi_H, \mathsf{lbl}), \pi_N) \in \mathcal{Q}\backslash\{((x^*, x_H^*, \pi_H^*, \mathsf{lbl}^*), \pi_N^*)\}$, then there does not exist the query meeting the condition $x \notin \mathcal{L}(R)$, $H_k(x_H, x\|\mathsf{lbl}) = \pi_H$, and $\mathsf{V}_N(\mathsf{crs}_N, (x, x_H, \pi_H, \mathsf{lbl}), \pi_N) = 1$. The reason for this is as follows: If $x \in \mathcal{L}(R)$, this contradicts the assumption $x \notin \mathcal{L}(R)$ of the $(N-i)$-th query. If $x \notin \mathcal{L}(R)$, then the proof $\pi = (x_H, \pi_H, \pi_N)$ is invalid since the sfSim_N oracle samples π_H uniformly at random, and $H_k(x_H, x\|\mathsf{lbl}) \neq \pi_H$ holds with overwhelming probability. Hence, the $(N-i)$-th query such that $((x, x_H, \pi_H, \mathsf{lbl}), \pi_N) \in \mathcal{Q}\backslash\{((x^*, x_H^*, \pi_H^*, \mathsf{lbl}^*), \pi_N^*)\}$ does not meet the above condition $x \notin \mathcal{L}(R)$, $H_k(x_H, x\|\mathsf{lbl}) = \pi_H$, and $\mathsf{V}_N(\mathsf{crs}_N, (x, x_H, \pi_H, \mathsf{lbl}), \pi_N) = 1$. Hence, A must issue the $(N-i)$-th query such that $((x, x_H, \pi_H, \mathsf{lbl}), \pi_N) \notin \mathcal{Q}$, $x \notin \mathcal{L}(R)$, and the sfV oracle accepts in order to distinguish the two games. That is, if A issues the $(N-i)$-th query such that $(x, \pi, \mathsf{lbl}) \neq (x^*, \pi^*, \mathsf{lbl}^*)$,

$x \notin \mathcal{L}(R)$, $H_k(x_H, x\|\mathsf{lbl}) = \pi_H$, and $\mathsf{V}_N(\mathsf{crs}_N, (x, x_H, \pi_H, \mathsf{lbl}), \pi_N) = 1$, then this query satisfies the winning condition of the unbounded simulation-soundness game of Π_N (i.e., $((x, x_H, \pi_H, \mathsf{lbl}), \pi_N) \notin \mathcal{Q}$, $(x, x_H, \pi_H, \mathsf{lbl}) \notin \mathcal{L}(R_N)$, and $\mathsf{V}_N(\mathsf{crs}_N, (x, x_H, \pi_H, \mathsf{lbl}), \pi_N) = 1$). Therefore, the indistinguishability between $\mathsf{Game}_{1,i}$ and $\mathsf{Game}_{1,i+1}$ follows the property of Π_N, and the difference between success probabilities in Game_1 and Game_2 is at most $N \cdot \mathsf{Adv}^{\mathrm{uss}}_{\Pi_\mathsf{N}}(\lambda)$, where $\mathsf{Adv}^{\mathrm{uss}}_{\Pi_\mathsf{N}}(\lambda)$ is the maximum probability that any PPT algorithm breaks the unbounded simulation-soundness of Π_N.

Game_3 is identical to Game_2 except that sfGen is replaced by otfGen at the beginning of the one-time full zero-knowledge game. The difference between the two generators is whether ψ is generated or not. Game_2 and Game_3 are identical since ψ is not used in both of the two games.

From the discussion above, the adversary breaks the one-time full zero knowledge property of Π_DN with at most probability $\epsilon + N \cdot \mathsf{Adv}^{\mathrm{uss}}_{\Pi_\mathsf{N}}(\lambda)$, and the proof is completed. ■

Due to Lemmas 6, 7, 8, and 9, the proof of Theorem 2 is completed.

5 Instantiation of Our Construction

In this section, we show that all building blocks of our generic construction of keyed-FHE do not require iO, by discussing existing schemes which we can apply to our construction.

Although our generic construction of keyed-FHE requires only the IND-CCA1 security for the underlying FHE scheme, our strong DSS-NIZK system requires public verifiability for the IND-CCA1 secure FHE scheme, as discussed in Section 4.1. There is an IND-CCA1 secure publicly verifiable FHE scheme [12] under zk-SNARK [4, 5]. We can also construct strong DSS-NIZK using the following building blocks: (1) the NIZK systems for NP in the (quantum) random oracle model [13, 24] or the standard model [34], and (2) the smooth PHPS [3] for lattice-based ciphertexts. Therefore, we can obtain a keyed-FHE scheme secure in the (quantum) random oracle model or the standard model. These details are discussed below.

Canetti et al. [12] proposed generic constructions of IND-CCA1 secure FHE. One of these constructions is based on the Naor-Yung paradigm [40] with two IND-CPA secure FHE schemes and a zk-SNARK [4,5]. This one satisfies both IND-CCA1 security and public verifiability of ciphertexts, since it is possible to check the validity of ciphertexts owing to the public verifiability of the underlying zk-SNARK. Although they also proposed other generic constructions of IND-CCA1 secure FHE, these ones need iO or do not necessarily satisfy public verifiability. Hence, we have chosen the generic construction based on the Naor-Yung paradigm and instantiate this generic construction by using existing lattice-based FHE schemes such as [8–11,15,26,28] and existing zk-SNARKs such as [2,4,5,25,30,39,45].

The remaining part is strong DSS-NIZK. NIZKs used to obtain a strong DSS-NIZK for the Naor-Yung language can be constructed from a Σ-protocol [29] by using the Fiat-Shamir transformation [24], and there exist such NIZKs in the quantum random oracle model [13] and the standard model [34]. There exists a smooth (approximate) PHPS [3] for lattice-based ciphertexts. In addition, it is possible to instantiate the $\mathcal{G}$ and $\mathcal{E}_\mathcal{G}$ algorithms (see Section 4.1) which generate a statement-witness pair of the smooth PHPS [3], by using the IND-CCA1 FHE ciphertexts of [12]. These algorithms are constructed in the same way as those described in Section 4.1. Hence, we can obtain a strong DSS-NIZK for IND-CCA1 secure (publicly verifiable) FHE ciphertexts by using existing schemes.

To sum up, we can apply the following existing schemes to the above generic constructions of public verifiable IND-CCA1 secure FHE and strong DSS-NIZK:

- Existing schemes applied to the publicly verifiable IND-CCA1 secure FHE [12]:

  - $\mathsf{IND\text{-}CPA}$ secure FHE based on the LWE assumption, such as [8–11, 15, 26, 28].
  - zk-SNARKs for arithmetic circuits based on knowledge assumptions, such as [2, 4, 5, 25, 30, 39, 45], or zk-SNARKs for NP in the quantum random oracle model [16].

- Existing schemes applied to our strong DSS-NIZK:
  - Statistically secure smooth PHPS [3].
  - Unbounded simulation-sound NIZK such as the NIZK system for NP in the random oracle model from a Σ-protocol [29] using the Fiat-Shamir transformation [24], the NIZK system secure in the quantum random oracle model [13], or the NIZK system secure in the standard model [34][7].

Therefore, there exists a keyed-FHE scheme constructed from simpler primitives than iO, and its security is based on a knowledge assumption since such an assumption is necessary to ensure the security of a zk-SNARK.

6 Conclusion

In this paper, we proposed a keyed-FHE scheme constructed from simpler primitives than a strong primitive iO used in the existing keyed-FHE scheme [33]. To this end, we proposed a generic construction of keyed-FHE by using two publicly verifiable $\mathsf{IND\text{-}CCA1}$ secure FHE schemes and a strong DSS-NIZK system. In addition, we gave a generic construction of strong DSS-NIZK, which is constructed from a smooth PHPS and an unbounded simulation-sound NIZK. Furthermore, we showed that existing primitives can be applied to our generic constructions. As a result, we can obtain a keyed-FHE scheme constructed from simpler primitives than iO.

Acknowledgments: This work was supported by JST CREST Grant Number JPMJCR2113, Japan, and JSPS KAKENHI Grant Numbers 19K20267 and 21K11897. We would like to thank Prof. Thomas Peters since he gave us insightful suggestion to instantiate our keyed-FHE scheme in the standard model. The main part of this research was done when the second author, Keita Emura, was with the National Institute of Information and Communications Technology (NICT), Japan.

References

[1] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *CRYPTO*, volume 2139 of *LNCS*, pages 1–18. Springer, 2001.

[2] C. Baum, J. Bootle, A. Cerulli, R. del Pino, J. Groth, and V. Lyubashevsky. Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. In *CRYPTO (2)*, volume 10992 of *LNCS*, pages 669–699. Springer, 2018.

[3] F. Benhamouda, O. Blazy, L. Ducas, and W. Quach. Hash proof systems over lattices revisited. In *Public Key Cryptography (2)*, volume 10770 of *LNCS*, pages 644–674. Springer, 2018.

[7] Libert et al. proposed a simulation-sound NIZK system for LWE-like relations in the standard model [34] and do not give a security proof that it satisfies the zero-knowledge property after the trapdoor is revealed. Nevertheless, since their zero-knowledge property is statistical, it can be applied to our construction. However, their scheme is not very efficient, and thus it would be interesting to see that the efficiency of their NIZKs can be improved in future work.

[4] N. Bitansky, R. Canetti, A. Chiesa, S. Goldwasser, H. Lin, A. Rubinstein, and E. Tromer. The hunting of the SNARK. *J. Cryptol.*, 30(4):989–1066, 2017.

[5] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer. Recursive composition and bootstrapping for SNARKS and proof-carrying data. In *STOC*, pages 111–120. ACM, 2013.

[6] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *CRYPTO*, volume 7417 of *LNCS*, pages 868–886. Springer, 2012.

[7] Z. Brakerski, D. Cash, R. Tsabary, and H. Wee. Targeted homomorphic attribute-based encryption. In *TCC (B2)*, volume 9986 of *LNCS*, pages 330–360, 2016.

[8] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325. ACM, 2012.

[9] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, pages 97–106. IEEE Computer Society, 2011.

[10] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *CRYPTO*, volume 6841 of *LNCS*, pages 505–524. Springer, 2011.

[11] Z. Brakerski and V. Vaikuntanathan. Lattice-based FHE as secure as PKE. In *ITCS*, pages 1–12. ACM, 2014.

[12] R. Canetti, S. Raghuraman, S. Richelson, and V. Vaikuntanathan. Chosen-ciphertext secure fully homomorphic encryption. In *Public Key Cryptography (2)*, volume 10175 of *LNCS*, pages 213–240. Springer, 2017.

[13] M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, and G. Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *CCS*, pages 1825–1842. ACM, 2017.

[14] J. H. Cheon, J. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun. Batch fully homomorphic encryption over the integers. In *EUROCRYPT*, volume 7881 of *LNCS*, pages 315–335. Springer, 2013.

[15] J. H. Cheon, A. Kim, M. Kim, and Y. S. Song. Homomorphic encryption for arithmetic of approximate numbers. In *ASIACRYPT (1)*, volume 10624 of *LNCS*, pages 409–437. Springer, 2017.

[16] A. Chiesa, P. Manohar, and N. Spooner. Succinct arguments in the quantum random oracle model. In *TCC (2)*, volume 11892 of *LNCS*, pages 1–29. Springer, 2019.

[17] M. Clear and C. McGoldrick. Multi-identity and multi-key leveled FHE from learning with errors. In *CRYPTO (2)*, volume 9216 of *LNCS*, pages 630–656. Springer, 2015.

[18] J. Coron, A. Mandal, D. Naccache, and M. Tibouchi. Fully homomorphic encryption over the integers with shorter public keys. In *CRYPTO*, volume 6841 of *LNCS*, pages 487–504. Springer, 2011.

[19] R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT*, volume 2332 of *LNCS*, pages 45–64. Springer, 2002.

[20] Y. Desmedt, V. Iovino, G. Persiano, and I. Visconti. Controlled homomorphic encryption: Definition and construction. In *Financial Cryptography Workshops*, volume 10323 of *LNCS*, pages 107–129. Springer, 2017.

[21] K. Emura. On the security of keyed-homomorphic PKE: preventing key recovery attacks and ciphertext validity attacks. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 104-A(1):310–314, 2021.

[22] K. Emura, G. Hanaoka, K. Nuida, G. Ohtake, T. Matsuda, and S. Yamada. Chosen ciphertext secure keyed-homomorphic public-key cryptosystems. *Des. Codes Cryptogr.*, 86(8):1623–1683, 2018.

[23] K. Emura, G. Hanaoka, G. Ohtake, T. Matsuda, and S. Yamada. Chosen ciphertext secure keyed-homomorphic public-key encryption. In *Public Key Cryptography*, volume 7778 of *LNCS*, pages 32–50. Springer, 2013.

[24] S. Faust, M. Kohlweiss, G. A. Marson, and D. Venturi. On the non-malleability of the fiat-shamir transform. In *INDOCRYPT*, volume 7668 of *LNCS*, pages 60–79. Springer, 2012.

[25] R. Gennaro, C. Gentry, B. Parno, and M. Raykova. Quadratic span programs and succinct nizks without pcps. In *EUROCRYPT*, volume 7881 of *LNCS*, pages 626–645. Springer, 2013.

[26] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178. ACM, 2009.

[27] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO (1)*, volume 8042 of *LNCS*, pages 75–92. Springer, 2013.

[28] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO (1)*, volume 8042 of *LNCS*, pages 75–92. Springer, 2013.

[29] I. Giacomelli, J. Madsen, and C. Orlandi. Zkboo: Faster zero-knowledge for boolean circuits. In *USENIX Security Symposium*, pages 1069–1083. USENIX Association, 2016.

[30] J. Groth. On the size of pairing-based non-interactive arguments. In *EUROCRYPT (2)*, volume 9666 of *LNCS*, pages 305–326. Springer, 2016.

[31] C. S. Jutla and A. Roy. Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In *CRYPTO (2)*, volume 8617 of *LNCS*, pages 295–312. Springer, 2014.

[32] C. S. Jutla and A. Roy. Dual-system simulation-soundness with applications to UC-PAKE and more. In *ASIACRYPT (1)*, volume 9452 of *LNCS*, pages 630–655. Springer, 2015.

[33] J. Lai, R. H. Deng, C. Ma, K. Sakurai, and J. Weng. Cca-secure keyed-fully homomorphic encryption. In *Public Key Cryptography (1)*, volume 9614 of *LNCS*, pages 70–98. Springer, 2016.

[34] B. Libert, K. Nguyen, A. Passelègue, and R. Titiu. Simulation-sound arguments for LWE and applications to KDM-CCA2 security. In *ASIACRYPT (1)*, volume 12491 of *LNCS*, pages 128–158. Springer, 2020.

[35] B. Libert, T. Peters, M. Joye, and M. Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and cca2-secure encryption from homomorphic signatures. In *EUROCRYPT*, volume 8441 of *LNCS*, pages 514–532. Springer, 2014.

[36] J. Loftus, A. May, N. P. Smart, and F. Vercauteren. On cca-secure somewhat homomorphic encryption. In *Selected Areas in Cryptography*, volume 7118 of *LNCS*, pages 55–72. Springer, 2011.

[37] A. López-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *STOC*, pages 1219–1234. ACM, 2012.

[38] Y. Maeda and K. Nuida. Chosen ciphertext secure keyed two-level homomorphic encryption. In *ACISP*, volume 13494 of *LNCS*, pages 209–228. Springer, 2022.

[39] M. Maller, S. Bowe, M. Kohlweiss, and S. Meiklejohn. Sonic: Zero-knowledge snarks from linear-size universal and updatable structured reference strings. In *CCS*, pages 2111–2128. ACM, 2019.

[40] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437. ACM, 1990.

[41] A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553. IEEE Computer Society, 1999.

[42] S. Sato, K. Emura, and A. Takayasu. Keyed-fully homomorphic encryption without indistinguishability obfuscation. In *ACNS*, volume 13269 of *LNCS*, pages 3–23. Springer, 2022.

[43] H. Shinoki and K. Nuida. On extension of evaluation algorithms in keyed-homomorphic encryption. In *IWSEC*, volume 13504 of *LNCS*, pages 189–207. Springer, 2022.

[44] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT*, volume 6110 of *LNCS*, pages 24–43. Springer, 2010.

[45] Y. Zhang, A. Szepieniec, R. Zhang, S. Sun, G. Wang, and D. Gu. Voproof: Efficient zksnarks from vector oracle compilers. In *CCS*, pages 3195–3208. ACM, 2022.

A Keyed-FHE from IND-CPA secure FHE, zk-SNARK, and Strong DSS-NIZK

We describe the a keyed-FHE scheme constructed from IND-CPA secure FHE, zk-SNARK, and strong DSS-NIZK, by using a generic construction of IND-CCA1 secure FHE [12] and our keyed-FHE scheme in Section 3. When replacing IND-CCA1 secure FHE schemes used in the keyed-FHE scheme in Section 3, with IND-CPA secure FHE schemes, we can construct a keyed-FHE scheme by adding a zk-SNARK system. Namely, we can obtain a generic construction starting from IND-CPA secure FHE, zk-SNARK, and strong DSS-NIZK.

To show this fact, we describe the definition of zk-SNARKs by following [4].

Definition 17 (zk-SNARK)**.** *A zk-SNARK system for a relation* $R \subseteq \{0,1\}^* \times \{0,1\}^*$ *consists of three polynomial-time algorithms* $(\mathsf{Gen}, \mathsf{P}, \mathsf{V})$*: Let* $\mathcal{L}(R) = \{x \mid \exists w \text{ s.t. } (x,w) \in R\}$ *be the language defined by* R*.*

- $(\mathsf{crs}, \mathsf{vrs}) \leftarrow \mathsf{Gen}(1^\lambda)$*: The randomized algorithm* Gen *takes as input a security parameter* 1^λ*, and it outputs a CRS* crs *and a verification key* vrs*.*

- $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$*: The randomized algorithm* P *takes as input a CRS* crs*, a statement* x*, and a witness* w*, and it outputs a proof* π*.*

- $1/0 \leftarrow \mathsf{V}(\mathsf{vrs}, x, \pi)$*: The deterministic algorithm* V *takes as input a verification key* vrs*, a statement* x*, and a proof* π*, and it outputs* 1 *or* 0*.*

It is required that a zk-SNARK satisfies completeness*,* knowledge-soundness*,* zero-knowledge*, and* succinctness*:*

Completeness. *For every* $(x, w) \in R$*, it holds that*

$$\Pr\left[\begin{array}{l}(\mathsf{crs}, \mathsf{vrs}) \leftarrow \mathsf{Gen}(1^\lambda);\\ \pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w)\end{array} : \mathsf{V}(\mathsf{vrs}, x, \pi) = 1\right] \geq 1 - \mathsf{negl}(\lambda).$$

Knowledge-Soundness. *For any PPT algorithm* A*, there exists a polynomial-time extractor* Ext_A *such that*

$$\Pr\left[\begin{array}{lcr}(\mathsf{crs}, \mathsf{vrs}) \leftarrow \mathsf{Gen}(1^\lambda); & & \mathsf{V}(\mathsf{vrs}, x, \pi)\\ (x, \pi; w) & : & = 1\\ \quad \leftarrow (\mathsf{A} \| \mathsf{Ext}_\mathsf{A})^{\mathsf{V}(\mathsf{vrs}, \cdot, \cdot)}(\mathsf{crs}) & & \wedge (x, w) \notin R\end{array}\right] \leq \mathsf{negl}(\lambda).$$

Zero-Knowledge. *There exists a PPT simulator* $\mathsf{Sim} = (\mathsf{Sim}_0, \mathsf{Sim}_1)$ *such that for any PPT algorithm* A*, it holds that*

$$\Big| \Pr[(\mathsf{crs}, \mathsf{vrs}) \leftarrow \mathsf{Gen}(1^\lambda) : 1 \leftarrow \mathsf{A}^{\mathsf{P}(\mathsf{crs}, \cdot, \cdot)}(\mathsf{crs})] \\ - \Pr\left[\ (\mathsf{crs}, \mathsf{vrs}, \mathsf{td}) \leftarrow \mathsf{Sim}_0(1^\lambda)\ : 1 \leftarrow \mathsf{A}^{\mathsf{Sim}^*(\cdot, \cdot)}(\mathsf{crs})\right]\Big| \leq \mathsf{negl}(\lambda),$$

where the Sim^* *oracle on input* (x, w) *returns* $\perp$ *if* $(x, w) \notin R$*, and returns* $\pi \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \mathsf{td}, x)$ *otherwise, where the PPT algorithm* Sim_1 *takes as input* crs*, a trapdoor* td*, and a statement* x*, and outputs a simulated proof* π*.*

Succinctness. *The length of the proof generated by* P *, as well as the running time of* V*, is bounded by* $\mathsf{poly}(\lambda + |x|)$*, where* x *is a statement, and* poly *is a universal polynomial which does not depend on* R*.*

In addition, a zk-SNARK system is publicly verifiable *if* knowledge-soundness *holds against the adversary given* vrs*, and it is* designated verifier *otherwise.*

To construct the keyed-FHE scheme, we assume the following primitives: an FHE scheme $\Pi_{\mathsf{FHE},i} = (\mathsf{KGen}_{F,i}, \mathsf{Enc}_{F,i}, \mathsf{Dec}_{F,i}, \mathsf{Eval}_{F,i})$ for $i \in \{1, 2\}$,
a publicly-verifiable zk-SNARK system $\Pi_\mathsf{S} = (\mathsf{Gen}_S, \mathsf{P}_S, \mathsf{V}_S)$ for a relation

$$\left\{(\mathsf{ct}_1, \mathsf{ct}_2), (\mathsf{m}, r_1, r_2) \;\middle|\; \begin{array}{l}\mathsf{ct}_1 = \mathsf{Enc}_{F,1}(\mathsf{pk}_1, \mathsf{m}; r_1) \wedge\\ \mathsf{ct}_2 = \mathsf{Enc}_{F,2}(\mathsf{pk}_2, \mathsf{m}; r_2)\end{array}\right\}$$
$$\vee \left\{(\mathsf{ct}_1, \mathsf{ct}_2), (\{\mathsf{ct}_1^{(i)}, \mathsf{ct}_2^{(i)}, \pi_S^{(i)}\}_{i \in [\ell]}, \mathsf{C}, \hat{r}_1, \hat{r}_2) \;\middle|\; \begin{array}{l}\forall j \in \{1, 2\}, \widehat{\mathsf{ct}}_j = \mathsf{Eval}_{F,j}(\mathsf{C}, (\mathsf{ct}_j^{(k)})_{k \in [\ell]}; \hat{r}_j) \wedge\\ \forall i \in [\ell], \mathsf{V}_S(\mathsf{vrs}_S, (\mathsf{ct}_1^{(i)}, \mathsf{ct}_2^{(i)}), \pi_S^{(i)}) = 1\end{array}\right\},$$

a DSS-NIZK system Π_DN in the partial-simulation world $(\mathsf{sfGen}_N, \mathsf{sfSim}_N, \mathsf{pV}_N)$ for a relation $\{(\mathsf{ct}_1, \mathsf{ct}_2), (\mathsf{m}, r_1, r_2) \mid \mathsf{ct}_1 = \mathsf{Enc}_{F,1}(\mathsf{pk}_1, \mathsf{m}; r_1) \wedge \mathsf{ct}_2 = \mathsf{Enc}_{F,2}(\mathsf{pk}_2, \mathsf{m}; r_2)\}$, where $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow$

$\mathsf{KGen}_{F,1}(1^\lambda)$, $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \mathsf{KGen}_{F,2}(1^\lambda)$, and $(\mathsf{crs}_S, \mathsf{vrs}_S) \leftarrow \mathsf{Gen}_S(1^\lambda)$. Notice that regarding the DSS-NIZK system Π_{DN}, we also use the real world prover and verifier algorithms P_N and V_N, in the same way as the keyed-FHE scheme in Section 3.

By using these primitives, we describe the generic construction $\Pi'_{\mathsf{KFHE}} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$, as follows:

- $(\mathsf{pk}, \mathsf{sk}_d, \mathsf{sk}_h) \leftarrow \mathsf{KGen}(1^\lambda)$:
  1. $(\mathsf{pk}_1, \mathsf{sk}_1) \leftarrow \mathsf{KGen}_{F,1}(1^\lambda)$, $(\mathsf{pk}_2, \mathsf{sk}_2) \leftarrow \mathsf{KGen}_{F,2}(1^\lambda)$.
  2. $(\mathsf{crs}_S, \mathsf{vrs}_S) \leftarrow \mathsf{Gen}_S(1^\lambda)$.
  3. $(\mathsf{crs}_N, \mathsf{td}_{N,s}, \mathsf{td}_{N,v}) \leftarrow \mathsf{sfGen}_N(1^\lambda)$.
  4. Output $\mathsf{pk} = (\mathsf{pk}_1, \mathsf{pk}_2, \mathsf{crs}_S, \mathsf{vrs}_S, \mathsf{crs}_N)$, $\mathsf{sk}_d = \mathsf{sk}_1$, and $\mathsf{sk}_h = \mathsf{td}_{N,s}$.

- $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$:
  1. $\mathsf{ct}_1 \leftarrow \mathsf{Enc}_{F,1}(\mathsf{pk}_1, \mathsf{m}; r_1)$, $\mathsf{ct}_2 \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, \mathsf{m}; r_2)$.
  2. $\pi_S \leftarrow \mathsf{P}_S(\mathsf{crs}_S, (\mathsf{ct}_1, \mathsf{ct}_2), (\mathsf{m}, r_1, r_2))$.
  3. $\pi_N \leftarrow \mathsf{P}_N(\mathsf{crs}_N, (\mathsf{ct}_1, \mathsf{ct}_2), (\mathsf{m}, r_1, r_2), \mathsf{lbl})$, where $\mathsf{lbl} = \pi_S$.
  4. Output $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2, \pi_S, \pi_N)$.

- $\mathsf{m}/\bot \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$: $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2, \pi_S, \pi_N)$.
  1. If $\mathsf{V}_S(\mathsf{vrs}_S, (\mathsf{ct}_1, \mathsf{crs}_2), \pi_S) = 1$ and $\mathsf{V}_N(\mathsf{crs}_N, (\mathsf{ct}_1, \mathsf{ct}_2), \pi_N, \pi_S) = 1$, output $\mathsf{m} \leftarrow \mathsf{Dec}_{F,1}(\mathsf{sk}_1, \mathsf{ct}_1)$. Otherwise, output $\bot$.

- $\widehat{\mathsf{ct}}/\bot \leftarrow \mathsf{Eval}(\mathsf{sk}_h, \mathsf{C}, (\mathsf{ct}^{(1)}, \dots, \mathsf{ct}^{(\ell)}))$: Let $\mathsf{ct}^{(i)} = (\mathsf{ct}_1^{(i)}, \mathsf{ct}_2^{(i)}, \pi_S^{(i)}, \pi_N^{(i)})$ for $i \in [\ell]$.
  1. Output $\bot$ if $\mathsf{V}_S(\mathsf{vrs}_S, (\mathsf{ct}_1^{(i)}, \mathsf{ct}_2^{(i)}), \pi_S^{(i)}) = 0$ for some $i \in [\ell]$, or $\mathsf{V}_N(\mathsf{crs}_N, (\mathsf{ct}_1^{(i)}, \mathsf{ct}_2^{(i)}), \pi_N^{(i)}, \pi_S^{(i)}) = 0$ for some $i \in [\ell]$.
  2. $\widehat{\mathsf{ct}}_1^{(0)} \leftarrow \mathsf{Enc}_{F,1}(\mathsf{pk}_1, 0)$, $\widehat{\mathsf{ct}}_2^{(0)} \leftarrow \mathsf{Enc}_{F,2}(\mathsf{pk}_2, 0)$, where 0 is the additive identity in $\mathcal{M}$.
  3. $\widehat{\mathsf{ct}}'_1 \leftarrow \mathsf{Eval}_{F,1}(\mathsf{C}, (\mathsf{ct}_1^{(1)}, \dots, \mathsf{ct}_1^{(\ell)}); \hat{r}_1)$, $\widehat{\mathsf{ct}}'_2 \leftarrow \mathsf{Eval}_{F,2}(\mathsf{C}, (\mathsf{ct}_2^{(1)}, \dots, \mathsf{ct}_2^{(\ell)}); \hat{r}_2)$.
  4. $\widehat{\mathsf{ct}}_1 \leftarrow \mathsf{Add}_{F,1}(\widehat{\mathsf{ct}}'_1, \widehat{\mathsf{ct}}_1^{(0)})$, $\widehat{\mathsf{ct}}_2 \leftarrow \mathsf{Add}_{F,2}(\widehat{\mathsf{ct}}'_2, \widehat{\mathsf{ct}}_2^{(0)})$, where the PPT algorithm $\mathsf{Add}_{F,i}$ ($i \in \{1, 2\}$) evaluates the addition gate over $\mathcal{M}$ by using $\mathsf{Eval}_{F,i}$.
  5. $\widehat{\pi}_S \leftarrow \mathsf{P}_S(\mathsf{crs}_S, (\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2), (\{\mathsf{ct}_1^{(i)}, \mathsf{ct}_2^{(i)}, \pi_S^{(i)}\}_{i\in[\ell]}, \mathsf{C}, \hat{r}_1, \hat{r}_2))$: This is a proof for the witness $(\{\mathsf{ct}_1^{(i)}, \mathsf{ct}_2^{(i)}, \pi_S^{(i)}\}_{i\in[\ell]}, \mathsf{C}, \hat{r}_1, \hat{r}_2)$ such that
     - $\widehat{\mathsf{ct}}_j = \mathsf{Eval}_{F,j}(\mathsf{C}, (\mathsf{ct}_j^{(1)}, \dots, \mathsf{ct}_j^{(\ell)}); \hat{r}_j)$ for every $j \in \{1, 2\}$, and
     - $\mathsf{V}_S(\mathsf{vrs}_S, (\mathsf{ct}_1^{(i)}, \mathsf{ct}_2^{(i)}), \pi_S^{(i)}) = 1$ for every $i \in [\ell]$.
  6. $\widehat{\pi}_N \leftarrow \mathsf{sfSim}_N(\mathsf{crs}_N, \mathsf{td}_{N,s}, (\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2), 1, \widehat{\pi}_S)$.
  7. Output $\widehat{\mathsf{ct}} = (\widehat{\mathsf{ct}}_1, \widehat{\mathsf{ct}}_2, \widehat{\pi}_S, \widehat{\pi}_N)$.

The correctness of Π'_{KFHE} holds in the same way as the keyed-FHE scheme in Section 3. Namely, the first condition of the correctness follows the correctness of $\Pi_{\mathsf{FHE},1}$, and the completeness of Π_{S} and Π_{DN}. The second condition of the correctness also holds due to the composable partial zero-knowledge of Π_{DN} in addition to the correctness and completeness of the underlying primitives. In addition, it is clear that the compactness of Π'_{KFHE} follows the compactness of the two FHE schemes $\Pi_{\mathsf{FHE},1}$ and $\Pi_{\mathsf{FHE},2}$, and the succinctness of Π_{S}.

Theorem 3. *If both* $\Pi_{\mathsf{FHE},1}$ *and* $\Pi_{\mathsf{FHE},2}$ *are* IND-CPA *secure,* Π_{S} *is a zk-SNARK system, and* Π_{DN} *is a strong DSS-NIZK system, then the resulting* Π'_{KFHE} *is* KH-CCA *secure.*

The proof of this theorem is similar to that of Theorem 1. Even though a homomorphic evaluation key is revealed, we have to ensure the confidentiality of the challenge message. To this end, Π'_{KFHE} needs the knowledge-soundness and zero-knowledge of the underlying zk-SNARK, while the keyed-FHE scheme in Section 3 employs the IND-CCA1 security of the underlying FHE schemes.