

Three-Round Weak Non-Malleable Zero-Knowledge Argument

Xinxuan Zhang^{1,2}, Yuanju Wei^{1,2}, Zhichao Wang^{1,2}, Zhongliang Zhang^{1,2},
Ming Yang^{1,2}, Ruida Wang^{1,2}, Yi Deng^{*3}, and Hailong Wang⁴

¹ State Key Laboratory of Cyberspace Security Defense, Institute of Information
Engineering, CAS

² School of Cybersecurity, UCAS
{zhangxinxuan, weiyuanju, zhangzhongliang, yangming2023, wangruida}@iie.ac.cn
wangzhichao222@mailsucas.ac.cn

³ Xidian University
ydeng.cas@gmail.com

⁴ Digital Technologies, Ant Group
whl383799@antgroup.com

Abstract. Non-malleable zero-knowledge argument (NMZK) is a strong notion of zero-knowledge argument that ensures security against man-in-the-middle (MIM) attacks. While three-round constructions exist for various weak zero-knowledge arguments under standard assumptions, all known (weak) NMZK protocols in the plain model have required at least four rounds.

In this work, we construct the *first three-round weak non-malleable zero-knowledge argument* under standard cryptographic assumptions. Our protocol satisfies weak zero-knowledge and ϵ -non-malleability, where the latter allows an ϵ probability gap between the MIM experiment and the stand-alone experiment for any polynomial inverse ϵ . Our construction relies only on well-established primitives, such as the existence of two-message oblivious transfer protocols and delayed-input WI arguments, non-interactive commitments, and circuit-privacy fully homomorphic encryptions.

Keywords: Weak Zero-Knowledge · Non-Malleability.

1 Introduction

Zero-knowledge proofs [21] (ZKPs) are a foundational cryptographic primitive that allows a prover to convince a verifier of the validity of a statement without revealing any information beyond the truth of the statement itself. Round complexity stands as a fundamental research direction in this domain. In the plain model, zero-knowledge protocols are known to require at least three rounds [20]. On the positive side, four-round zero-knowledge protocols for NP with negligible soundness error are known under minimal assumptions such as the existence of

* Corresponding author

one-way functions [2]. However, achieving the optimal three-round bound has proven significantly more challenging.

Current approaches to three-round ZK fall into two broad categories. One line of work relies on *non-standard assumptions*, including auxiliary-input knowledge assumptions [25,3], and the existence of multi-collision-resistant hash functions [6].

The second line of research, which is also the primary focus of this paper, focuses on *weakened notions of zero-knowledge*, such as *weak zero-knowledge* (WZK) [16] and *witness hiding* [17]. These notions relax the simulator’s requirements, for example, WZK only requires that for every verifier and distinguisher, there exists a simulator that fools this specific distinguisher with distinguishing advantage at most ϵ for any inverse-polynomial ϵ (the simulator additionally takes $1^{1/\epsilon}$ as input). A series of works has shown how to construct three-round delayed-input distributional weak zero-knowledge protocols from various standard assumptions, such as the DDH or QR assumptions [26], the factoring assumption [12], the existence of two-round oblivious transfer (OT) protocols [13], or trapdoor permutations [28]. Moreover, Bitansky et al. [7] constructed three-round protocols for weak zero-knowledge and witness hiding under standard assumptions such as the existence of fully homomorphic encryption, LWE, and either Factoring or standard Bilinear-Group assumptions.

This line of research highlights a promising strategy: although achieving standard zero-knowledge in three rounds remains challenging, carefully defined relaxations can still yield meaningful progress under standard assumptions. Our work follows this theoretical tradition, but focuses on the *non-malleable* variants of these notions.

Non-Malleable Zero-Knowledge. Non-malleable zero-knowledge (NMZK), introduced by Dolev, Dwork, and Naor [15], is a strengthening of zero-knowledge that resists man-in-the-middle attacks. In an NMZK protocol, a malicious verifier interacting with an honest prover should not be able to produce a valid proof of any related statement. NMZK plays a crucial role in cryptographic applications that require composability and concurrent security, such as secure multiparty computation and coin-tossing protocols.

Constructing NMZK arguments appears to be substantially harder than constructing standalone zero-knowledge arguments. In particular, *all known NMZK protocols under standard assumptions require at least four rounds of interaction*. The landmark result in this area is due to Goyal et al. [24], who constructed a four-round NMZK argument from one-way functions. Their protocol relies on a specific parallel composition of a zero-knowledge argument of knowledge and a non-malleable commitment (NMC). Subsequently, Ciampi et al. designed a four-round *delayed-input* NMZK [11] which is also a delayed-input many-many synchronous NMZK argument. Recently, Kiyoshima[30] achieved the first four-round *statistical* non-malleable zero-knowledge. Additional NMZK constructions provided by Kim et al. [29] and Botta et al. [8] also require at least four rounds. This situation contrasts sharply with that of standard zero-knowledge, where

certain relaxed notions can already be realized in three rounds under standard assumptions. This gap naturally motivates the following question:

Is it possible to construct three-round NMZK, even in a weakened form, that matches the same round complexity as weak zero-knowledge?

1.1 Our Results

In this work, we resolve this question by constructing the *first three-round weak non-malleable zero-knowledge argument* under standard cryptographic assumptions. Our protocol satisfies weak zero-knowledge and ϵ -non-malleability, where the latter allows an ϵ probability gap between the MIM experiment⁵ and the standalone experiment for any inverse polynomial ϵ . Our construction relies only on well-established primitives, such as the existence of circuit-privacy fully homomorphic encryption, two-round delayed-input witness-indistinguishable arguments, two-round oblivious transfer, and non-interactive commitments. All these primitives can be realized from the polynomial-secure Learning With Errors (LWE) assumption with a corresponding circular security assumption for FHE and trapdoor permutations.

Our main technical contributions, which may be of independent interest, are as follows:

- A new notion called “multi-bit-output distinguisher”, which can strengthen the conception of distinguisher-dependent simulation.
- A weak zero-knowledge argument of knowledge against multi-bit-output distinguisher and non-adaptive B -rewinding malicious verifier who, after receiving the first-round message, can simultaneously generate B distinct second-round messages and observe the prover’s responses to each. This construction also improved Bitansky et al’s non-black-box technique (the version without compute-and-compare obfuscation).
- A simulation-extraction strategy for 3-round weak NMZK.

1.2 Technical Overview

In this work, we directly construct a weak NMZK protocol that satisfies weak simulation-extractability⁶. This property is a relaxation of standard simulation-extractability [34], analogous to how weak zero-knowledge relaxes standard zero-knowledge. Concretely, this notion permits a distinguisher-dependent simulator and allows an ϵ distinguishing gap for any inverse polynomial ϵ ; crucially, it directly implies ϵ -non-malleability⁷.

⁵ In this paper, we require the statement proved by the MIM adversary in the right execution to be static, i.e., announced before the left execution begin.

⁶ The term “weak simulation-extractability” has also been used in the SNARK literature with a different meaning, where the extraction is only required for *different* forged statement [31,1]; our notion is unrelated and should not be confused with it.

⁷ This follows from the proof in [34, Proposition 3.6], observing that the honest verifier fulfills the role of the distinguisher in the “weak” setting.

A Straightforward Attempt. A natural starting point for constructing three-round weak NMZK protocols is the paradigm from [24], which combines a NMC to the witness w with a zero-knowledge argument proving that w is valid and correctly committed. This paradigm enables the construction of four-round NMZK protocols. However, a direct compression to three rounds, by simply replacing them with a three-round NMC and a three-round weak zero-knowledge (WZK) argument, fails. This is because the simulation techniques available for existing three-round WZK protocols[7] differ fundamentally from those used in four-round ZK protocols, making the required extraction in this framework unattainable. The most immediate and foremost obstacles are as follows:

1. In the definition and construction of WZK, the simulator is required to be secure against a *single, fixed* distinguisher, and it may depend on that distinguisher. In contrast, the simulator for simulation-extractibility must satisfy a dual requirement: it must output an indistinguishable view for *one* distinguisher, and it must also additionally output a valid witness when the right-session is acceptable – a condition that can be formalized as satisfying a *second, separate* distinguisher.
2. Unlike the four-round constructions, the simulator of existing three-round WZK[7] needs to extract the trapdoor fixed in the *second-round* messages, while the extractor of underlying NMC (used for the extraction of witness) will rewind the man-in-the-middle (MIM) adversary to the second round, which may render irrelevant the trapdoor extracted previously.

Multi-Bit-Output Distinguisher. To overcome the first obstacle, we introduce a stronger variant of indistinguishability. For any multi-bit-output distinguisher D (*in this work we only consider constant-bit-output distinguishers*) and for any two distributions $\mathcal{X}$ and $\mathcal{Y}$, we say that $\mathcal{X}$ and $\mathcal{Y}$ are indistinguishable by D except with probability ϵ , denoted $\mathcal{X} \approx_{D,\epsilon} \mathcal{Y}$, if the statistical distance between $D(\mathcal{X})$ and $D(\mathcal{Y})$ is less than ϵ .

While it is true that if $\mathcal{X} \approx_{D_s,\text{negl}} \mathcal{Y}$ holds for *all* 1-bit-output distinguisher D_s , then $\mathcal{X} \approx_{D_m,\text{negl}} \mathcal{Y}$ also holds for *all* multi-bit-output distinguisher D_m , this implication does not extend to the case of indistinguishability with respect to a specific distinguisher (or to distinguisher-dependent distributions). For example, given two specific 1-bit-output distinguishers D_1 and D_2 , the facts that $\mathcal{X} \approx_{D_1,\text{negl}} \mathcal{Y}$ and $\mathcal{X} \approx_{D_2,\text{negl}} \mathcal{Y}$ do not imply $\mathcal{X} \approx_{D,\text{negl}} \mathcal{Y}$ for the *combined* 2-bit-output distinguisher $D(x) = D_1(x) \| D_2(x)$ ⁸. This makes the notion meaningful for WZK, where the simulator may depend on (and be tailored to) a specific distinguisher.

WZK against Multi-Bit-Output Distinguishers. We employ a new variant of knowledge encryption(KE)[13] to instantiate Bitansky’s simulation technique. Specifically, knowledge encryption is a bit encryption scheme that allows

⁸ As a concrete instantiation, consider $\mathcal{X} \leftarrow \{00, 11\}, \mathcal{Y} \leftarrow \{01, 10\}$, and define distinguishers $D_1(x_0 \| x_1) = x_0, D_2(x_0 \| x_1) = x_1$. Then, $D_1(\mathcal{X}) \equiv D_1(\mathcal{Y})$ and $D_2(\mathcal{X}) \equiv D_2(\mathcal{Y})$; each outputs 1 with probability 1/2. Yet, since $D_1(\mathcal{X}) \| D_2(\mathcal{X}) \equiv \mathcal{X}$ and $D_1(\mathcal{Y}) \| D_2(\mathcal{Y}) \equiv \mathcal{Y}$, the joint output (D_1, D_2) is trivially distinguishable.

extraction of (partial) secret keys from a bit-ciphertext distinguisher (one that distinguishes ciphertexts encrypting different bits) with 1-bit-output. Our variant extends it to distinguishers with constant-bit output and we show that every standard KE scheme already meets this extended requirement by a simple observation: a “good” k -bit-output distinguisher D implies at least one “good” 1-bit-output distinguisher in the form of $D_\alpha(x) = \text{equal}(\alpha, D(x))$ for $\alpha \in \{0, 1\}^k$. The extractor can thus enumerate all such D_α , attempt extraction from each, and verify whether a valid (partial) secret key is recovered.

We now demonstrate how to construct a WZK protocol⁹ by combining a KE scheme with (modified) Bitansky’s simulation technique. We begin by a basic protocol: The verifier sends a KE public key to the prover. The prover then proves an OR statement that $x \in L$ or it knows the secret key except where the proof is *encrypted* bit-by-bit. The simulation of this protocol is divided into the following two cases:

- **Case 1.** If the distinguisher can distinguish encrypted honest proofs from encrypted dummy message, then the simulator can extract the KE secret key and subsequently use it as trapdoor to conclude the proof of OR statement.
- **Case 2.** If the distinguisher cannot distinguish them, then the simulator can simply generate an encryption of a dummy message to complete the simulation.

The simulator first attempts to extract the KE secret key and decides which case applies based on whether a valid secret key is obtained. However, the extraction of KE secret key is not straightforward. Because the simulator has to construct a bit-ciphertext distinguisher (from the WZK distinguisher, who only outputs after seeing complete proof) to enable the extraction of secret key. This, however, requires the simulator itself to be able to generate an encrypted honest proof (or an indistinguishable one) in order to embed the bit-ciphertext to be distinguished into a specific bit position. This seems to result in a circular dependency.

Bitansky et al.’s non-black-box simulation technique [7] provides a beautiful way to address above issue. Its core idea is to enable the simulator to *homomorphically* generate valid proofs and extract the secret key, and then use the *encrypted* secret key as trapdoor directly. Specifically, the verifier sends an FHE encryption $\text{FHE.Enc}(r)$ of some value r . The prover’s OR statement is then modified to: “ $x \in L$ or it knows a trapdoor TD (which is either r or the *FHE-encrypted* KE secret key¹⁰).” In this setting, the first simulation case (case 1) where the distinguisher can distinguish proofs can be handled as follows: the simulator

⁹ Proving directly that the original WZK construction[7] remains secure against multi-bit-output distinguishers is not straightforward, primarily because the use of randomized self-reducible encryption rather than KE scheme. In the former, one cannot efficiently verify whether an extracted plaintext is correct, thereby hard to prove its extractibility against multi-bit-output distinguishers

¹⁰ To prove knowledge of the FHE-encrypted KE secret key, the prover can first homomorphically evaluate the secret key verification algorithm, producing an output ciphertext. It then proves the correctness of this computation and includes the re-

uses $\text{FHE.Enc}(r)$ to homomorphically generate valid proofs and then homomorphically invokes the extractor of the KE scheme to obtain an FHE-encrypted KE secret key. Once this FHE-encrypted secret key is obtained, the simulator can treat it as a trapdoor and successfully complete the simulation.

Another issue arises as the simulator now cannot verify the validity of the extracted trapdoor (to determine which cases applies). Moreover, due to the use of KE, one cannot simply adopt the compute-and-compare obfuscator as done by Bitansky et al. : KE would cause issues in the soundness proof when trying to replace the obfuscator. Bitansky et al. observe that the WZK distinguisher could be further used to check whether the extracted trapdoor is valid in this setting, but doing so considerably increases the complexity of both the construction and the security proof.

We improve Bitansky et al.'s simulation technique by the following modifications: First modify the KE extractor so that if extraction fails, it always outputs $\perp$. Then, instead of considering distinguishing encrypted honest proofs and encrypted dummy messages, we replace the dummy message with *dummy proofs*, which is the invalid proofs generated using $\text{FHE.Enc}(\perp)$ as the trapdoor. This allows the simulator to directly use whichever FHE ciphertext is extracted (whether $\text{FHE.Enc}(sk)$ or $\text{FHE.Enc}(\perp)$) to generate a proof, *without needing to verify the extracted trapdoor*. Concretely, our two cases now become:

- **Case 1.** If the distinguisher can distinguish the encrypted honest proof from the encrypted dummy proof, the simulator will extract $\text{FHE.Enc}(sk)$ and successfully simulate the proof.
- **Case 2.** If the distinguisher cannot distinguish them, the simulator will extract $\text{FHE.Enc}(\perp)$. It then uses $\text{FHE.Enc}(\perp)$ to generate a proof, which is exactly the dummy proof and hence indistinguishable from the distinguisher.

A schematic illustration of our simulation is shown in Fig.1, where the boxes indicate FHE encryption.

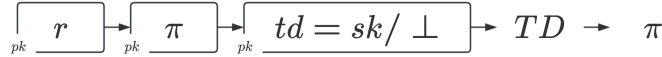


Fig. 1. Schematic illustration of our simulation.

WZK against Non-Adaptive B -Rewinding Verifier. In the construction of NMZK, NMC is used to extract the witness used in the right session. However, when the NMC extractor rewinds the MIM adversary, it also rewinds the left session. This would normally require a zero-knowledge protocol with resettable security, which cannot be achieved in three rounds.

Fortunately, depending on the chosen NMC protocol, the rewinding strategy is already known. When using the NMC scheme from [24], it suffices for the WZK

sulting ciphertext in the proof. The verifier checks both the validity of the proof and that the FHE ciphertext indeed encrypts 1. When such a statement is embedded within an OR statement, the proof proceeds in a similar manner.

protocol to be secure against a non-adaptive B -rewinding verifier: a verifier who, after receiving one first-round message, can simultaneously prepare B distinct second-round messages and obtain the prover’s responses to each (we elaborate on this later).

The main difficulty in constructing such a WZK is that our simulator relies on the distinguisher to extract trapdoor homomorphically. However, the distinguisher only outputs after seeing the *complete* view of adversary. Since the adversary may generate different FHE public keys pk_i and send $\text{FHE.Enc}_{pk_i}(r_i)$ in different sessions, it disrupts the procedure of using $\{r_i\}$ to generate *complete* valid proofs homomorphically (a ciphertext $\text{FHE.Enc}_{pk_i}(r_i)$ cannot be used to help produce a complete proof encrypted under a different key pk_j).

Our solution adopts a recursive simulation strategy. Consider an adversary V^* that performs non-adaptive B rewinds. The simulator Sim_0 , which simulates all B sessions, proceeds as follows. First, it uses $\text{FHE.Enc}(r_1)$ to *homomorphically* generate an encrypted valid proof π_1 for the first session. Next, to produce valid proofs $\pi_2, \dots, \pi_B$ for the remaining sessions—still encrypted under pk_1 —it **homomorphically** invokes the simulator Sim_1 dedicated to simulating the remaining $B - 1$ sessions.

This process continues recursively: the homomorphically evaluated Sim_1 will, in turn, homomorphically invoke Sim_2 for the last $B - 2$ sessions. As a result, Sim_2 operates *under a nested encryption that involves both pk_1 and pk_2* .

Once Sim_0 successfully generates complete proofs for all B sessions—all still encrypted under pk_1 —it employs the distinguisher to homomorphically extract the KE secret key in the first session. This extracted key then allows π_1 to be generated **in plaintext**. Finally, Sim_0 executes Sim_1 **in plaintext** on the remaining $B - 1$ sessions to complete the simulation.

Furthermore, we replace the underlying witness-indistinguishable (WI) argument with a rewinding-secure one [23]. This ensures that the simulated B -rewinding transcript remains computationally indistinguishable. Because the simulator’s recursive structure leads to an exponential blow-up in running time, our WZK scheme can only guarantee security for a constant number of rewinds. Fortunately, this constraint is already sufficient for our final construction.

A schematic illustration of our simulation against a non-adaptive 2-rewinding malicious verifier is shown in Fig.2, where the boxes indicate FHE encryption.

Simulation-Extraction Strategy for Weak NMZK. Our three-round weak NMZK construction is straightforward: The prover uses a three-round NMC to commit the witness and then uses a three-round WZK prove that the NMC transcript commits to a valid witness. We instantiate the scheme with Goyal et al.’s NMC[24] (specifically, their basic construction without ZKAoK) and our WZK protocol. Next, we show how it satisfies the weak simulation-extractibility.

First, we recall how Goyal et al.’s NMC extracts the message. Informally, on input a real transcript, named main execution, containing left and right sessions, their extractor operates by: first, rewind and send a constant number (specifically, two) of random second-round challenges simultaneously to the MIM adversary in the right session, while simulating the left session for the adversary. Then

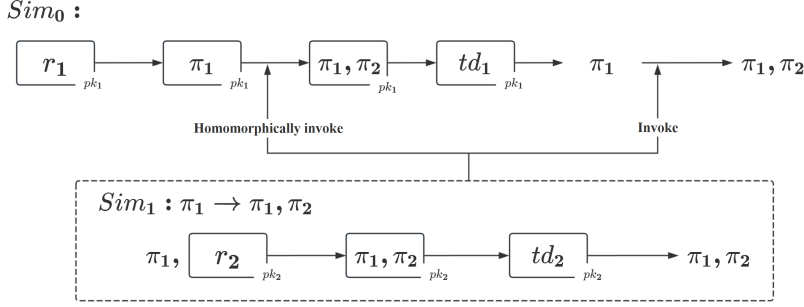


Fig. 2. Schematic illustration of simulation against non-adaptive 2-rewinding adversary

try to extract the message from these transcripts. If extraction fails, repeat the above two steps. Two crucial properties we rely on are: 1) the extractor rewinds the MIM adversary only twice per attempt (we call the extractor invoked in each such attempt a *per-attempt extractor*); and 2) the second-round challenges sent by the extractor consist solely of random coins, which allows the rewinding executions and the main execution to be carried out simultaneously (we call such a rewinding *non-adaptive*). We refer to Fig.7 for more details.

Next, we describe how to construct the simulation-extractor for our construction. As in prior works, we only need to consider two types of adversaries: the non-synchronizing MIM adversary and the synchronizing MIM adversary.

For a *non-synchronizing MIM adversary*, which starts the right session only after the left session has ended, we can use the argument of knowledge extractor of our WZK protocol to extract the witness used in the right session. In this setting, rewinding performed by the extractor in the right session does not affect the ended left session.

More precisely, we construct a malicious WZK verifier V^* that interacts with the external prover P . This verifier runs the MIM adversary as follows: it first completes the left session with the help of the external prover (the non-malleable commitment in this session simply commits to 0, which is indistinguishable from committing to a valid witness), then completes the right session acting an honest verifier and invokes the extractor of our WZK for the right session. Finally, V^* outputs the joint view together with the extracted witness. For any NMZK distinguisher D , we build a two-bit-output distinguisher D^* whose first output bit is the output of D , and second output bit indicates whether the extracted witness is invalid when the right session is acceptable. The WZK simulator with respect to V^* and D^* then directly serves as the simulation-extractor for our weak NMZK protocol.

For a synchronizing MIM adversary, one that advances to the next round only after both the left and right sessions have completed the current one, we employ the NMC per-attempt extractor to obtain the committed witness from the right session. The soundness of the WZK protocol guarantees the validity of the committed witness. The core difficulty is that using the NMC extractor requires the

WZK protocol on the left to be simulated correctly. As noted earlier, since the NMC per-attempt extractor performs only a constant number of non-adaptive rewinds, and thereby the MIM adversary in the left session also rewinds at most a constant number of times non-adaptively, their combined behavior effectively defines a non-adaptive B -rewinding WZK verifier V^* .

More precisely, V^* runs the MIM adversary $\mathcal{A}$ and NMC per-attempt extractor Ext as illustrated in Fig.3, where S_{NMC} is an NMC committer that commits to 0 (indistinguishable from committing to a valid witness), R_{NMC} is an NMC receiver, and V_{WZK} is an honest WZK verifier. V^* finally outputs the joint view in the main execution together with the witness extracted by the NMC per-attempt extractor. (A more detailed description is presented in Fig.8)

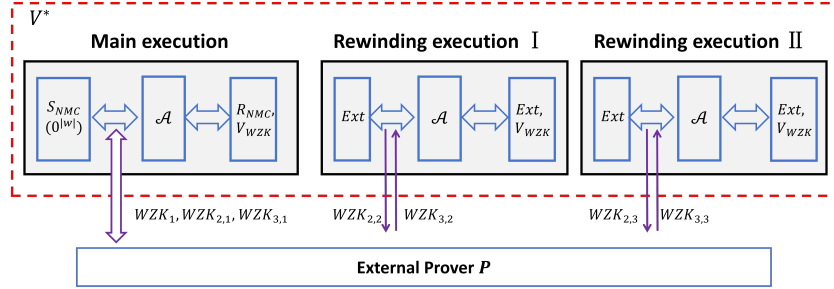


Fig. 3. Execution flow of V^*

For any NMZK distinguisher D , we build a three-bit-output distinguisher D^* whose first output bit is the output of D , and whose second and third bits indicate, respectively, whether the extracted witness is invalid and whether the proof in right session is acceptable. Then, the WZK simulator w.r.t. V^* and D^* can generate a view indistinguishable to D . However, since the NMC single-attempt extractor, given a valid main transcript, succeeds with only polynomial-inverse probability, the WZK simulator likewise extracts a valid witness with only polynomial-inverse probability. Thus, after the first run of the WZK simulator generating an indistinguishable joint view, if the right session is acceptable, we repeat the WZK simulation sufficiently many times to extract a valid witness. This repeated simulation is performed solely to extract the witness for the *static* statement in the right session.

2 Preliminaries

We use $[k]$ to denote the set $\{0, \dots, k-1\}$, the abbreviation PPT to denote probabilistic polynomial time, λ to denote a security parameter, and $\text{negl}(\lambda)$ to denote a negligible function. We use $\approx_c$ to represent computationally indistinguishable, i.e., for every PPT distinguisher, it cannot distinguish the two distributions except with negligible probability. We also use $\approx_{D,\epsilon}$ to represent that the two dis-

tributions are indistinguishable for the distinguisher D except with probability at most ϵ . Due to space limits, several preliminaries are deferred to Appendix.A.

2.1 Non-Malleable Commitments

At a high level, non-malleable commitments are immune to any man-in-the-middle attack. In such an attack, the MIM adversary intercepts an interactive commitment protocol between the committer $\mathcal{C}$ and the receiver $\mathcal{R}$ for message m , and modifies the messages of the protocol in order to efficiently generate a commitment to message m' which is related to m . In other words, the MIM adversary participates in two interactions, where in the left interaction, it interacts with an honest committer $\mathcal{C}$, and in the right interaction, it interacts with an honest receiver $\mathcal{R}$.

Note that a man-in-the-middle attack that does nothing but copy each message from the left to the right can never be prevented. Therefore, existing works [15,34,23,24,22,27] require a public identity for each party, and the MIM adversary has to use an identity different from the identity of $\mathcal{C}$.

Definition 1 (Non-Malleable Commitment [24]). *A commitment $\langle \mathcal{C}, \mathcal{R} \rangle$ is non-malleable if for every PPT MIM adversary $\mathcal{M}$, there exists a simulator Sim , such that for all $id, m \in \{0, 1\}^\lambda$, and $z \in \{0, 1\}^*$,*

$$\{\text{MIM}_{\langle \mathcal{C}, \mathcal{R} \rangle}^{\mathcal{M}}(id, m, z)\} \stackrel{c}{\approx} \{\text{SIM}_{\langle \mathcal{C}, \mathcal{R} \rangle}^{\text{Sim}}(id, z)\}.$$

where,

- $\text{MIM}_{\langle \mathcal{C}, \mathcal{R} \rangle}^{\mathcal{M}}$ is a random variable in the real world under the man-in-the-middle attack, which on input the identity id of $\mathcal{C}$, the committed message m of the left interaction, and an auxiliary information z , outputs $(\tilde{id}, \tilde{m}, v)$. The output $\tilde{id}$ is the identity chosen by $\mathcal{M}$ that should be different from id (if $\tilde{id} = id$, $\tilde{m}$ is set to be $\perp$), $\tilde{m}$ is the message committed in the right interaction, v is the view of $\mathcal{M}$ in both the left and right interaction.
- $\text{SIM}_{\langle \mathcal{C}, \mathcal{R} \rangle}^{\text{Sim}}(id, z)$ is a random variable in the simulated world where $\text{Sim}(id, z)$ directly interacts with an honest receiver. It outputs $(\tilde{id}, \tilde{m}, v)$, which are the chosen identity, the committed message (if $\tilde{id} = id$, $\tilde{m}$ is set to be $\perp$) and the output view of the simulator Sim .

In existing NMC constructions, the above indistinguishability is typically proved by constructing an extractor that, given a MIM transcript, can efficiently extract the committed message in the right session. In this paper, we use a weak variant of NMC. Here “weak” means that when the commitment in the right session is invalid, the extracted value can be different from $\perp$. We rewrite the definition of weak NMC w.r.t. synchronizing adversaries in [27] as follows:

Definition 2 (Weak Non-Malleable Commitments w.r.t. Synchronizing Adversaries). *A statistically binding commitment scheme $\langle \mathcal{C}, \mathcal{R} \rangle$ is said to be weak non-malleable w.r.t. synchronizing adversaries, if there exists a probabilistic “over”-extractor Ext parameterized by ϵ , that given a PPT synchronizing*

MIM (generating valid commitment in the right session with probability larger than ϵ), and given the transcript of an interaction τ , outputs value v in time $\text{poly}(\lambda, \epsilon^{-1})$ such that:

- If the right-session commitment in τ commits to a valid message m , then $v = m$ over the randomness of the extractor and the transcript, except with probability at most ϵ .
- If the right-session commitment in τ commits to some invalid message, then v is not necessarily $\perp$.

As discussed in [27], the three-round sub-protocol within the NMC scheme proposed in [24], specifically, their basic construction using non-interactive statistically binding commitments without the ZKAoK, constitutes a weak non-malleable commitment w.r.t. synchronizing adversaries. Furthermore, it relies solely on non-interactive statistically binding commitments.

More importantly, their extractor construction [24, Fig.3] satisfies a **specific property**: given an input, the extractor repeatedly invokes a “basic” extractor for $\text{poly}(\lambda, \epsilon^{-1})$ times, and this basic extractor only rewinds the MIM adversary a constant number of times in a *non-adaptive* manner. Here, “non-adaptively” means that the second-round challenges sent by the extractor in the right execution in any rewinding session are independent from other sessions. This property is crucial for our construction and a detailed discussion is deferred to Section 5.

2.2 Weak Non-Malleable Zero-Knowledge

Weak Zero-Knowledge. In a zero-knowledge argument for language $\mathcal{L}$, the prover $\mathcal{P}$ with input (x, w) tries to convince the verifier $\mathcal{V}$ with input x that it knows a witness w such that $(x, w) \in R_{\mathcal{L}}$. And during this process, any malicious verifier cannot gain any other knowledge. We provide weak zero-knowledge property[7] as follows:

Definition 3 (Weak Zero-Knowledge[7]). A protocol $(\mathcal{P}, \mathcal{V})$ is Weak Zero-Knowledge if there exists a PPT simulator Sim , such that for any polynomial-size verifier $\mathcal{V}^* = \{\mathcal{V}_{\lambda}^*\}_{\lambda}$, distinguisher $D = \{D_{\lambda}\}_{\lambda}$, inverse polynomial $\epsilon(\lambda)$, it holds that for any $\lambda \in \mathbb{N}$, $x \in \mathcal{L} \cap \{0, 1\}^{\lambda}$, and w such that $(x, w) \in R_{\mathcal{L}}$,

$$\text{Out}_{\mathcal{V}_{\lambda}^*}(\mathcal{P}(w), \mathcal{V}_{\lambda}^*)(x) \approx_{D_{\lambda}, \epsilon} \text{Sim}(x, \mathcal{V}_{\lambda}^*, D_{\lambda}, 1^{1/\epsilon}).$$

We also use the term “delayed-input” in our constructions. A delayed-input interactive protocol between $\mathcal{P}$ and $\mathcal{V}$ for relation $R_{\mathcal{L}}$ proceeds by having both $\mathcal{P}$ and $\mathcal{V}$ receive the size of the instance from the beginning, but sending the prover $(x, w) \in R_{\mathcal{L}}$ and the receiver x just before $\mathcal{P}$ generates its last round message. For a delayed-input protocol, *adaptive weak zero-knowledge* further allows $\mathcal{V}^*$ to select x adaptively.

Non-Malleability. The man-in-the-middle attack to zero-knowledge proofs considers a MIM adversary $\mathcal{M}$ that participates in two interactions. In the left interaction $\mathcal{M}(id, x, z)$ verifies the validity of the statement x by interacting with

$P(x, w)$ using identity id . In the right session interaction $\mathcal{M}$ proves the validity of the statement $\tilde{x}^{11}$ to the honest verifier V , using identity $\tilde{id} \neq id$ of $\mathcal{M}$'s choice. Let $\text{mim}_V^{\mathcal{M}}(id, \tilde{id}, x, \tilde{x}, w, z)$ be a random variable describing the output of V in the above experiment when the random tapes of P , $\mathcal{M}$ and V are uniformly and independently chosen.

In a stand-alone execution only one interaction takes place. The stand-alone adversary $S(id, \tilde{id}, x, \tilde{x}, z)$, using identity $\tilde{id} \neq id$, directly interacts with the honest verifier $V(\tilde{x})$. Let $\text{sta}_V^S(id, \tilde{id}, x, \tilde{x}, z)$ be a random variable describing the output of V in the above experiment when the random tapes of S and V are uniformly and independently chosen.

Definition 4 (Weak Non-Malleable Zero-Knowledge Argument [34]). *A weak non-malleable zero-knowledge argument $\langle P, V \rangle$ is a weak zero-knowledge argument satisfying ϵ -non-malleability defined as follows: for any MIM adversary $\mathcal{M}$ and any inverse polynomial ϵ , there exists a PPT stand-alone prover S , such that for every $(x, w) \in R_{\mathcal{L}}$, every $\tilde{x} \in \{0, 1\}^{|\tilde{x}|}$, every $id \neq \tilde{id}$, and every $z \in \{0, 1\}^*$,*

$$\left| \Pr \left[\text{mim}_V^{\mathcal{M}}(id, \tilde{id}, x, \tilde{x}, w, z) = 1 \right] - \Pr \left[\text{sta}_V^S(id, \tilde{id}, x, \tilde{x}, z) = 1 \right] \right| < \epsilon(\lambda),$$

Non-malleability can often be established using simulation-extractability, which not only requires the existence of a simulation-extractor simulating both the left and the right interactions, but also requires the simulation-extractor to output a valid witness of the right-session statement when the verifier accepts in the right interaction. Therefore, simulation-extractability is a stronger property that implies non-malleability[34]. Here, we give the notion of weak simulation-extractability, of which simulator can depend on the distinguisher.

Definition 5 (Weak Simulation-Extractibility). *A weak zero-knowledge argument $\langle P, V \rangle$ is weak simulation-extractable if for any PPT MIM adversary $\mathcal{M}$, there exists a PPT simulation-extractor $\mathcal{SE}$ such that for any polynomial-sized distinguisher D and any inverse polynomial ϵ , $\mathcal{SE}$ takes input $(D, 1^{1/\epsilon}, id, \tilde{id}, x, \tilde{x}, z)$, outputs $(v, \tilde{w})$, such that,*

1. *For every $(x, w) \in R_{\mathcal{L}}$, every $\tilde{x} \in \{0, 1\}^{|\tilde{x}|}$, every $id \neq \tilde{id}$, and every $\mathcal{M}$'s auxiliary input $z \in \{0, 1\}^*$,*

$$\{\text{MIM}_{\langle P, V \rangle}^{\mathcal{M}}(id, \tilde{id}, x, \tilde{x}, w, z)\} \approx_{D, \epsilon} \{\text{SIM}_V^{\mathcal{SE}}(D, 1^{1/\epsilon}, id, \tilde{id}, x, \tilde{x}, z)\}$$

where,

- *The random variable $\text{MIM}_{\langle P, V \rangle}^{\mathcal{M}}(id, \tilde{id}, x, \tilde{x}, w, z)$ denotes the joint view v of $\mathcal{M}(id, \tilde{id}, x, \tilde{x}, z)$ and the honest verifier $V(\tilde{x})$, when $\mathcal{M}(id, \tilde{id}, x, \tilde{x}, z)$ is verifying the validity of the statement x by interacting with $P(x, w)$ using identity id , and proving on the right the statement $\tilde{x}$ using identity $\tilde{id}$.*

¹¹ In this paper, we require the statement $\tilde{x}$ proved by the MIM adversary in the right execution to be statically by announcing it before the left execution begin.

- The random variable $\mathbf{SIM}_{\mathcal{V}}^{\mathcal{SE}}(D, 1^{1/\epsilon}, id, \tilde{id}, x, \tilde{x}, z)$ denotes the first output of $\mathcal{SE}(D, 1^{1/\epsilon}, id, \tilde{id}, x, \tilde{x}, z)$.
- 2. For every $(x, w) \in R_{\mathcal{L}}$, $\tilde{x} \in \{0, 1\}^{|x|}$, $id \neq \tilde{id}$, and every $\mathcal{M}$'s auxiliary input $z \in \{0, 1\}^*$, let $(v, \tilde{w})$ be the outputs of $\mathcal{SE}$. Then the probability that, the right interaction in v is acceptable while $(\tilde{x}, \tilde{w}) \notin R_{\mathcal{L}}$, is less than ϵ . The probability is over the randomness of $\mathcal{SE}$.

Similar to how simulation-extractibility implies non-malleability, it is straightforward to observe that the above notion of weak simulation-extractibility also implies ϵ -non-malleability. This follows directly from the proof of Proposition 3.6 in [34], observing that the honest verifier fulfills the role of the distinguisher in the “weak” setting.

2.3 B-Rewinding Security

We give the definition of B -rewinding security for witness indistinguishability and zero-knowledge as follows, where B is any positive integer. In this paper, we only consider “non-adaptive” rewinding V^* . Here “non-adaptive” means that the B -rewinding adversary V^* , on input the first round message, needs to generate B two-round messages at the same time.

Definition 6 (3-Round Delayed-Input WI with B -Rewinding Security [23]). A delayed-input 3-round interactive argument for NP relation $R_{\mathcal{L}}$ is WI with B -Rewinding Security if for any non-uniform PPT $\mathcal{V}^*$, it holds that

$$\{\text{Out}_{\mathcal{V}^*}(\mathcal{P}_1, \mathcal{V}_\lambda^*)(1^\lambda)\}_\lambda \stackrel{c}{\approx} \{\text{Out}_{\mathcal{V}^*}(\mathcal{P}_2, \mathcal{V}_\lambda^*)(1^\lambda)\}_\lambda,$$

where $\langle \mathcal{P}_b, \mathcal{V}_\lambda^* \rangle$ for $b \in \{0, 1\}$ is defined as follows:

$\langle \mathcal{P}_b, \mathcal{V}_\lambda^* \rangle(1^\lambda) :$

1. Run $\mathcal{P}$ of the original interactive argument, which generates its first message WI_1 , and a secret state σ . Send WI_1 to $\mathcal{V}_\lambda^*$.
2. $\mathcal{V}_\lambda^*(\text{WI}_1)$ generates $\{(x^i, w^i)\}_{i \in [B-1]}$, x^B, w_0^B, w_1^B and messages $\{\text{WI}_2^i\}_{i \in [B]}$.
3. For $i \in [B-1]$, run $\mathcal{P}(\sigma, \text{WI}_2^i, x^i, w^i)$. For $i = B$, run $\mathcal{P}(\sigma, \text{WI}_2^B, x^B, w_b^B)$. Send $\{\text{WI}_3^i\}_{i \in [B]}$ to $\mathcal{V}_\lambda^*$.

Definition 7 (3-Round Weak ZK with B -Rewinding Security). A 3-round interactive argument for NP relation $R_{\mathcal{L}}$ is WZK against non-adaptive B -rewinding malicious verifier if there exists a simulator Sim , such that for any polynomial-size $\mathcal{V}^* = \{\mathcal{V}_\lambda^*\}_\lambda$, distinguisher $D = \{D_\lambda\}_\lambda$, and inverse-polynomial $\epsilon(\lambda)$, any $\lambda \in \mathbb{N}$, $x \in \mathcal{L} \cap \{0, 1\}^\lambda$, and w such that $(x, w) \in R_{\mathcal{L}}$, it holds that

$$\{\text{Out}_{\mathcal{V}_\lambda^*}(\mathcal{P}, \mathcal{V}_\lambda^*)(1^\lambda, x, w)\}_\lambda \approx_{D_\lambda, \epsilon} \{\text{Sim}(x, \mathcal{V}_\lambda^*, D_\lambda, 1^{1/\epsilon})\}_\lambda,$$

where $\langle \mathcal{P}, \mathcal{V}_\lambda^* \rangle$ is defined as follows:

$\langle \mathcal{P}, \mathcal{V}_\lambda^* \rangle(1^\lambda, x, w) :$

1. Run $\mathcal{P}(x, w; r)$ of the original interactive argument, which generates its first message WI_1 , and a secret state σ . Send WI_1 to $\mathcal{V}_\lambda^*$.
2. $\mathcal{V}_\lambda^*(\text{WI}_1)$ generates B second messages $\{\text{WI}_2^i\}_{i \in [B]}$.
3. For $i \in [B]$, run $\mathcal{P}(\sigma, \text{WI}_2^i, x, w)$. Send $\{\text{WI}_3^i\}_{i \in [B]}$ to $\mathcal{V}_\lambda^*$.

2.4 Knowledge Encryption

Definition 8 (Knowledge Encryption[13]). A knowledge encryption scheme with respect to an NP relation R_L is a triple of PPT algorithms (KE.Gen, KE.Enc, KE.Dec):

- **KE.Gen**($1^\lambda, x, w$) : On input the security parameter $\lambda \in \mathbb{N}$ and statement $x \in L \cap \{0, 1\}^\lambda, w \in R_L(x)$, Gen outputs a key pair (pk, sk) , where the public key is of the form $\text{pk} = (\text{k}, x)$.
- **KE.Enc**(pk, m) : On input the public key pk and a message $m \in \{0, 1\}$, KE.Enc outputs a ciphertext c .
- **KE.Dec**(sk, c) : On input the secret key sk and ciphertext c , KE.Dec outputs a message m (if c is undecryptable, we set m to be “ $\perp$ ”).

We require the following properties from above scheme:

- **Completeness:** For any $\lambda \in \mathbb{N}$, $m \in \{0, 1\}$ and $x \in L \cap \{0, 1\}^\lambda, w \in R_L(x)$:

$$\Pr \left[\text{KE.Dec}(\text{sk}, c) = m \mid \begin{array}{l} (\text{pk}, \text{sk}) \leftarrow \text{KE.Gen}(1^\lambda, x, w) \\ c \leftarrow \text{KE.Enc}(\text{pk}, m) \end{array} \right] = 1.$$

- **Indistinguishability:** For any polynomial-size distinguisher $D = \{D_\lambda\}_{\lambda \in \mathbb{N}}$, there exists a negligible function negl such that for any security parameter $\lambda \in \mathbb{N}$ and $x \in L \cap \{0, 1\}^\lambda, w \in R_L(x)$:

$$\Pr \left[D_\lambda(\text{pk}, c) = m \mid \begin{array}{l} (\text{pk}, \text{sk}) \leftarrow \text{KE.Gen}(1^\lambda, x, w) \\ m \leftarrow \{0, 1\}; c \leftarrow \text{KE.Enc}(\text{pk}, m) \end{array} \right] < \frac{1}{2} + \text{negl}(\lambda).$$

- **Witness Extractability:** There exists a PPT extractor E satisfying that, for any public key $\text{pk}^* = (\text{k}^*, x)$, polynomial-size distinguisher $D = \{D_\lambda\}_{\lambda \in \mathbb{N}}$ and inverse polynomial ϵ , if

$$|\Pr[D_\lambda(\text{KE.Enc}(\text{pk}^*, 0)) = 1] - \Pr[D_\lambda(\text{KE.Enc}(\text{pk}^*, 1)) = 1]| \geq \epsilon,$$

then

$$\Pr[E^{D_\lambda}(\text{pk}^*, 1^{1/\epsilon}) = w \wedge (x, w) \in R_L] \geq 1 - \text{negl}(\lambda),$$

where E runs in time polynomial in ϵ^{-1} and λ .

- **Public Key Simulation:** There exists a PPT simulator KE.KeySim such that for any (x, w) where $x \in L \cap \{0, 1\}^\lambda, w \in R_L(x)$:

$$\{\text{KE.Gen}(1^\lambda, x, w)\} \stackrel{c}{\approx} \{\text{KE.KeySim}(1^\lambda, x)\}.$$

Knowledge encryption schemes can be constructed from any two-round OT protocols [13]. In addition to this, a knowledge encryption with two rounds of key generation can be obtained from trapdoor one-way permutation. We give the construction in Appendix B.

3 Multi-bit-output Distinguisher

To complete the construction of non-malleable zero-knowledge arguments, we need to consider multi-bit-output distinguishers¹². A natural approach is to adopt an alternative definition of indistinguishability. That is, we call $\mathcal{X}$ and $\mathcal{Y}$ are indistinguishable for distinguisher D except ϵ probability (or shortly, $\mathcal{X} \approx_{D,\epsilon} \mathcal{Y}$) if the statistic distance of $D(\mathcal{X})$ and $D(\mathcal{Y})$ is less than ϵ . In this way, we can naturally extend D to multi-bit-output one.

Definition 9. For any $D : \{0, 1\}^* \rightarrow \{0, 1\}^c$, we call two distributions $\mathcal{X}$ and $\mathcal{Y}$ are indistinguishable for distinguisher D except ϵ probability (or shortly, $\mathcal{X} \approx_{D,\epsilon} \mathcal{Y}$) if:

$$\text{Dist}(D(\mathcal{X}), D(\mathcal{Y})) \leq \epsilon.$$

where Dist means the statistic distance.

Fact 1 If $\mathcal{X}$ and $\mathcal{Y}$ are indistinguishable for all distinguishers with 1-bit output except negligible probability, then $\mathcal{X}$ and $\mathcal{Y}$ are also indistinguishable for all distinguishers with constant-bit output except negligible probability.

While we have above fact, this implication does not extend to the case of indistinguishability with respect to a specific distinguisher (or to distinguisher-dependent distributions). For example, given two specific 1-bit-output distinguishers D_1 and D_2 , the facts that $\mathcal{X} \approx_{D_1, \text{negl}} \mathcal{Y}$ and $\mathcal{X} \approx_{D_2, \text{negl}} \mathcal{Y}$ do not imply $\mathcal{X} \approx_{D, \text{negl}} \mathcal{Y}$ for the combined 2-bit-output distinguisher $D(x) = D_1(x) \| D_2(x)$. This makes this notion meaningful for WZK of which simulator relies on and works well for specific distinguisher.

Here we similarly define WZK against multi-bit-output distinguisher:

Definition 10 (Weak Zero-Knowledge Against Multi-Bit-Output Distinguisher). A protocol $\langle \mathcal{P}, \mathcal{V} \rangle$ is weak zero-knowledge against multi-bit-output distinguisher if there exists a PPT simulator Sim , such that for any polynomial-size verifier $\mathcal{V}^* = \{\mathcal{V}_\lambda^*\}_\lambda$, multi-bit-output distinguisher $D = \{D_\lambda\}_\lambda$, and inverse polynomial $\epsilon(\lambda)$, it holds that for any $\lambda \in \mathbb{N}$, $x \in \mathcal{L} \cap \{0, 1\}^\lambda$, and w such that $(x, w) \in R_\mathcal{L}$,

$$\text{Out}_{\mathcal{V}_\lambda^*}(\mathcal{P}(w), \mathcal{V}_\lambda^*)(x) \approx_{D_\lambda, \epsilon} \text{Sim}(x, \mathcal{V}_\lambda^*, D_\lambda, 1^{1/\epsilon}).$$

Furthermore, we establish that the key building block, knowledge encryption, can be extended to achieve security against multi-bit-output distinguishers. The corresponding definition is provided below.

Definition 11 (Knowledge Encryption Against Multi-Bit-Output Distinguisher). The extended definition of knowledge encryption differs from the previous one only in the witness extraction.

¹² Throughout this paper, all multi-bit-output distinguishers are assumed to be constant-bit-output; i.e., their output length is a fixed constant.

- **Witness Extractability:** *There exists a PPT extractor E satisfying that, for any public key $\text{pk}^* = (k^*, x)$, polynomial-size multi-bit-output distinguisher $D = \{D_\lambda\}_{\lambda \in \mathbb{N}}$ and inverse polynomial ϵ , if*

$$\text{Dist}(D_\lambda(\text{KE.Enc}(\text{pk}^*, 0)), D_\lambda(\text{KE.Enc}(\text{pk}^*, 1))) \geq \epsilon$$

then

$$\Pr[E^{D_\lambda}(\text{pk}^*, 1^{1/\epsilon}) = w \wedge (x, w) \in R_L] \geq 1 - \text{negl}(\lambda),$$

where E runs in time polynomial in ϵ^{-1} and λ .

Corollary 1. *A standard knowledge encryption scheme is naturally a knowledge encryption against multi-bit-output distinguisher defined in Definition 11.*

proof sketch. Given a ciphertext-distinguisher D with k -bit output, we define 2^k single-bit-output distinguishers D'_i : each D'_i runs D and outputs 1 iff D 's output equals the binary representation of i . If the statistical distance between $D(\text{KE.Enc}(\text{pk}^*, 0))$ and $D(\text{KE.Enc}(\text{pk}^*, 1))$ is at least $\epsilon(\lambda)$, then there exists some index i^* for which D'_{i^*} has advantage at least $\epsilon(\lambda)/2^k$. Running the origin KE extractor $E^{D'_{i^*}}$ for every i simultaneously (since k is constant) yields a witness for the statement encoded in the public key. $\square$

A detailed proof of this corollary is provided in Appendix C.

The corollary 1 shows that the knowledge encryption schemes secure against multi-bit-output distinguishers can be instantiated exactly as the standard knowledge encryption outlined in the preliminaries.

Remark 1 (An intermediate “multi-distinguisher” notion). We close this section by discussing an intermediate notion, which we call *multi-distinguisher*¹³, that lies between the traditional single-distinguisher and our multi-bit-output distinguisher notion for WZK.

Let $\mathcal{D} = \{D_1, \dots, D_k\}$ be a finite collection of polynomial-size 1-bit-output distinguishers, where each $D_i = \{D_{i,\lambda}\}_{\lambda \in \mathbb{N}}$. WZK against multi-distinguisher requires the simulator Sim , which may now depend on the entire collection $\mathcal{D}$, to fool *every individual distinguisher simultaneously*; that is, for every $i \in [k]$,

$$\text{Out}_{\mathcal{V}_\lambda^*} \langle \mathcal{P}(w), \mathcal{V}_\lambda^* \rangle(x) \approx_{D_{i,\lambda}, \epsilon} \text{Sim}(x, \mathcal{V}_\lambda^*, \mathcal{D}, 1^{1/\epsilon}).$$

On the one hand, this notion is stronger than traditional WZK, where the simulator need only fool a *single* distinguisher and may depend on it. On the other hand, it is weaker than our multi-bit-output definition, since indistinguishability against each individual distinguisher does *not* imply indistinguishability against their joint output, as demonstrated by the example in the technique overview. We remark that the multi-distinguisher notion is highly likely to be sufficient for our WNMZK construction.

¹³ This notion was raised by an anonymous reviewer, to whom we are grateful for suggesting this insightful alternative.

4 B -rewinding Secure WZK Against Multi-bit-output Distinguisher

In this section, we provide a construction of a weak zero-knowledge argument of knowledge against multi-bit-output distinguisher and non-adaptive B -rewinding malicious verifier. Our construction uses the following ingredients:

- A non-interactive commitment COM.
- An IND-CPA public-key encryption scheme $(\text{Gen}, \text{Enc}, \text{Dec})$ ¹⁴.
- A circuit-privacy FHE scheme $(\text{FHE.Gen}, \text{FHE.Enc}, \text{FHE.Eval}, \text{FHE.Dec})$.
- Two two-round delayed-input WI arguments, i.e., $\text{WI}^1 = (\text{WI}_1^1, \text{WI}_2^1)$ for language $\mathcal{L}_1$ and $\text{WI}^2 = (\text{WI}_1^2, \text{WI}_2^2)$ for language $\mathcal{L}_2$.
- A three-round delayed-input WIAoK protocol $\text{WI}' = (\text{WI}'_1, \text{WI}'_2, \text{WI}'_3)$ for language $\mathcal{L}_3$.
- A knowledge encryption scheme $(\text{KE.Gen}, \text{KE.Enc}, \text{KE.Dec})$ for any NP language $\mathcal{L}_{\text{KE}}$ with a hard distribution on it.

where the languages $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ are defined as follows:

$$\begin{aligned} \mathcal{L}_1 : & \left\{ (c_{\text{FHE}}, \text{com}, pk, pk_{\text{FHE}}) \mid \exists r \text{ s.t. } \text{com} = \text{COM}(0; r), pk_{\text{FHE}} \in \text{FHE.Gen}(1^\lambda) \right\} \\ & \left\{ (c_{\text{FHE}} \in \text{FHE.Enc}_{pk_{\text{FHE}}}(r) \vee pk \in \text{Gen}(1^\lambda)) \right\} \\ \mathcal{L}_2 : & \left\{ \left(\begin{array}{l} \text{com}, pk, pk', \\ x, c_{\text{KE}}, pk_{\text{KE}}, \\ ct_0, ct'_0, ct_1, ct_2 \end{array} \right) \mid \begin{array}{l} \exists r, sk' \text{ s.t. } \text{com} = \text{COM}(0; r), ct_0 \in \text{Enc}_{pk}(sk'), r = \text{Dec}_{sk'}(ct'_0) \vee \\ (x, w) \in R_{\mathcal{L}}, ct_1 \in \text{Enc}_{pk}(w) \vee \\ \exists c_{w_{\text{KE}}} \text{ s.t. } c_{\text{KE}} = \text{KE.Enc}_{pk_{\text{KE}}}(\text{FHE.Eval}(g_{x_{\text{KE}}}(\cdot), c_{w_{\text{KE}}})) , ct_2 \in \text{Enc}_{pk}(c_{w_{\text{KE}}}) \end{array} \right\} \\ \mathcal{L}_3 : & \{ (x, pk) \mid x \in \mathcal{L} \vee pk \in \text{Gen}(1^\lambda) \} \end{aligned}$$

where $g_{x_{\text{KE}}}(y) = 1$ if and only if y is a valid witness of x_{KE} , where x_{KE} is the statement in pk_{KE} , and 0 otherwise; and $pk \in \text{Gen}(1^\lambda)$ means that there exist sk and randomness r s.t. $(pk, sk) = \text{Gen}(1^\lambda; r)$. The construction is shown in Fig.4.

Theorem 1. *The protocol constructed in Fig.4 is a (delayed-input) weak zero-knowledge argument of knowledge against multi-bit-output distinguisher.*

Proof. The completeness of the protocol is obvious.

(Delayed-input adaptive) weak zero-knowledge: To simplify the security proof, we use a direct corollary of knowledge encryption, that is, there exists an extractor that can extract the witness from a distinguisher distinguishing $(\text{KE.Enc}(0), \text{KE.Enc}(1))$ from $(\text{KE.Enc}(1), \text{KE.Enc}(0))$ with inverse-polynomial probability.

For any polynomial-size verifier $\mathcal{V}^* = \{\mathcal{V}_\lambda^*\}_\lambda$, any polynomial-size multi-bit-output distinguisher $D = \{D_\lambda\}_\lambda$, and any inverse polynomial $\epsilon(\lambda)$, we construct a simulator as follows:

$\text{Sim}(x, \mathcal{V}^*, D, 1^{1/\epsilon}) :$

¹⁴ We require it to support efficient sampling of an indistinguishable “fake” public key, where “fake” means this key does not belong to the set of honestly generated keys. A scheme meeting this requirement can be easily built from any standard IND-CPA scheme, e.g., set the public key as $pk^* \leftarrow pk \parallel \text{Enc}_{pk}(0)$, then $pk \parallel \text{Enc}_{pk}(1)$ is an indistinguishable “fake” public key.

Three-round weak zero-knowledge argument.

Prover Input: $(x, w) \in R_{\mathcal{L}}$.

Verifier Input: x

- **Prover Message:** Generate the first-round messages WI'_1 and WI_1^1 of the WIAoK WI' and the two-round WI WI^1 , respectively. Generate a key pair $(pk, sk) \leftarrow \text{Gen}(1^\lambda)$. Send (WI'_1, WI_1^1, pk) to the verifier.
- **Verifier Message:** The verifier runs as follows:
 1. Sample (x_{KE}, w_{KE}) from the hard relation over $R_{\mathcal{L}_{KE}}$ and generate a key pair $(pk_{KE}, sk_{KE}) \leftarrow \text{KE.Gen}(x_{KE}, w_{KE})$.
 2. Generate the second-round message WI'_2 of the WIAoK protocol WI' , and the first-round message WI_1^2 of WI^2 .
 3. Commit 0 with fresh randomness r , i.e., $com \leftarrow \text{COM}(0; r)$.
 4. Encrypt r via the FHE scheme with randomly generated pk_{FHE} , i.e., $(pk_{FHE}, sk_{FHE}) \leftarrow \text{FHE.Gen}(1^\lambda)$, $c_{FHE} \leftarrow \text{FHE.Enc}(r)$.
 5. Generate the second-round message WI_2^1 of WI^1 for the statement $(com, c_{FHE}, pk, pk_{FHE}) \in \mathcal{L}_1$, using r and necessary randomness as the witness.

Send $(com, c_{FHE}, pk_{KE}, pk_{FHE}, WI'_2, WI_2^1, WI_1^2)$ to the prover.

- **Prover Message:** The prover first parses $pk_{KE} = (x_{KE}, k_{KE})$ and verifies that (WI_1^1, WI_2^1) is an acceptable proof for the statement $(com, c_{FHE}, pk, pk_{FHE}) \in \mathcal{L}_1$. If the verification fails, it aborts; otherwise, it proceeds as follows:
 1. Generate a key pair $(pk', sk') \leftarrow \text{Gen}(1^\lambda)$. Then, encrypt 1 under the FHE public key pk_{FHE} , i.e., $c'_{FHE} \leftarrow \text{FHE.Enc}_{pk_{FHE}}(1)$. Subsequently, compute the following ciphertexts:

$$ct_0 \leftarrow \text{Enc}_{pk}(0), ct'_0 \leftarrow \text{Enc}_{pk'}(0)$$

$$ct_1 \leftarrow \text{Enc}_{pk}(w), ct_2 \leftarrow \text{Enc}_{pk}(0)$$

Set $ct = (ct_0, ct'_0, ct_1, ct_2)$.

2. Encrypt c'_{FHE} under the KE public key pk_{KE} to obtain $c_{KE} \leftarrow \text{KE.Enc}_{pk_{KE}}(c'_{FHE})$.
3. Generate the second-round message WI_2^2 of WI^2 for the statement $(com, pk, pk', x, c_{KE}, pk_{KE}, ct) \in \mathcal{L}_2$, using w and necessary randomness as the witness.
4. Generate the third-round message WI'_3 of WI' for the statement $(x, pk) \in \mathcal{L}_3$, using w as the witness.

Send $(pk', c_{KE}, ct, WI_2^2, WI'_3)$ to the verifier.

- **Verifier's Output:** Decrypt c_{KE} to obtain c'_{FHE} . Output 1 if the following checks pass, else output 0.
 - (WI_1^2, WI_2^2) is a valid proof for $(com, pk, pk', x, c_{KE}, pk_{KE}, ct) \in \mathcal{L}_2$;
 - $\text{FHE.Dec}(c'_{FHE}) = 1$;
 - (WI'_1, WI'_2, WI'_3) is a valid proof for $(x, pk) \in \mathcal{L}_3$;

Fig. 4. Three-round weak zero-knowledge argument

1. Sample coins r^* for the malicious verifier $\mathcal{V}^*$.
2. Generate the first-round message $(\text{WI}'_1, \text{WI}_1^1, pk)$ honestly, and then send this message to $\mathcal{V}^*$.
3. Receive $(com, c_{\text{FHE}}, pk_{\text{KE}}, pk_{\text{FHE}}, \text{WI}'_2, \text{WI}_2^1, \text{WI}_1^2)$ from $\mathcal{V}^*$ and check whether $(\text{WI}_1^1, \text{WI}_2^1)$ is acceptable. If not, abort and output the view of $\mathcal{V}^*$.
4. Compute the third-round message WI'_3 of WI' for the statement $(x, pk) \in \mathcal{L}_3$, using the randomness generating pk as witness. Write the transcript generated so far as $tran$, i.e., $tran$ contains r^* , the first-round and second-round messages, and WI'_3 .
5. Construct the circuit $\text{HS}(r)$ that will be used in the fully homomorphic evaluation:

- (a) For each $k \in [n]$, construct multi-bit-output distinguisher $D'_k(\hat{c})$ for knowledge encryption, where n is the length of FHE ciphertexts and $\hat{c} = (\hat{c}_0, \hat{c}_1)$ is a pair of KE ciphertexts, each encrypting a single bit:

- i. Generate a key pair $(pk', sk') \leftarrow \text{Gen}(1^\lambda)$ and compute $c'_{\text{FHE}} \leftarrow \text{FHE.Enc}_{pk_{\text{FHE}}}(1)$. Compute the following ciphertexts:

$$\begin{aligned} ct_0 &\leftarrow \text{Enc}_{pk}(sk'), ct'_0 \leftarrow \text{Enc}_{pk'}(r) \\ ct_1 &\leftarrow \text{Enc}_{pk}(0), ct_2 \leftarrow \text{Enc}_{pk}(0) \end{aligned}$$

Set $ct = (ct_0, ct'_0, ct_1, ct_2)$.

- ii. Compute $c^*_{\text{FHE}} \leftarrow \text{FHE.Enc}_{pk_{\text{FHE}}}(0)$. Encrypt c'_{FHE} and c^*_{FHE} **bit by bit** using the knowledge encryption, i.e., $c_{\text{KE}} = \text{KE.Enc}_{pk_{\text{KE}}}(c'_{\text{FHE}})$, $c^*_{\text{KE}} = \text{KE.Enc}_{pk_{\text{KE}}}(c^*_{\text{FHE}})$.
- iii. Replace the first $k-1$ ciphertexts of c_{KE} with the first $k-1$ ciphertexts of c^*_{KE} . If the bits encrypted by the k -th ciphertexts in c_{KE} and c^*_{KE} differ, then replace the k -th ciphertext in c_{KE} with $\hat{c}_{b'}$, where b' denotes the bit encrypted by the original k -th ciphertext of c_{KE} . Otherwise, leave the k -th ciphertext unchanged. Denote the result ciphertexts as c'_{KE} .
- iv. Generate the second-round message WI_2^2 of WI^2 for the statement $(com, pk, pk', x, c'_{\text{KE}}, pk_{\text{KE}}, ct) \in \mathcal{L}_2$, using r (rather than w) and necessary randomness as the witness.
- v. Run and output $D(tran, pk', c'_{\text{KE}}, ct, \text{WI}_2^2)$. (note that $tran$ already contains WI'_3)

- (b) For each k , run the extractor of knowledge encryption $E^{D'_k}(pk_{\text{KE}}, 1^{n/\epsilon})$. If a valid witness is extracted, output it. Otherwise, output $\perp$.

6. Compute $c_{w_{\text{KE}}} = \text{FHE.Eval}(\text{HS}, c_{\text{FHE}})$, $c'_{\text{FHE}} = \text{FHE.Eval}(g_{x_{\text{KE}}}(\cdot), c_{w_{\text{KE}}})$, and a key pair $(pk', sk') \leftarrow \text{Gen}(1^\lambda)$. Then, compute the following ciphertexts:

$$\begin{aligned} ct_0 &\leftarrow \text{Enc}_{pk}(0), ct'_0 \leftarrow \text{Enc}_{pk'}(0) \\ ct_1 &\leftarrow \text{Enc}_{pk}(0), ct_2 \leftarrow \text{Enc}_{pk}(c_{w_{\text{KE}}}) \end{aligned}$$

Set $ct = (ct_0, ct'_0, ct_1, ct_2)$.

7. Compute $c_{\text{KE}} \leftarrow \text{KE.Enc}_{pk_{\text{KE}}}(c'_{\text{FHE}})$.
8. Generate the second-round message WI_2^2 of the WI argument WI^2 for the statement $(com, pk, pk', x, c_{\text{KE}}, pk_{\text{KE}}, ct) \in \mathcal{L}_2$, using $c_{w_{\text{KE}}}$ and necessary randomness as the witness.

9. Set $(pk', c_{KE}, ct, WI_2^2, WI_3')$ as the third message and output the view of $\mathcal{V}^*$.

Now we prove that $\mathcal{V}^*$'s views in the simulation and real interaction are indistinguishable via hybrid simulators. Let $\text{Sim}(x)$ be the original simulator.

$\text{HSIM}_1(x, w)$: it is identical to $\text{Sim}(x)$ except that it uses w as witness to generate WI_3' in step 4. From WI property, we directly have that $\text{Sim}(x) \stackrel{c}{\approx} \text{HSIM}_1(x, w)$.

$\text{HSIM}_2(x, w)$: it is identical to $\text{HSIM}_1(x, w)$ except that it sets $ct_1 \leftarrow \text{Enc}_{pk}(w)$ in step 5(a)i.

Lemma 1. $\text{HSIM}_2(x, w) \stackrel{c}{\approx} \text{HSIM}_1(x, w)$

Proof. Although this lemma relies fundamentally on the IND-CPA security of the underlying encryption scheme, its proof is not straightforward. The difficulty arises because we alter the construction of the ciphertext distinguisher D'_k , which is itself invoked by the KE extractor when the latter is run homomorphically.

The key observation is that, the KE extractor invokes each D'_k in a **black-box manner**. Therefore, during the homomorphic evaluation of the KE extractor, whenever a D'_k is invoked, we can replace the ciphertext ct_1 generated in step 5(a)i with a challenge ciphertext (which encrypts either 0 or w , and needs to be FHE-encrypted to make the replacement feasible). Thereby we can use the distinguisher distinguishing $\text{HSIM}_2(x, w)$ and $\text{HSIM}_1(x, w)$ to break the CPA-security¹⁵. Note that a hybrid argument is necessary here because the KE extractor invokes D'_k multiple times, requiring us to switch the ciphertexts for each invocation one by one. $\square$

$\text{HSIM}_3(x, w)$: it is identical to $\text{HSIM}_2(x, w)$ except that it uses w as witness to generate the WI proof WI_2^2 in step 5(a)iv. By the WI property and applying a proof strategy similar to Lemma 1, we have that $\text{HSIM}_3(x, w) \stackrel{c}{\approx} \text{HSIM}_2(x, w)$.

$\text{HSIM}_4(x, w)$: it is identical to $\text{HSIM}_3(x, w)$ except that it generates $ct_0 \leftarrow \text{Enc}_{pk}(0)$ in step 5(a)i. By the IND-CPA security and applying the proof strategy identical to Lemma 1, we have that $\text{HSIM}_4(x, w) \stackrel{c}{\approx} \text{HSIM}_3(x, w)$.

$\text{HSIM}_5(x, w)$: it is identical to $\text{HSIM}_4(x, w)$ except that it generates $ct'_0 \leftarrow \text{Enc}_{pk'}(0)$ in step 5(a)i.

Lemma 2. $\text{HSIM}_5(x, w) \stackrel{c}{\approx} \text{HSIM}_4(x, w)$

Proof. We complete the proof using a proof strategy similar to that of Lemma 1. The only complication here is that r is homomorphically encrypted—under the identical proof strategy we do not have access to r in the plaintext, which would cause the reduction to IND-CPA security to fail. Fortunately, r is already fixed at

¹⁵ Roughly speaking, the black-box invocation is necessary. In this setting, the execution of each D'_k can be treated as external to the KE extractor. This separation allows us to alter certain internal steps of D'_k in order to embed the challenge ciphertext. If the KE extractor invoked D'_k in a non-black-box manner, we would have no insight into how—or even whether—its steps are executed, making it impossible to embed the challenge ciphertext.

the moment pk' is generated (this is precisely why we encrypt r with the new pk' rather than directly with the original pk), which allow us to exhaustively search for r and supply r as the auxiliary input of the IND-CPA adversary (the search itself is inefficient, but the resulting IND-CPA adversary remains efficient).

To be more detailed, we split the random tape of the simulator into two parts, r_1 and r_2 , where r_1 is used in the first four steps and r_2 in the remaining steps. Assume there exists a distinguisher D that distinguishes $\text{HSIM}_5(x, w)$ from $\text{HSIM}_4(x, w)$ with noticeable probability. Then there must exist some fixed r_1^* such that, when the simulator's first-part randomness is set to r_1^* , D still distinguishes the two hybrids with noticeable probability, and the ciphertext c_{FHE} generated in step 2 correctly encrypts a string denoted by r . (We note that the probability that c_{FHE} is an invalid ciphertext is negligible. Otherwise, $\mathcal{V}^*$ would be able to produce an invalid FHE encryption with noticeable probability while still generating an acceptable WI^1 proof; such a $\mathcal{V}^*$ could then be used directly to break the soundness of the WI protocol by switching the public key pk to an indistinguishable “fake” key.) Given this, we can exhaustively search over all possible values of r_1^* and r .

Now we can essentially copy the proof strategy of Lemma 1. The only modification is that we now supply both r_1^* and r as the auxiliary input of IND-CPA adversary and fix the first part of the simulator's randomness to r_1^* . $\square$

$\text{HSIM}_6(x, w)$: it is identical to $\text{HSIM}_5(x, w)$ except that it sets $ct_1 \leftarrow \text{Enc}(w)$ and uses w as witness to generate the proof in step 6 and step 8. From the WI property and IND-CPA security, we have that $\text{HSIM}_6(x, w) \stackrel{c}{\approx} \text{HSIM}_5(x, w)$.

$\text{HSIM}_7(x, w)$: it is identical to $\text{HSIM}_6(x, w)$ except that it generates $ct_2 \leftarrow \text{Enc}(0)$ in step 6. From IND-CPA security we have that $\text{HSIM}_7(x, w) \stackrel{c}{\approx} \text{HSIM}_6(x, w)$.

$\text{HSIM}_8(x, w)$: it is identical to $\text{HSIM}_7(x, w)$ except that it computes $c'_{\text{FHE}} = \text{FHE.Enc}(1)$ in step 6.

Lemma 3. $\text{HSIM}_7(x, w) \approx_{D, \epsilon} \text{HSIM}_8(x, w)$

Proof. Note that the only difference between them appears that, when c_w is not a valid witness, then c'_{FHE} in $\text{HSIM}_7(x, w)$ is an *evaluated* ciphertext which encrypts 0, while the one in $\text{HSIM}_8(x, w)$ is a *fresh* ciphertext encrypting 1 (here *fresh* means the ciphertext is generated by FHE.Eval). Therefore, if there exists D distinguishing $\text{HSIM}_7(x, w)$ from $\text{HSIM}_8(x, w)$ with noticeable probability, it must happen that the extractor fails to extract witness (i.e., c_w is not a valid witness) with noticeable probability.

Fix a transcript $tran$ generated in step 4, and suppose that, conditioned on $tran$, a distinguisher D can distinguish $\text{HSIM}_7(x, w)$ from $\text{HSIM}_8(x, w)$ with noticeable advantage ϵ . Furthermore, let sk_{FHE} be the secret key corresponding to pk_{FHE} in $tran$ (the underlying WI^1 ensures that such a secret key exists almost always). In what follows, whenever we write HSIM , it is always understood to mean $\text{HSIM} \mid tran$.

Now, we modify $\text{HSIM}_7(x, w)$ to obtain a new hybrid, denoted $\text{HSIM}'_7(x, w)$: when c_w is not a valid witness, we replace c'_{FHE} with a *fresh* ciphertext that

encrypts 0. To do this, the simulator is additionally given sk_{FHE} so that it can check whether c_w is a valid witness. By the circuit privacy of FHE, $\text{HSIM}_7(x, w)$ is computationally indistinguishable from $\text{HSIM}'_7(x, w)$.

Note that c_{FHE} is only used to generate c_{KE} . Now we change the ciphertexts in c_{KE} in the step 7 in HSIM'_7 into the corresponding ciphertexts in HSIM_8 one by one, which introduce $n + 1$ hybrid simulators HSIM'_7^i , where $\text{HSIM}'_7^0 = \text{HSIM}'_7$ and $\text{HSIM}'_7^n = \text{HSIM}_8$. Therefore, the distinguisher can distinguish some two neighboring hybrid simulators, denote by HSIM'_7^k and HSIM'_7^{k+1} , with probability at least ϵ/n on condition that the extractor fails to extract witness.

The key observation involves the input distribution of D during the execution of D'_{k+1} in Step 5(a)v. This distribution is identical to the output distribution of HSIM conditioned on the event that the extractor fails to extract a witness. Specifically:

- If $\hat{c} = (\text{KE.Enc}(0), \text{KE.Enc}(1))$, it matches the distribution of HSIM'_7^k conditioned on extraction failure.
- If $\hat{c} = (\text{KE.Enc}(1), \text{KE.Enc}(0))$, it matches the distribution of HSIM'_7^{k+1} conditioned on extraction failure.

It means that the distinguisher D'_{k+1} can distinguish $(\text{KE.Enc}(0), \text{KE.Enc}(1))$ from $(\text{KE.Enc}(1), \text{KE.Enc}(0))$ with probability at least ϵ/n , which means that the extractor should extract valid witness except negligible probability, meeting a contradiction. $\square$

Note that the output of $\text{HSIM}_8(x, w)$ (step 5 is now useless) is identical to real transcript, the weak zero-knowledge property is concluded. The delayed-input adaptive weak zero-knowledge property can be proved similarly since the simulator doesn't use x until receiving the second-round message.

(Adaptive) argument of knowledge: Due to the space limits, we only provide a proof sketch here and defer the detailed proof to the Appendix.D.

Proof sketch. We construct an extractor $E^{\mathcal{P}^*}$ that runs the WIAoK extractor for $\mathcal{L}_3$ to extract either w or sk . If it returns a witness w for $\mathcal{L}$, E outputs w ; if it returns a secret key sk , E decrypts ct_1 and outputs the result.

Note that when sk is extracted, the soundness of the WI argument for $\mathcal{L}_2$ ensures that (except with negligible probability) at least one of the following holds: (1) com commits to 0 using randomness decrypted from ct_0 ; (2) decrypting ct_1 gives a valid w ; (3) decrypting ct_2 yields a valid witness for a hard statement. Using witness indistinguishability, IND-CPA security of FHE, commitment hiding, and the simulation property of the knowledge encryption scheme, we show that Events (1) and (3) occur with negligible probability. Hence E extracts a valid witness with overwhelming probability. $\square$

In the following, we will provide a modified version of the above protocol that remains secure against a non-adaptive B -rewinding verifier—who, after receiving one first-round message, can simultaneously generate B distinct second-round messages and observe the prover's responses to each.

Theorem 2. *When 1) using B -rewinding secure delayed-input WIAoK¹⁶ as underlying WIAoK and 2) the prover generates a pseudo-random function in the first round and generates the randomness used in the third round through the pseudo-random function on input the second round message, the resulting scheme satisfies WZK property against both multi-bit-output distinguisher and non-adaptive B -rewinding malicious verifier.*

Proof. Since the prover now uses a pseudo-random function to generate its randomness (that's a standard technique used in resettable ZK[10]), we can assume that in each rewinding the prover uses fresh randomness. From the B -rewinding security of underlying WI (recall that two-round WI protocol is naturally rewinding secure) and WIAoK protocol, transcripts produced using the witness w and those produced using the trapdoor (extracted during simulation) are indistinguishable. The key difficulty here is that, the simulation relies on the distinguisher that only outputs after seeing the *complete* view of adversary. And the adversary may generate different FHE public keys pk_i and send $\text{FHE.Enc}_{pk_i}(r_i)$ in different sessions, it breaks the procedure of using $\{r_i\}$ to directly generate complete valid proofs homomorphically.

Therefore, to complete the simulation, we introduce a recursive simulation strategy. Denote by $\text{Sim}_{\text{basic}}$ the simulator shown in the proof of Theorem 2. We construct several simulators as follows:

Sim_0 : Run the first two steps of $\text{Sim}_{\text{basic}}$ and receiving B second-round messages from $\mathcal{V}^*$. For each WI'_2 contained these B second-round messages, generate the corresponding WI'_3 using the randomness that generated pk as the witness. Set trans the transcript (additionally containing all these WI'_3) so far and output it.

From $k = B$ to 1, we construct Sim_k as follows.

$\text{Sim}_k(\text{trans})$: The simulator has input trans , which containing the entire transcript in the first $k - 1$ rewinding sessions and the WI'_3 components for all sessions.

1. Sim_k first learns the second-round message in the k -th rewinding session from trans and checks that whether the proof $(\text{WI}'_1, \text{WI}'_2)$ in this session is acceptable. If so, continue; otherwise, set $\perp$ the third-round message of the k -th session, update the transcript trans and go to step 4.
2. Run step 5 – 8 of $\text{Sim}_{\text{basic}}$ using the messages from the k -th session, with the exception that
 - (a) In step 5(a)v, $\text{HS}(r)$ sets generated $(pk', c'_{\text{KE}}, ct, \text{WI}'_2)$ and the corresponding WI'_3 in trans as the third-round message of the k -th session, and updates trans . If $k < B$, $\text{HS}(r)$ additionally runs $\text{Sim}_{k+1}(\text{trans})$ to obtain the remaining view of $\mathcal{V}^*$. $\text{HS}(r)$ runs D on the complete view.
 - (b) In step 5(b), run the extractor with input $1^{nB/\epsilon}$ rather than $1^{n/\epsilon}$.
3. After obtaining $(pk', c_{\text{KE}}, ct, \text{WI}'_2)$ in the previous step, set it the third-round message of the k -th session as $(pk', c_{\text{KE}}, ct, \text{WI}'_2, \text{WI}'_3)$ where WI'_3 is the k -th such proof recorded in trans . Update trans .

¹⁶ Goyal et al. [23, Figure 1] demonstrate how to instantiate such a scheme from a non-interactive commitment; their construction enjoys the proof of knowledge property (see Section 3.3 of [23]).

4. Sim_k proceeds as follows:
 - (a) If $k = B$, output the view of $\mathcal{V}^*$;
 - (b) Otherwise, run Sim_{k+1} on input trans .

Now, the final simulator Sim can be constructed directly via running Sim_0 firstly and then running Sim_1 to obtain the view of $\mathcal{V}^*$.

The proof is similar to the one of Theorem 1.

We first construct HSIM_1 that is identical to Sim except that all WI'_3 are generated using w as witness. From the B -rewinding security of underlying WIAoK, we have that HSIM_1 is indistinguishable from Sim .

Then for each k from 1 to B , we construct a series hybrid simulation: For $i \in [2, 8]$, HSIM_i^k is the same with HSIM_{i-1}^k (or HSIM_8^{k-1} when $i = 2$) except for the k -th rewinding session, HSIM_i^k makes the same changes as the one in the HSIM_i in the proof of theorem 1.

For the same reason shown in the proof of Theorem 1, all neighbor hybrid simulations are computationally indistinguishable for D (except for each k , HSIM_7^k and HSIM_8^k are ϵ/B indistinguishable for D). Furthermore, HSIM_8^B is identical to the view of $\mathcal{V}^*$ interacting with an honest prover. $\square$

Remark 2. A key observation here is that, Sim_0 constructed above treats V^* only in a black-box manner. And Sim_1 only uses the code of V^* *after it has generated the second-round messages* for all sessions. The simulator constructed for Theorem 1 can be similarly divided into $(\text{Sim}_0, \text{Sim}_1)$, where Sim_0 consists of the first four steps and Sim_1 the remainder. This property is important in the proofs of the next section.

5 Weak Non-Malleable Zero-Knowledge Argument

In this section, we show how to construct a weak zero-knowledge argument that satisfies weak simulation-extractability, which implies a weak NMZK argument. The construction using the weak zero-knowledge argument $\text{WZK} = (\text{WZK}_1, \text{WZK}_2, \text{WZK}_3)$ constructed in Section 4, and the weak NMC scheme $\text{NMC} = (\text{NMC}_1, \text{NMC}_2, \text{NMC}_3)$ w.r.t. synchronizing adversaries constructed in [24].

Theorem 3. *The protocol constructed in Fig.5 is a weak zero-knowledge argument satisfying weak simulation-extractability.*

Proof. The completeness, soundness and weak zero-knowledge properties follow directly from the properties of the underlying weak zero-knowledge protocol and the hiding of the underlying commitment.

Now we show how it achieves weak simulation-extractability. Consider a MIM adversary $\mathcal{A}(x, x', id, id', aux)$ that interacts with an honest prover $P(x, w)$ in a left session using identity id , then tries to prove a statement x' when interacting with an honest verifier in a different (right) session with different identity id' . Since our protocol only has three rounds, we only need to consider two kinds of

Non-Malleable Weak Zero-Knowledge Argument

Public Input: $x \in \mathcal{L}$, and an identity id used by underlying NMC scheme.

Prover's Private Input: A witness w s.t. $(x, w) \in R_{\mathcal{L}}$.

- **Prover Message:** Generate the first-round message WZK_1 of WZK and the first-round message NMC_1 of NMC committing w . Send (WZK_1, NMC_1) to the verifier.
- **Verifier Message:** Generate and send (WZK_2, NMC_2) to the prover.
- **Prover Message:** Generate the third-round message NMC_3 of NMC, and the third-round message WZK_3 of WZK for statement (x, NMC_1, NMC_2, NMC_3) , which is true if and only if there exists w s.t. $(x, w) \in R_{\mathcal{L}}$ and $(NMC_1, NMC_2, NMC_3) \in NMC(w)$. Send (WZK_3, NMC_3) to the verifier.
- **Verifier's Output:** Output 1 iff. the WZK argument (WZK_1, WZK_2, WZK_3) for the statement (x, NMC_1, NMC_2, NMC_3) is accepted.

Fig. 5. Weak Non-Malleable Zero-Knowledge Argument

adversaries, synchronizing adversaries and non-synchronizing adversaries¹⁷. By non-synchronizing, we mean that the MIM adversary completes the entire left session before beginning the right session.

Non-synchronizing adversary: Given the non-synchronizing MIM adversary $\mathcal{A}$ and a distinguisher D , we first construct a malicious WZK verifier $\mathcal{V}^*$ and a 2-bit-output distinguisher D^* as follows:

$\mathcal{V}^*$:

1. Upon receiving WZK_1 from the external prover, compute the first-round message NMC_1 of NMC committing $0^{|w|}$. Run $\mathcal{A}$ with auxiliary aux and fresh randomness $r_{\mathcal{A}}$ itself, and send (WZK_1, NMC_1) to $\mathcal{A}$.
2. Upon receiving (WZK_2, NMC_2) from $\mathcal{A}$, compute the third-round message NMC_3 of NMC honestly. Set the statement of WZK as $x_{wzk} := (x, NMC_1, NMC_2, NMC_3)$ and send WZK_2 to the external prover.
3. Eliminate the randomness used to generate the NMC, and denote the current state (and hence the residual algorithm) of V^* as V_{res}^* (this step is merely for convenience in describing the subsequent proof).
4. Upon receiving WZK_3 from the prover, send (WZK_3, NMC_3) to $\mathcal{A}$.

¹⁷ For an adversary $\mathcal{A}$ that runs synchronously with some probability and non-synchronously otherwise, we can construct a synchronizing adversary $\mathcal{A}'$ as follows: $\mathcal{A}'$ simply runs $\mathcal{A}$; if $\mathcal{A}'$ detects that $\mathcal{A}$ has employed an asynchronous strategy, it aborts and outputs a special abort symbol. Similarly, a non-synchronizing adversary $\mathcal{A}''$ can be constructed. To build the simulation-extractor for $\mathcal{A}$, we first run the one for $\mathcal{A}'$. If it outputs anything other than the special abort symbol, we output its result. Otherwise, we repeatedly run the simulation-extractor for $\mathcal{A}''$ until it does not output the special abort symbol, and then output its result.

5. Act as an honest right-session verifier V and interact with $\mathcal{A}$ until the right session ends. Denote the right-session statement as x' . If the right session is acceptable, extract the witness of x' using the extractor of the underlying WZK scheme in the right session.
6. Output the joint view of MIM adversary $\mathcal{A}$ and right-session verifier V , and the extracted witness.

D^* : On input the output of V^* , run D with the joint view and obtain its output b_1 . Check whether the right session is acceptable and whether the extracted witness is valid, set $b_2 := 0$ iff. the right session is acceptable and the extracted witness is invalid. Output $b_1 \parallel b_2$.

Let Sim_{WZK} be the WZK simulator against above V^* and D^* with distinguish gap $\epsilon/2$. Then, Sim_{WZK} is also the simulation-extractor $\mathcal{SE}$ against $\mathcal{A}$ and D with the distinguish/probability gap ϵ . To show it, we construct several hybrids:
 HYD_0 : Same as $\mathcal{SE}$.

HYD_1 : Same as $\mathcal{SE}$ except that in the construction of V^* , the commitment NMC in the left session is now committing w .

HYD_2 : Same as HYD_1 except that it uses an honest prover to interact with the malicious verifier rather than $\mathcal{SE}$.

Lemma 4. $HYD_0 \stackrel{c}{\approx} HYD_1$

Proof. Above lemma follows from the hiding property of underlying NMC in the left session. It is important to point out that, our non-black-box WZK simulator can be divided into two parts, Sim_0 and Sim_1 (see Remark 2). Sim_0 treats V^* in a black-box manner and Sim_1 only uses the code of V^* after the second round, i.e., V_{res}^* , when *all the messages of NMC in the left session has been fixed and both V^* and D will not use the randomness of NMC any longer*, which makes the reduction feasible.

Specifically, if there exists a distinguisher distinguishing above two hybrids, we can construct a malicious NMC receiver as follows (Fig.5), where V^* generates NMC via interacting with a external NMC sender (the view of Sim , including V_{res}^* , remains the same).

Lemma 5. $HYD_1 \approx_{D^*, \epsilon/2} HYD_2$

The above lemma follows directly from the weak zero-knowledge property of the underlying WZK directly.

Hence $HYD_0 \approx_{D^*, \epsilon} HYD_2$. Then weak simulation-extractibility holds because:

1. The first-output-bit of $D^*(HYD_2)$ (resp. $D^*(HYD_0)$) is exactly the output of D on input the real (resp. simulated) joint view.
2. The probability that second output bit of $D^*(HYD_0)$ is 0 is exactly the probability that $\mathcal{SE}$ fails to extract the witness while the right interactive is acceptable. From the adaptive argument-of-knowledge property of underlying WZK, the second output bit of $D^*(HYD_2)$ is almost always 0.

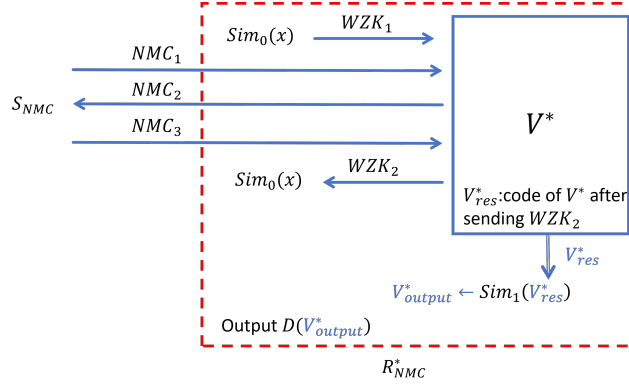


Fig. 6. Construction of R_{NMC}^*

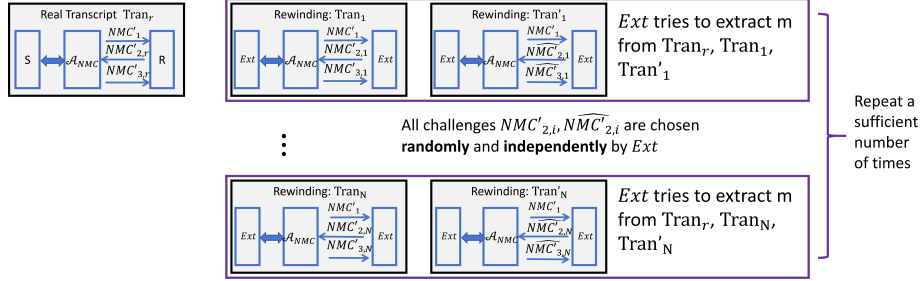


Fig. 7. Extractor of the weak NMC from [24]

Synchronizing adversary: Prior to describing our simulation-extractor, we will first introduce the extraction mechanism of the weak NMC from [24]. The execution flow is illustrated in the following figure (Fig.7):

Roughly, upon receiving a real NMC transcript Tran_r , the extractor first rewinds the MIM adversary $\mathcal{A}_{NMC}$ to the point just before $\mathcal{A}_{NMC}$ receives the second-round messages in the right session. The extractor then sends a fresh second-round NMC challenge (consisting only of several random coins) and receives its response (while simultaneously continuing to interact with $\mathcal{A}_{NMC}$ in the left session) to obtain Tran_1 . Subsequently, the extractor repeats this process to obtain Tran'_1 and attempts to extract the committed message m . This entire process is repeated a sufficient number of times to enhance the success probability.

Denote by EXT the complete extractor and Ext the per-attempt extractor (the extractor invoked in a single attempt). From the definition of weak NMC and the Theorem 2 of [24], for any polynomial inverse ϵ (require that MIM adversary generates valid commitment in the right session with probability larger than ϵ), the complete extractor parameterized by ϵ satisfies $\Pr[EXT = m | \text{the right session commits a valid message } m] > 1 - \epsilon$. Since EXT runs Ext independently for N times where N is a specific polynomial in λ, ϵ^{-1} (i.e., $N = \omega(\lambda^{11} \epsilon^{-18})$), we directly have that $\Pr[Ext = m | \text{the right session commits a valid}$

message $m]$ $> poly_{ext}(\lambda^{-1}, \epsilon)$ for some polynomial $poly_{ext}$ independent with MIM adversary.

Then, given the synchronizing MIM NMZK adversary $\mathcal{A}$ and a distinguisher D , we construct a malicious WZK verifier V^* and a three-bit-output distinguisher D^* as follows, where V^* is a non-adaptive B -rewinding verifier:

V^* :

1. Upon receiving WZK_1 from the external prover, compute the first message NMC_1 of NMC committing $0^{|w|}$. Run $\mathcal{A}$ with auxiliary aux and fresh randomness $r_{\mathcal{A}}$ itself and send (WZK_1, NMC_1) in the left session to $\mathcal{A}$.
2. Upon receiving (WZK'_1, NMC'_1) in the right session from $\mathcal{A}$, act as an honest verifier (using fresh randomness r_V) and send (WZK'_2, NMC'_2) to $\mathcal{A}$.
3. Upon receiving (WZK_2, NMC_2) from $\mathcal{A}$, compute the third message NMC_3 of NMC. Set the statement of WZK as $x' := (x, NMC_1, NMC_2, NMC_3)$. Send WZK_2 to the external prover.
4. Upon receiving WZK_3 from the prover, send (WZK_3, NMC_3) to $\mathcal{A}$. When receiving (WZK'_3, NMC'_3) from $\mathcal{A}$, set $View_{\mathcal{A}, V} := (aux, r_{\mathcal{A}}, r_V, WZK, NMC, WZK', NMC')$ where WZK containing (WZK_1, WZK_2, WZK_3) and similar as NMC, WZK', NMC' .
5. Run the per-attempt extractor Ext of NMC on transcript (NMC, NMC') :
 - (a) When Ext tries to non-adaptively rewind the adversary to the second round and send fresh NMC'_2 , generate fresh WZK'_2 , rewind $\mathcal{A}$ to the second round and send (WZK'_2, NMC'_2) to $\mathcal{A}$.
 - (b) Upon receiving (WZK_2, NMC_2) from $\mathcal{A}$, send NMC_2 to Ext and receive NMC_3 from it. Send WZK_2 to the external prover.
 - (c) Upon receiving WZK_3 from the prover, send (WZK_3, NMC_3) to $\mathcal{A}$. Upon receiving (WZK'_3, NMC'_3) from $\mathcal{A}$, send NMC'_3 to Ext .
6. After Ext outputs w' committed in NMC' , V^* outputs $View_{\mathcal{A}, V}$ and w' .

As discussed regarding the NMC extractor, steps 5.(a) and (b) can be executed simultaneously with step 3, effectively making V^* a non-adaptive adversary. The detailed execution flow of V^* is illustrated as follows (Fig.8).

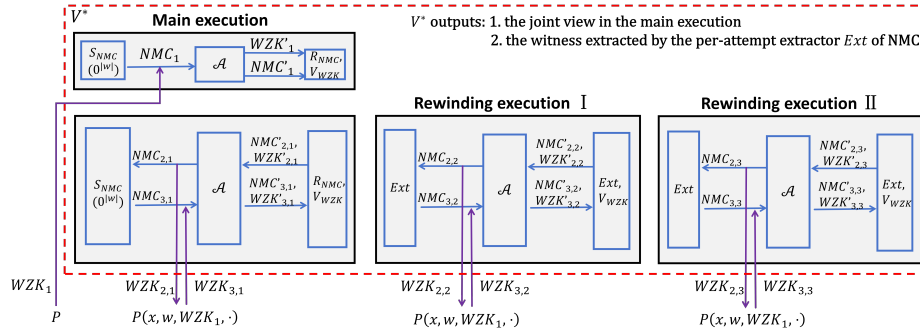


Fig. 8. Execution flow of V^*

D^* : On input the output of V^* , run D with the joint view $\text{View}_{\mathcal{A},V}$ and obtain its output b_1 . Check whether the right session is acceptable and whether the extracted witness is valid, and obtain b_2, b_3 respectively. Output $b_1 \| b_2 \| b_3$.

Define $\mu = \min(\text{poly}_{ext}(\epsilon/3, \lambda^{-1}), \epsilon/3)$. Suppose that Sim_{WZK} be the simulator of WZK against V^* and D^* and distinguish gap $\mu/2$. Now we construct the simulation-extractor $\mathcal{SE}$ against $\mathcal{A}$ and D with the distinguish/probability gap ϵ for our construction as follows:

$\mathcal{SE}$:

1. Run Sim_{WZK} and obtain $(\text{View}_{\mathcal{A},V}, w')$. Set $\text{View}_{\mathcal{A},V}$ as the its first part of output.
2. If the right interactive in $\text{View}_{\mathcal{A},V}$ is acceptable, continue; otherwise, set $\perp$ as the second part of the output and go to step 4.
3. Run Sim_{WZK} with fresh randomness for T times, where $T \geq 2\lambda \cdot \mu^{-1}$. If there are any extracted $w' \neq \perp$, set it as the second part of output.
4. Output the two-part result.

Now we construct several hybrid simulators of Sim_{WZK} to conclude the proof:

HYD_0 : Same as Sim_{WZK} .

HYD_1 : Same as Sim_{WZK} except that in the construction of V^* , the commitment NMC in the left session of the main execution is now committing w .

HYD_2 : Same as HYD_1 except that it uses an honest prover to interact with the malicious verifier and return the output of V^* .

Lemma 6. $\text{HYD}_0 \stackrel{c}{\approx} \text{HYD}_1$

Proof. The above lemma follows the same proof strategy of Lemma.4. Again, the simulator only uses the code of V^* after all the messages of NMC in the left session of the main execution has been fixed. The lemma then follows from the hiding property of NMC.

Lemma 7. $\text{HYD}_1 \approx_{D^*, \mu/2} \text{HYD}_2$

The above lemma follows from the non-adaptive B -rewinding security of the underlying WZK.

Now we show that $\mathcal{SE}$ is a valid simulation-extractor for our construction.

The first requirement of simulation-extractibility follows that: the $\text{View}_{\mathcal{A},V}$'s distributions of $\mathcal{SE}$ is identical to the one of HYD_0 , $\text{HYD}_0 \approx_{D^*, \epsilon} \text{HYD}_2$, the first-output-bit of D^* is $D(\text{View}_{\mathcal{A},V})$, and the $\text{View}_{\mathcal{A},V}$'s distribution of HYD_2 is exactly the one of real executions.

As for the second requirement, we first assume that the probability that MIM adversary $\mathcal{A}$ convinces the verifier in the right session with probability larger than $\epsilon/2$, otherwise the probability that the right session in the output view $\text{View}_{\mathcal{A},V}$ is acceptable is already less than ϵ (recall that whether the right session is acceptable is the second output bit of D^*).

Now, from the soundness of underlying WZK in the main execution of V^* in HYD_2 (which now interacts with an honest prover), the probability that NMC in the right session in $\text{View}_{\mathcal{A},V}$ commits to a valid w' is larger than $\epsilon/2 - \text{negl}$. Therefore the per-attempt extractor Ext of NMC would extract w' with probability larger than μ in HYD_2 . Therefore, each running of Sim_{WZK} will extract w' with probability larger than $\mu/2$. Since $\mathcal{SE}$ will run Sim_{WZK} for $T \geq 2\lambda \cdot \mu^{-1}$ when the verifier in the right session accepts the proof, $\mathcal{SE}$ would extract w' with probability $1 - \text{negl}(\lambda)$, which concludes the proof. $\square$

Acknowledgments. We would like to thank the anonymous reviewers for their valuable suggestions, especially the reviewer who proposed the “multi-distinguisher” notion, which helped clarify the relationships among different distinguisher-dependent definitions. We are supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB0690200), the National Key Research and Development Program of China (Grant No. 2023YFB4503203), the National Natural Science Foundation of China (Grant Nos. 62372447 and 62502513), the Postdoctoral Fellowship Program of China Postdoctoral Science Foundation (Grant No. GZC20252181).

References

1. Bagheri, K., Kohlweiss, M., Siim, J., Volkhov, M.: Another look at extraction and randomization of groth’s zk-snark. In: Borisov, N., Díaz, C. (eds.) Financial Cryptography and Data Security - 25th International Conference, FC 2021, Virtual Event, March 1-5, 2021, Revised Selected Papers, Part I. Lecture Notes in Computer Science, vol. 12674, pp. 457–475. Springer (2021). https://doi.org/10.1007/978-3-662-64322-8_22, https://doi.org/10.1007/978-3-662-64322-8_22
2. Bellare, M., Jakobsson, M., Yung, M.: Round-optimal zero-knowledge arguments based on any one-way function. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 280–305. Springer (1997)
3. Bellare, M., Palacio, A.: Towards plaintext-aware public-key encryption without random oracles. In: Lee, P.J. (ed.) Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3329, pp. 48–62. Springer (2004). https://doi.org/10.1007/978-3-540-30539-2_4
4. Bitansky, N., Brakerski, Z., Kalai, Y., Paneth, O., Vaikuntanathan, V.: 3-message zero knowledge against human ignorance. In: Theory of Cryptography - TCC’16. pp. 57–83. LNCS 9985, Springer (2016). https://doi.org/10.1007/978-3-662-53641-4_3
5. Bitansky, N., Canetti, R., Paneth, O., Rosen, A.: On the existence of extractable one-way functions. In: Proceedings of the 45th Annual ACM Symposium on the Theory of Computing - STOC’14. pp. 505–514. ACM Press (2014). <https://doi.org/10.1145/2591796.2591859>
6. Bitansky, N., Kalai, Y.T., Paneth, O.: Multi-collision resistance: a paradigm for keyless hash functions. In: Diakonikolas, I., Kempe, D., Henzinger, M. (eds.) Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018. pp. 671–684.

- ACM (2018). <https://doi.org/10.1145/3188745.3188870>, <https://doi.org/10.1145/3188745.3188870>
7. Bitansky, N., Khurana, D., Paneth, O.: Weak zero-knowledge beyond the black-box barrier. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing - STOC'19. pp. 1091–1102. ACM press (2019). <https://doi.org/10.1145/3313276.3316382>
 8. Botta, V., Ciampi, M., Orsini, E., Siniscalchi, L., Visconti, I.: Black-box (and fast) non-malleable zero knowledge. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part IX. Lecture Notes in Computer Science, vol. 14928, pp. 458–490. Springer (2024), https://doi.org/10.1007/978-3-031-68400-5_14
 9. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE . SIAM J. Comput. **43**(2), 831–871 (2014). <https://doi.org/10.1137/120868669>, <https://doi.org/10.1137/120868669>
 10. Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: Yao, F.F., Luks, E.M. (eds.) Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21–23, 2000, Portland, OR, USA. pp. 235–244. ACM (2000). <https://doi.org/10.1145/335305.335334>, <https://doi.org/10.1145/335305.335334>
 11. Ciampi, M., Ostrovsky, R., Siniscalchi, L., Visconti, I.: Four-round concurrent non-malleable commitments from one-way functions. In: Advances in Cryptology—CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20–24, 2017, Proceedings, Part II 37. pp. 127–157. Springer (2017)
 12. Deng, Y.: Individual simulations. In: Advances in Cryptology – ASIACRYPT'20. pp. 805–836. LNCS 12493, Springer (2020). https://doi.org/10.1007/978-3-030-64840-4_27
 13. Deng, Y., Zhang, X.: Knowledge encryption and its applications to simulatable protocols with low round-complexity. In: Agrawal, S., Lin, D. (eds.) Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part III. Lecture Notes in Computer Science, vol. 13793, pp. 334–362. Springer (2022), https://doi.org/10.1007/978-3-031-22969-5_12
 14. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings. pp. 24–43. Lecture Notes in Computer Science, Springer (2010). https://doi.org/10.1007/978-3-642-13190-5_2, https://doi.org/10.1007/978-3-642-13190-5_2
 15. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography (extended abstract). In: Koutsougeras, C., Vitter, J.S. (eds.) Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5–8, 1991, New Orleans, Louisiana, USA. pp. 542–552. ACM (1991), <https://doi.org/10.1145/103418.103474>
 16. Dwork, C., Naor, M., Reingold, O., Stockmeyer, L.: Magic functions: In memoriam: Bernard m. dwork 1923–1998. Journal of the ACM (JACM) **50**(6), 852–921 (2003)
 17. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: Ortiz, H. (ed.) Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13–17, 1990, Baltimore, Maryland, USA. pp. 416–426. ACM (1990), <https://doi.org/10.1145/100216.100272>

18. Gentry, C.: A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University, USA (2009), <https://searchworks.stanford.edu/view/8493082>
19. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009. pp. 169–178. ACM (2009), <https://doi.org/10.1145/1536414.1536440>
20. Goldreich, O., Oren, Y.: Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology* **7**(1), 1–32 (1994)
21. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. *SIAM Journal on Computing* **18**(1), 186–208 (1989). <https://doi.org/10.1137/0218012>
22. Goyal, V., Pandey, O., Richelson, S.: Textbook non-malleable commitments. In: Wichs, D., Mansour, Y. (eds.) Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18–21, 2016. pp. 1128–1141. ACM (2016), <https://doi.org/10.1145/2897518.2897657>
23. Goyal, V., Richelson, S.: Non-malleable commitments using goldreich-levin list decoding. In: Zuckerman, D. (ed.) 60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9–12, 2019. pp. 686–699. IEEE Computer Society (2019), <https://doi.org/10.1109/FOCS.2019.00047>
24. Goyal, V., Richelson, S., Rosen, A., Vald, M.: An algebraic approach to non-malleability. In: 2014 IEEE 55th Annual Symposium on Foundations of Computer Science. pp. 41–50. IEEE (2014)
25. Hada, S., Tanaka, T.: On the existence of 3-round zero-knowledge protocols. In: Krawczyk, H. (ed.) Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23–27, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1462, pp. 408–423. Springer (1998). <https://doi.org/10.1007/BFB0055744>
26. Jain, A., Kalai, Y.T., Khurana, D., Rothblum, R.: Distinguisher-dependent simulation in two rounds and its applications. In: Advances in Cryptology - CRYPTO'17. pp. 158–189. LNCS 10402, Springer (2017). https://doi.org/10.1007/978-3-319-63715-0_6
27. Khurana, D.: Round optimal concurrent non-malleability from polynomial hardness. In: Kalai, Y., Reyzin, L. (eds.) Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12–15, 2017, Proceedings, Part II. Lecture Notes in Computer Science, vol. 10678, pp. 139–171. Springer (2017), https://doi.org/10.1007/978-3-319-70503-3_5
28. Khurana, D., Malavolta, G., Tomer, K.: Weak zero-knowledge via the goldreich-levin theorem. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 142–173. Springer (2023)
29. Kim, A., Liang, X., Pandey, O.: A new approach to efficient non-malleable zero-knowledge. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 13510, pp. 389–418. Springer (2022), https://doi.org/10.1007/978-3-031-15985-5_14
30. Kiyoshima, S.: Four-round statistical non-malleable zero-knowledge. In: Applebaum, B., Lin, H.R. (eds.) Theory of Cryptography - 23rd International Conference, TCC 2025, Aarhus, Denmark, December 1–5, 2025, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 16271, pp. 138–170. Springer

- (2025). https://doi.org/10.1007/978-3-032-12290-2_5, https://doi.org/10.1007/978-3-032-12290-2_5
31. Kosba, A.E., Zhao, Z., Miller, A., Qian, Y., Chan, T.H., Papamanthou, C., Pass, R., Shelat, A., Shi, E.: How to use snarks in universally composable protocols. *IACR Cryptol. ePrint Arch.* **2015**, 1093 (2015), <http://eprint.iacr.org/2015/1093>
 32. Lindell, Y., Pinkas, B.: A proof of security of yao's protocol for two-party computation. *J. Cryptol.* **22**(2), 161–188 (2009). <https://doi.org/10.1007/S00145-008-9036-8>, <https://doi.org/10.1007/s00145-008-9036-8>
 33. Ostrovsky, R., Paskin-Cherniavsky, A., Paskin-Cherniavsky, B.: Maliciously circuit-private FHE. In: Garay, J.A., Gennaro, R. (eds.) *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference*, Santa Barbara, CA, USA, August 17–21, 2014, *Proceedings, Part I*. pp. 536–553. *Lecture Notes in Computer Science*, Springer (2014). https://doi.org/10.1007/978-3-662-44371-2_30, https://doi.org/10.1007/978-3-662-44371-2_30
 34. Pass, R., Rosen, A.: New and improved constructions of nonmalleable cryptographic protocols. *SIAM J. Comput.* **38**(2), 702–752 (2008), <https://doi.org/10.1137/060671553>
 35. Yao, A.C.: Protocols for secure computations (extended abstract). In: *23rd Annual Symposium on Foundations of Computer Science*, Chicago, Illinois, USA, 3–5 November 1982. pp. 160–164. *IEEE Computer Society* (1982). <https://doi.org/10.1109/SFCS.1982.38>, <https://doi.org/10.1109/SFCS.1982.38>

A Other Preliminaries

We put other preliminaries used in this paper here.

A.1 Non-Interactive Commitment Scheme

Definition 12 (Non-Interactive Commitment Scheme). A non-interactive commitment scheme is a pair of algorithms $(\text{COM}, \text{DECOM})$:

- $\text{COM}(1^\lambda, m, r) \rightarrow (c, d)$: On input a message m and a randomness r , the sender $\mathcal{S}$ outputs a commitment c and a decommitment information d which is required to open the commitment.
- $\text{DECOM}(c, d) \rightarrow \{0, 1\}$: On input the commitment c and the decommitment information d , the receiver $\mathcal{R}$ outputs a bit indicating whether it accepts or not.

The non-interactive commitment scheme satisfies two properties:

- *Hiding*: For any messages $m_0 \neq m_1$,

$$\{c_0 \mid (c_0, d_0) \leftarrow \text{COM}(1^\lambda, m_0, r_0)\}_{r_0} \stackrel{c}{\approx} \{c_1 \mid (c_1, d_1) \leftarrow \text{COM}(1^\lambda, m_1, r_1)\}_{r_1}.$$

- *Binding*: For any PPT adversary $\mathcal{A}$,

$$\Pr \left[\begin{array}{l} \text{DECOM}(c, d_0) = 1 \\ \wedge \text{DECOM}(c, d_1) = 1 \end{array} \middle| (c, d_0, d_1) \leftarrow \mathcal{A}(1^\lambda) \right] < \text{negl}(\lambda).$$

Remark 3. In the commitment scheme used in Section 4, $d = (m, r)$, and the DECOM algorithm is essentially running the COM algorithm and then checking whether the output commitment is equal to the input c .

A.2 Zero-Knowledge Argument of Knowledge

In a zero-knowledge argument for language $\mathcal{L}$, the prover $\mathcal{P}$ with input (x, w) tries to convince the verifier $\mathcal{V}$ with input x that it knows a witness w such that $(x, w) \in R_{\mathcal{L}}$, during this process, any malicious verifier cannot gain any other knowledge.

Definition 13 (Interactive Zero-Knowledge Argument). A protocol $\langle \mathcal{P}, \mathcal{V} \rangle$ for relation $R_{\mathcal{L}}$ is an interactive zero-knowledge argument, if it satisfies the following properties:

- *Completeness*. For any $\lambda \in \mathbb{N}$, $x \in \mathcal{L} \cap \{0, 1\}^\lambda$, and w such that $(x, w) \in R_{\mathcal{L}}$,

$$\Pr[\langle \mathcal{P}(w), \mathcal{V} \rangle(x) = 1] > 1 - \text{negl}(\lambda).$$

- *Soundness*: For any PPT prover $\mathcal{P}^*$, $\lambda \in \mathbb{N}$, $\lambda \in \mathbb{N}$, and $x \in \{0,1\}^\lambda$, if $x \notin \mathcal{L}$, then

$$\Pr[\langle \mathcal{P}^*, \mathcal{V} \rangle(x) = 1] \leq \text{negl}(1^\lambda).$$

- *Zero-Knowledge*: There exists a simulator Sim , such that for any PPT $\mathcal{V}^*$, any distinguisher D , any $x \in \mathcal{L} \cap \{0,1\}^\lambda$,

$$\left| \Pr[D(x, \text{Out}_{\mathcal{V}^*}(\langle \mathcal{P}(w), \mathcal{V}^* \rangle(x))) = 1] - \Pr[D(x, \text{Sim}^{\mathcal{V}^*}(x)) = 1] \right| < \text{negl}(\lambda).$$

Argument of Knowledge. For an delayed-input argument, the adaptive argument of knowledge property is defined in similar way to that for non-delayed-input arguments, except that here we need to deal with the issue that *the statement may be chosen adaptively*. We follow the definition in [5,4,13] to define three-round adaptive argument of knowledge.

Definition 14 (Adaptive Argument of Knowledge). A three-round delayed-input argument system with message (a_1, a_2, a_3) for NP language L is called an adaptive argument of knowledge if there exists an oracle extractor Ext and a polynomial poly such that for any PPT malicious prover P^* , any noticeable function ϵ and any security parameter $\lambda \in \mathbb{N}$:

$$\begin{aligned} \text{if } \Pr \left[V(x, (a_1, a_2, a_3)) = 1 \mid \begin{array}{l} a_1 \leftarrow P^* \\ a_2 \leftarrow V(\lambda, a_1) \\ x, a_3 \leftarrow P^*(a_1, a_2) \end{array} \right] &\geq \epsilon(\lambda), \\ \text{then } \Pr \left[\begin{array}{l} V(x, (a_1, a_2, a_3)) = 1 \wedge \\ \text{Ext}^{P^*}(x, (a_1, a_2, a_3)) \notin R_L(x) \end{array} \mid \begin{array}{l} a_1 \leftarrow P^* \\ a_2 \leftarrow V(\lambda, a_1) \\ x, a_3 \leftarrow P^*(a_1, a_2) \end{array} \right] &\leq \text{negl}(\lambda), \end{aligned}$$

where Ext runs in expected time bounded by $\text{poly}(\lambda)/\epsilon$.

A.3 Fully Homomorphic Encryption

A fully homomorphic encryption (FHE) public key scheme [19,9] consists of four algorithms: key generation, encryption, decryption, and ciphertext evaluation.

- $\text{KeyGen}(1^\lambda) \rightarrow (pk, sk)$: This algorithm takes a security parameter λ as input and outputs the public key pk for encryption and evaluation, and the secret key sk for decryption.
- $\text{Enc}(pk, m) \rightarrow c$: This algorithm takes a plaintext message m and the public key pk as inputs and outputs the corresponding ciphertext c .
- $\text{Dec}(sk, c) \rightarrow m$: This algorithm takes a ciphertext c and the secret key sk as inputs and outputs the corresponding plaintext message m .
- $\text{Eval}(pk, c_1, \dots, c_n, f) \rightarrow c_f$: This algorithm takes a set of ciphertexts $c_1, \dots, c_n$, a function $f : M^n \rightarrow M$ to be computed, and the public key pk , and outputs the ciphertext result c_f .

Security. A valid FHE scheme must first be a secure encryption scheme, typically satisfying the property of indistinguishability under chosen-plaintext attack (IND-CPA). In addition to security, the FHE scheme must also meet correctness compactness and circuit privacy requirements:

Correctness. Given a key pair (pk, sk) generated by the key generation algorithm, and a function $f : M^n \rightarrow M$, for any valid ciphertexts $c_1, c_2, \dots, c_n$, with corresponding plaintexts $m_1, m_2, \dots, m_n$, i.e., $m_i = \text{Dec}(sk, c_i)$, running the ciphertext evaluation algorithm

$$\text{Eval}(pk, c_1, \dots, c_n, f) \rightarrow c_f$$

should result in a ciphertext c_f such that

$$\text{Dec}(sk, c_f) = f(m_1, m_2, \dots, m_n).$$

Compactness. Compactness refers to the property that the size of the output ciphertext c_f of the evaluation algorithm should not depend on the function f . **(Passive) Circuit Privacy[19,18].** An FHE scheme $(\text{KeyGen}, \text{Enc}, \text{Dec}, \text{Eval})$ is said to satisfy *circuit privacy* if for any $(pk, sk) \in \text{KeyGen}$, any polynomial size circuit C , any messages $m_1, \dots, m_n$ and any ciphertexts $c_i \in \text{Enc}(pk, m_i)$ for $1 \leq i \leq n$, the following distributions (over the random coins in Enc, Eval) are (statistically) indistinguishable:

$$\text{Enc}(pk, f(m_1, \dots, m_n)) \approx \text{Eval}(pk, C, c_1, \dots, c_n)$$

A FHE scheme satisfying above circuit privacy can be directly constructed from assumptions over ideal lattice[19].

Furthermore, the above definition only requires that the properties hold for well-formed public keys, secret keys, and ciphertexts. As described in [14,33], it is always possible to convert a compact homomorphic encryption scheme into one that offers similar passive circuit privacy using Yao’s garbled circuits[35,32] and a two-flow oblivious transfer protocol. More specifically, replace the (evaluated) encryption with a Yao’s garbled circuit for decryption. Our scheme can also employ such an FHE scheme, replacing every occurrence of the ciphertext ct'_{FHE} in both the construction and the proof with the Yao’s garbled circuit, without requiring any further modifications to the proof. Consequently, we can construct the required FHE encryption scheme from LWE assumptions with a corresponding circular security assumption[9].

B From Trapdoor Permutation to Knowledge Encryption

The work in [28] presented a three-round weak zero-knowledge protocol based on trapdoor injective one-way permutation, which we now generalize through a knowledge encryption framework. Building upon this foundation, we demonstrate how to systematically construct knowledge encryption from trapdoor one-way permutation. Our construction only uses the following ingredient:

- A commitment scheme that satisfies perfect binding, computational hiding $\text{COM} = (\text{COM.Com}, \text{COM.Open})$.
- A trapdoor one way permutation $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ with trapdoor $t \in \{0, 1\}^l$, and h is the Goldreich-Levin hardcore predicate of f .

where $l = \text{poly}(n)$.

Knowledge Encryption from Trapdoor One-way Permutation

Key Generation:

- The encrypting party EP randomly selects $r_0 \in \{0, 1\}^n$, computes the commitment $c = \text{COM.Com}(r_0)$, and sends c to the decrypting party DP .
- Upon receiving com , the decrypting party DP randomly selects $r_1 \in \{0, 1\}^n$ and sends (f, r_1) to the encrypting party EP , where (f, r_1) denotes the public key pk for the EP .

Encryption:

- To encrypt message $m \in \{0, 1\}$, random chose $r, s \in \{0, 1\}^n$ and compute $c = (c_1, c_2, c_3) = (h(s, r) \oplus m, r, f(s))$. Then EP sends c to the DP .

Decryption:

- Upon receiving the ciphertext tuple $c = (c_1, c_2, c_3)$, the decryption process proceeds by first applying the trapdoor to invert c_3 and recover the secret value s .
- With s now available and $c_2 = r$, the receiver computes the hardcore predicate evaluation $h(s, r)$, and computes $m = h(r, s) \oplus c_1$.

Fig. 9. Knowledge encryption from trapdoor one-way permutation

To demonstrate witness extractability, we construct the following security game:

Witness extractability game

- 1 The challenger C randomly selects $r_0 \in \{0, 1\}^n$, computes the commitment $c = \text{COM.Com}(r_0)$, and sends c to the adversary $\mathcal{A}$.
- 2 The adversary return (f, r_1) to the challenger C .
- 3 The challenger C randomly selects $b \in \{0, 1\}, r, s \in \{0, 1\}^n$, then compute $c = (h(s, r) \oplus b, r, f(s))$. The challenger C sends c to the adversary $\mathcal{A}$.
- 4 Upon receiving c , the adversary $\mathcal{A}$ output b' . We say $\mathcal{A}$ wins if and only if $b' = b$.

Fig. 10. Witness extractability game

Lemma 8. *There exists a PPT extractor E satisfying that, for any polynomial-size adversary $\mathcal{A} = \{\mathcal{A}_\lambda\}_\lambda$ and inverse polynomial ϵ and , if*

$$|\Pr[\mathcal{A} \text{ wins the witness extractability game}]| \geq \epsilon,$$

then

$$\Pr[E^{D^\lambda}(r_0, r_1) = r' \wedge f(r') = r_0 \oplus r_1] \geq 1 - \text{negl}(\lambda),$$

where E runs in time polynomial in ϵ^{-1} and λ .

Proof. Below we give the construction of extractor E :

- 1 E prepares the Goldreich-Levin set R of size m as follows: Randomly select $k = \lfloor \log m \rfloor + 1$ strings $\{\hat{r}_0, \dots, \hat{r}_{k-1}\}$ where each $\hat{r}_i \in \{0, 1\}^n$ and, Then for each non-empty subset $S \subset [0, k-1]$, compute $\tilde{r}_S = \bigoplus_{j \in S} \hat{r}_j$.
- 2 E randomly selects $r_0 \in \{0, 1\}^n$, computes the commitment $c = \text{COM.Com}(r_0)$, and sends c to the adversary $\mathcal{A}$.
- 3 After receiving (f, r) , For each of $i \in [m]$, E performs the following steps:
 - Randomly select a bit $a \in \{0, 1\}$ and send the challenge $c = (a, \tilde{r}_i, r_0 \oplus r_1)$ to adversary $\mathcal{A}$.
 - Upon receiving b' from the adversary $\mathcal{A}$, compute the hardcore predicate value as $HCP = a \oplus b'$
- 4 Compute $f^{-1}(r_0 \oplus r_1)$ by Goldreich-Levin theorem.

If adversary $\mathcal{A}$ has decryption capability, then $\mathcal{A}$ can necessarily compromise the hardcore predicate of the one-way permutation. Furthermore, by applying the Goldreich-Levin theorem, this capability enables E to invert the one-way permutation with non-negligible probability.

C Proof of Corollary 1

Corollary 1 *A standard knowledge encryption scheme is naturally a knowledge encryption against multi-bit-output distinguisher defined in Definition 11.*

Proof. Let D be a ciphertext-distinguisher whose output length is k . We define 2^k single-bit-output distinguishers $D'_0, \dots, D'_{2^k-1}$ as follows. On input a ciphertext c , distinguisher D'_i proceeds:

1. Run $D(c)$ to obtain an output $output \in \{0, 1\}^c$.
2. If $output$ equals the binary representation of i , D'_i outputs 1; otherwise D'_i outputs 0.

Because k is a fixed constant, the support of D is also constant. If the statistical distance between the distributions $D_\lambda(\text{KE.Enc}(\text{pk}^*, 0))$ and $D_\lambda(\text{KE.Enc}(\text{pk}^*, 1))$ is at least ϵ , then

$$\sum_{X \in \text{supp}(D)} \frac{1}{2} \left| \Pr[D(\text{KE.Enc}(\text{pk}^*, 0)) = X] - \Pr[D(\text{KE.Enc}(\text{pk}^*, 1)) = X] \right| \geq \epsilon(\lambda).$$

Consequently, there exists an index $i^* \in [0, 2^k - 1]$ such that

$$\left| \Pr[D(\text{KE.Enc}(\text{pk}^*, 0)) = \text{bin}(i^*)] - \Pr[D(\text{KE.Enc}(\text{pk}^*, 1)) = \text{bin}(i^*)] \right| \geq \frac{\epsilon(\lambda)}{2^c}.$$

where $\text{bin}(i^*)$ is the binary representation of i^* . Therefore, for the single-bit-output distinguisher D'_{i^*} , we have that:

$$\left| \Pr[D'_{i^*}(\text{KE.Enc}(\text{pk}^*, 0)) = 1] - \Pr[D'_{i^*}(\text{KE.Enc}(\text{pk}^*, 1)) = 1] \right| \geq \frac{\epsilon(\lambda)}{2^c}.$$

Therefore, one can simultaneously run the origin KE extractor $E^{D'_i}(\text{pk}^*, 1^{\frac{2^c}{\epsilon(\lambda)}})$ for all D'_i , $i \in [0, 2^k - 1]$ (as i^* is unknown for extraction) to recover the witness for the statement $x \in L$, which is encoded in the public key pk^* . $\square$

D Detailed Proof of the (Adaptive) Argument-of-Knowledge Property in Theorem 1

Suppose the malicious prover $\mathcal{P}^*$ interacts with $\mathcal{V}$ and generates an acceptable transcript with a polynomial-inverse probability. We consider the following extractor $E^{\mathcal{P}^*}$.

On input the statement x and a transcript tr , the extractor $E^{\mathcal{P}^*}(x, tr)$:

1. Run the extractor of the underlying WIAoK for language $\mathcal{L}_3$ which extracts a witness of $\mathcal{L}_3$.
2. If w is extracted such that $(x, w) \in \mathcal{L}$, $E^{\mathcal{P}^*}$ outputs w .
3. if sk is extracted, $E^{\mathcal{P}^*}$ decrypt ct_1 to obtain and output w

Now, we show that the probability tr is acceptable but $E^{\mathcal{P}^*}(x, tr)$ fails to output a valid w is negligible.

First, from the (adaptive) argument of knowledge property of underlying delayed-input WIAoK, the probability that tr is acceptable but $E^{\mathcal{P}^*}$ fails to extract either a valid witness w or a valid secret key sk from the extractor of WIAoK is negligible.

Second, from the soundness of WI, the probability that tr is acceptable and $E^{\mathcal{P}^*}$ extracts sk , but none of the following events happens is negligible:

1. The decryption of ct_0 is sk' satisfying that $r = \text{Dec}_{sk'}(ct'_0)$, and $com = \text{COM}(0; r)$.
2. The decryption of ct_1 is w such that $(x, w) \in R_{\mathcal{L}}$.
3. The decryption of ct_2 is $c_{w_{\text{KE}}}$ such that $(x_{\text{KE}}, w_{\text{KE}}) \in R_{\mathcal{L}_{\text{KE}}}$ where $w_{\text{KE}} = \text{FHE.Dec}(c_{w_{\text{KE}}})$.

Therefore, to complete the proof it suffices to establish the following claim:

The probability that $E^{\mathcal{P}^*}$ extracts sk and (either the first or the last event happens) is negligible.

Assume that $E^{\mathcal{P}^*}(x, tr)$ extracts sk and the first event happens with a noticeable probability. Without loss of generality, let $\mathcal{P}^*$ be deterministic. Then, we modify the verifier (used for generating tr) as follows: First, use the extracted sk as a witness to conclude the generation of WI_2^1 . Then, change the FHE encryption of r into encryption of another random string (recall that the extract

of sk doesn't need to decrypt c'_{FHE}). And third, change the commitment com of 0 into the one of 1. From WI, IND-CPA security of FHE, and hiding property of commitment, the probability of first event happens should still be a noticeable probability. However, com now is a commitment of 1, meeting a contraction.

Assume that $E^{\mathcal{P}^*}$ extracts sk and the third event happens with a noticeable probability. Then, we can modify the verifier so that it generates simulated knowledge encryption public keys from statements chosen from the hard distribution. From the public-key simulation property, the probability of the third event happening should still be a noticeable probability. It means that $E^{\mathcal{P}^*}$ extracts a valid witness for the statement chosen from a hard distribution, meeting a contraction. $\square$