Black-Box Impossibilities of Obtaining 2-Round Weak ZK and Strong WI from Polynomial Hardness

Susumu Kiyoshima

NTT Research
susumu.kiyoshima@ntt-research.com

September 23, 2021

Abstract

We study the problem of obtaining 2-round interactive arguments for NP with *weak zero-knowledge (weak ZK)* [Dwork et al., 2003] or with *strong witness indistinguishability (strong WI)* [Goldreich, 2001] under polynomially hard falsifiable assumptions. We consider both the *delayed-input* setting [Jain et al., 2017] and the standard non-delayed-input setting, where in the delayed-input setting, (i) prover privacy is only required to hold against *delayed-input verifiers* (which learn statements in the last round of the protocol) and (ii) soundness is required to hold even against *adaptive provers* (which choose statements in the last round of the protocol).

Concretely, we show the following black-box (BB) impossibility results by relying on standard cryptographic primitives (such as one-way functions and trapdoor permutations).

1. It is impossible to obtain 2-round delayed-input weak ZK arguments under polynomially hard falsifiable assumptions if BB reductions are used to prove soundness. This result holds even when non-black-box techniques are used to prove weak ZK.
2. It is impossible to obtain 2-round non-delayed-input strong WI arguments and 2-round publicly verifiable delayed-input strong WI arguments under polynomially hard falsifiable assumptions if a natural type of BB reductions, called "oblivious" BB reductions, are used to prove strong WI. (Concretely, a BB reduction for strong WI is called oblivious if it is black-box not only about the cheating verifier but also about the statement distributions.)
3. It is impossible to obtain 2-round delayed-input strong WI arguments under polynomially hard falsifiable assumptions if BB reductions are used to prove both soundness and strong WI (the BB reductions for strong WI are required to be oblivious as above). Compared with the above result, this result no longer requires public verifiability in the delayed-input setting.

This article is a full version of the following article: Black-Box Impossibilities of Obtaining 2-Round Weak ZK and Strong WI from Polynomial Hardness, TCC 2021, ©IACR 2021.

Contents

1 Introduction

Zero-knowledge (ZK) proofs and arguments have been extensively used in cryptography due to their powerful security. Informally, their security guarantees that an honest prover can convince a verifier of the validity of a statement without revealing anything beyond it. More formally, the *zero-knowledgeness* (ZK) guarantees that for any verifier there exists a (efficient) *simulator* such that for any distinguisher, the output of the simulator (which is given only a statement and is executed alone) is indistinguishable from the output of the verifier (which interacts with an honest prover that proves the validity of the statement).

The powerful security of ZK protocols[1] however comes with a cost: it is known that ZK protocols require at least three rounds for any language outside of BPP [GO94]. This lower bound limits the applicability of ZK protocols since many applications require that the number of interactions is at most two rounds.

Fortunately, it has been shown that by carefully weakening the definition of ZK, we can obtain several useful security notions that can be achieved in less than three rounds.[2] Such security notions include *witness indistinguishability (WI)* [FS90, DN07], *witness hiding (WH)* [FS90, BKP19], *strong WI* [Gol01, JKKR17], *weak ZK* [DNRS03, BKP19], *super-polynomial-time simulation (SPS) ZK* [Pas03], and *ZK against bounded-size verifiers* [BCPR16].

Still, the state-of-the-art is not satisfactory since many of the existing 2-round constructions for these notions are based on super-polynomially hard assumptions (i.e., assumptions against adversaries that run in fixed super-polynomial time) [JKKR17, BKP19, Pas03, KS17, BGI+17, BCPR16, KKS18, BFJ+20, GJJM20, LVW20]. Indeed, for some of the above-listed security notions (such as strong WI and weak ZK as explained below), no 2-round construction is currently known under polynomially hard standard assumptions. This situation is frustrating since for WI, it has long been known that 2-round (or even non-interactive) constructions can be obtained from polynomially hard standard assumptions [DN07, GOS12].

In this work, we study whether the use of super-polynomially hard assumptions is unavoidable in these existing 2-round protocols, focusing on the cases of weak ZK and strong WI.

Weak ZK. Weak ZK is defined identically with ZK except that the order of the quantifier is reversed, i.e., it is now required that for any verifier V^* and any distinguisher D, there exists a simulator S (which may depend on both V^* and D) such that the distinguisher D cannot distinguish the output of the simulator S from the output of the verifier V^*. Weak ZK is weaker than ZK but still implies WI and WH.

Currently, two positive results are known about 2-round weak ZK, where one is shown in the *delayed-input setting* [JKKR17]—i.e., in the setting where (i) an honest verifier can create its first-round message without knowing the statement to be proven, (ii) soundness is required to hold even against any *adaptive prover*, which can choose the statement to prove in the last round of the protocol (i.e., after seeing the verifier's first-round message), and (iii) weak ZK is only required to hold against any *delayed-input verifier*, which creates its first-round message without knowing the statement to be proven. Note that the delayed-input setting and the standard (non-delayed-input) setting are incomparable since the former considers soundness against stronger provers whereas the latter considers weak ZK against stronger verifiers.

In the delayed-input setting, Jain et al. [JKKR17] constructed a 2-round argument that satisfies *distributional ϵ-weak ZK* for any inverse polynomial ϵ, where distributional ϵ-weak ZK is weaker than the standard weak ZK in that (i) the simulator is only required to work for random statements that are sampled from a distribution $\mathcal{D}$ and (ii) the distinguishing gap between the verifier's output and the simulator's output is only bounded by the inverse polynomial ϵ (the simulator is allowed to depend on both $\mathcal{D}$ and ϵ). The security of their protocol is proven under a quasi-polynomially hard assumption (concretely, the existence of 2-round oblivious transfer protocols that are secure against malicious polynomial-time receivers and quasi-polynomial-time semi-honest senders).

In the standard setting, Bitansky et al. [BKP19] constructed a 2-round argument that is ϵ-weak ZK for any inverse polynomial ϵ under super-polynomially hard assumptions (such as quasi-polynomial hardness of the Learning with Errors problem).[3]

[1] We use the term "zero-knowledge protocols" to refer to both zero-knowledge proofs and zero-knowledge arguments.

[2] Throughout this paper, we focus on interactive proofs/arguments for all NP.

[3] Weak ZK is defined slightly differently in Bitansky et al. [BKP19], where weak ZK is defined to be satisfying ϵ-weak ZK for any inverse polynomial ϵ We follow other prior works [DNRS03, CLP15, JKKR17] and require the distinguishing gap to be negligible.

Strong WI. Strong WI guarantees that for any two indistinguishable distributions $\mathcal{D}^0, \mathcal{D}^1$ over statements, no verifier can distinguish a proof for a random statement $x \leftarrow \mathcal{D}^0$ from a proof for a random statement $x \leftarrow \mathcal{D}^1$. A typical application of strong WI is proof of honest behaviors: for example, when a strong WI protocol is used to prove that a commitment is correctly generated, it directly guarantees that the hiding property of the commitment is preserved. (In contrast, the standard WI does not guarantee anything when the commitment is perfectly binding.)

In the delayed-input setting, Jain et al. [JKKR17] constructed a 2-round strong WI argument under a quasi-polynomially hard assumption (concretely, the existence of 2-round oblivious transfer protocols that are secure against malicious polynomial-time receivers and quasi-polynomial-time semi-honest senders). In the standard setting, the above-mentioned result about 2-round weak ZK [BKP19] also holds for 2-round strong WI since ϵ-weak ZK implies strong WI.

1.1 Our Results

At a high level, we show impossibility results about obtaining 2-round weak ZK and strong WI protocols under "standard assumptions" by using "standard techniques." Following previous works (e.g., [GW11]), we formalize "standard assumptions" and "standard techniques" by using *falsifiable assumptions* and *black-box (BB) reductions*, respectively. Roughly speaking, (polynomially hard) falsifiable assumptions are the assumptions that are modeled as interactive games between a polynomial-time adversary and a polynomial-time challenger, where a falsifiable assumption (C, c) is considered true if no polynomial-time adversary can make the challenger C output 1 with probability non-negligibly higher than the threshold $c \in [0, 1]$. Essentially all standard cryptographic assumptions are falsifiable, including both general assumptions (e.g., the existence of one-way functions) and concrete ones (e.g., the RSA, DDH, and LWE assumptions). Regarding BB reductions, we consider two types of BB reductions, one is for soundness and the other is for strong WI. These two types are explained below with our results.

BB impossibility of 2-round weak ZK. Our first impossibility result is about obtaining 2-round weak ZK protocols while using BB reductions in the proof of soundness. Here, BB reductions are defined for soundness as follows: for a 2-round weak ZK argument (P, V), we say that *the soundness of* (P, V) *is proven by a BB reduction based on a falsifiable assumption* (C, c) if there exists a polynomial-time oracle machine (or *BB reduction*) R such that for any verifier V^* that breaks the soundness of (P, V), the machine R^{V^*} breaks the assumption (C, c).

Theorem (informal). *Assume the existence of one-way functions. Then, there exists an NP language* $\mathbf{L}$ *such that if*

- *there exists a 2-round delayed-input distributional* ϵ*-weak ZK argument for* $\mathbf{L}$ *and*
- *its adaptive soundness is proven by a BB reduction based on a falsifiable assumption* (C, c)*,*

then the assumption (C, c) *is false.*

(The formal statement is given as Theorem 1 in Section 7.) We note that using BB reductions in the proof of soundness is quite common, and in particular, BB reductions are used in the proof of soundness in the above-mentioned two positive results of 2-round weak ZK [JKKR17, BKP19].[4] (In fact, to the best of our knowledge, currently there do not exist any non-BB technique that can be used to prove the soundness of 2-round interactive arguments.) This result therefore matches with the positive result of [JKKR17] (note that this result holds even for the distributional ϵ-weak ZK version of weak ZK) and thus explains why the use of super-polynomial-time hardness is required in [JKKR17]. Finally, we note that this result holds even when non-BB techniques are used in the proof of weak ZK.

Let us explain informally what this result says about the difficulty of obtaining 2-round weak ZK protocols under polynomially hard assumptions. First, in the delayed-input setting, this result directly explains the difficulty: to overcome this result, we need to prove the soundness of 2-round arguments by using non-BB techniques,[5] but given the state-of-the-art, this approach unfortunately seems to require novel techniques. Second,

[4] In [BKP19], weak ZK is proven by a non-black-box technique, but soundness is proven by a BB reduction.

[5] It is easy to verify that for interactive proofs (rather than arguments) in the delayed-input setting, the classical impossibility result of 2-round ZK [GO94] can be extended to 2-round weak ZK.

even in the standard setting, this result partially explains the difficulty: to overcome this result, we need to consider protocols that are inherently not adaptively sound, and thus, we need to be careful when using the popular *FLS paradigm* [FLS99]. Indeed, if we naively use the FLS paradigm (where the verifier sets up a "trapdoor statement" in the first round and the prover gives a WI proof in the second round to prove that either the actual statement is true or the trapdoor statement is true), it is often the case that the first-round message is independent of the statement and as a result adaptive soundness holds whenever soundness holds.

BB impossibility of 2-round strong WI (non-delayed-input or publicly verifiable). Our second impossibility result is about obtaining 2-round strong WI protocols while using a certain type of BB reductions in the proof of strong WI. Specifically, we consider BB reductions that we call *oblivious BB reductions*, which are defined roughly as follows: for a 2-round strong WI protocol (P, V), we say that *the strong WI of* (P, V) *is proven by an oblivious BB reduction based on a falsifiable assumption* (C, c) if there exists a polynomial-time oracle machine (or oblivious BB reduction) R such that for any verifier V^* that breaks the strong WI of (P, V) w.r.t. some distributions $\mathcal{D}^0, \mathcal{D}^1$, the machine $R^{V^*,\mathcal{D}^0,\mathcal{D}^1}$ either breaks the assumption (C, c) or distinguishes the distributions $\mathcal{D}^0$ and $\mathcal{D}^1$. We note that R is oblivious to the distributions $\mathcal{D}^0, \mathcal{D}^1$ in the sense that R is defined before the distributions $\mathcal{D}^0, \mathcal{D}^1$ are specified.[6] (We emphasize that during the execution, R is given oracle access to $\mathcal{D}^0, \mathcal{D}^1$.)

Theorem (informal). *Assume the existence of CCA-secure public-key encryption schemes. Then, there exists an NP language* $\mathbf{L}$ *such that the following hold.*

1. *If there exists a 2-round (non-delayed-input) strong WI protocol for* $\mathbf{L}$ *and its strong WI is proven by an oblivious BB reduction based on a falsifiable assumption* (C, c)*, then the assumption* (C, c) *is false.*

2. *If there exists a 2-round publicly verifiable delayed-input strong WI protocol*[7] *for* $\mathbf{L}$ *and its strong WI is proven by an oblivious BB reduction based on a falsifiable assumption* (C, c)*, then the assumption* (C, c) *is false.*

(The formal statement is given as Theorem 3 and Theorem 4 in Section 7.) We note that obliviousness is a natural property for BB reductions, and for example oblivious reductions are used in the above-mentioned positive result of 2-round strong WI [JKKR17] and in the trivial proof showing that ZK implies strong WI [Gol01, Proposition 4.6.3]. (Indeed, we are not aware of any non-oblivious reduction that can be used to prove strong WI for NP w.r.t. all distributions.) We also note that the second part of this result in particular holds for strong WI versions of *ZAPs* [DN07] and *ZAP arguments* [BFJ+20, GJJM20, LVW20].

Let us explain informally what this result says about the difficulty of obtaining 2-round strong WI protocols under polynomially hard assumptions. In particular, since the only way to overcome this result is to use non-BB or non-oblivious techniques in the proof of strong WI (as long as we consider non-delayed-input or publicly verifiable protocols), let us explain informally the difficulty of using these two types of techniques.

- Let us first see the difficulty of using non-BB techniques. We first note that for witness hiding, there exists a non-BB technique [BKP19] such that (i) it can be used to prove the prover privacy of 2-round arguments under polynomially hard assumptions and (ii) we can use it while proving soundness under polynomially hard assumptions (such as the existence of *witness encryption schemes* [GGSW13]). Unfortunately, the usage of this technique in the witness hiding setting strongly relies on a certain property of witness hiding (concretely, the property that a successful cheating verifier against witness hiding outputs a witness for the statement). As a result, it is currently unclear whether we can use this (or any other) non-BB technique in the setting of strong WI while proving soundness under polynomially hard assumptions.

- Let us next see the difficulty of using non-oblivious techniques. The main difficulty is that when we consider strong WI that holds for all NP w.r.t. all distributions over statements, we currently do not have any technique that makes non-oblivious use of distributions. As a result, it is currently unclear whether any non-oblivious technique is useful to obtain 2-round strong WI under polynomially hard assumptions.

[6] This type of obliviousness is considered previously on BB reductions for witness hiding [HRS09].

[7] that is, a 2-round delayed-input strong WI protocol such that anyone can decide whether a proof is accepting or not given the protocol transcript (without knowing the verifier randomness).

BB impossibility of 2-round strong WI (delayed-input). Our third impossibility result is about obtaining 2-round strong WI arguments while using BB reductions in the proofs of soundness and strong WI. The motivation behind this result is to give an impossibility result about 2-round privately verifiable delayed-input strong WI protocols (for which the above result does not hold).

Theorem (informal). *Assume the existence of trapdoor permutations. Then, there exists an NP language* $\boldsymbol{L}$ *such that if*

- *there exists a 2-round delayed-input strong WI argument for* $\boldsymbol{L}$*,*
- *its soundness is proven by a BB reduction based on a falsifiable assumption* (C, c)*, and*
- *its strong WI is proven by an oblivious BB reduction based on a falsifiable assumption* (C', c')*,*

then either the assumption (C, c) *or the assumption* (C', c') *is false.*

(The formal statement is given as Theorem 2 in Section 7.) We note that this result matches with the positive result of [JKKR17] since BB reductions are used for both soundness and strong WI in the result of [JKKR17] (the one for strong WI is oblivious). Thus, this result explains why the use of super-polynomial-time hardness is required in [JKKR17].

Let us explain informally what this result says about the difficulty of obtaining 2-round strong WI protocols under polynomially hard assumptions. Compared with the above result, this result holds even for 2-round privately verifiable delayed-input strong WI protocols, but it holds only when BB reductions are used in the proof of soundness. Still, it seems reasonable to think that this result explains the difficulty of obtaining 2-round strong WI protocols almost as strongly as the above one since, as in the case of 2-round weak ZK, novel techniques are likely to be required to obtain 2-round strong WI protocols without using BB reductions in the proof of soundness.

Summary. In Table 1, we summarize the settings that we consider in our impossibility results (standard v.s. delayed-input) for each combination of the types of reductions (BB and non-BB reductions for soundness and weak ZK, and oblivious BB, non-oblivious BB, and non-BB reductions for strong WI). For example, "delayed-input" in the cell that corresponds to BB for soundness and BB for weak ZK indicates that one of our results (concretely, the first result) shows the impossibility of 2-round delayed-input weak ZK arguments when BB techniques are used for both soundness and weak ZK.

Table 1: Summary of the settings that we consider in our impossibility results.

		weak ZK		strong WI	
		BB	non-BB	obl. BB	non-obl. BB / non-BB
Soundness	BB	delayed-input	delayed-input	standard delayed-input	
	non-BB			standard pub-verifiable delayed-input	

2 Our Techniques

2.1 BB Impossibility of 2-Round Delayed-Input Weak ZK

We first explain how we obtain our BB impossibility result about 2-round delayed-input weak ZK. This result is technically less involved and is used in a non-modular way in one of our BB impossibility results about strong WI.

At a very high level, we obtain our result about weak ZK by obtaining a BB impossibility result about (t, ϵ)-*zero-knowledge* [CLP15], which is defined identically with the standard zero-knowledge except that (i) the definition is parameterized by a polynomial t and an inverse polynomial ϵ, (ii) the running time of the distinguisher is bounded by t, and (iii) the distinguishing gap is bounded by ϵ (the simulator is allowed to depend on both t and ϵ). Note that (t, ϵ)-ZK is defined with the same order of quantifier as the standard ZK (i.e.,

in the form "$\forall V^* \exists S \forall D \ldots$") and thus seems much stronger than weak ZK. Nonetheless, it is known that weak ZK implies (t, ϵ)-ZK for every polynomial t and inverse polynomial ϵ (with no modification to the protocol) [CLP15]. Thus, to obtain a BB impossibility result on weak ZK, it suffices to obtain it on (t, ϵ)-ZK.

Before explaining how we obtain a BB impossibility result about (t, ϵ)-ZK, let us explain a subtle difference between (t, ϵ)-ZK and the standard ZK. Specifically, we note that in (t, ϵ)-ZK (in particular, the one that is defined in [CLP15] and shown to be implied by weak ZK), the indistinguishability between a real proof and simulation is only guaranteed to hold against uniform distinguishers, i.e., distinguishers that take no auxiliary input other than the one that is given to the verifier and the simulator. In the standard ZK (where the running time of the distinguisher can be longer than that of the simulator), we can think as if the distinguisher takes an additional auxiliary input since we can assume without loss of generality that a suffix of the common auxiliary input is only read by the distinguisher (see, e.g., [Gol01, Section 4.3.3]). Yet, in (t, ϵ)-ZK (where the simulator can depend on t and thus can run longer than the distinguisher), we cannot use this argument anymore.

Somewhat surprisingly, this subtle difference causes difficulties when we try to obtain impossibility results about (t, ϵ)-ZK by using known techniques. Indeed, the classical impossibility result of 2-round ZK [GO94] does not hold for (t, ϵ)-ZK exactly due to this difference. Also, known techniques in BB impossibility literature, such as those that have been used for the BB impossibility of other 2-round interactive protocols [GW11, CLMP12, DJKL12], also require non-uniform indistinguishability and thus cannot be used for (t, ϵ)-ZK directly.

Roughly speaking, we overcome the difficulties as follows. First, we observe that weak ZK implies (t, ϵ)-ZK with non-uniform indistinguishability if we allow the simulator of (t, ϵ)-ZK to run in a "pre-processing" manner, i.e., in a manner that the simulator is computationally unbounded before receiving the statement. (More specifically, the simulator is separated into two parts, a *pre-processing simulator* and a *main simulator*, where the pre-processing simulator is computationally unbounded and creates short trapdoor information without knowing the statement, and the main simulator takes the statement along with the trapdoor information and simulates the verifier's output in polynomial time.) Second, we observe that the *meta-reduction* techniques, which have been used for the BB impossibility of other 2-round interactive protocols [GW11, CLMP12, DJKL12], can be used naturally to obtain a BB impossibility result about 2-round delayed-input pre-processing (t, ϵ)-ZK. Let us now give more details about these two steps below.

Step 1. Showing that weak ZK implies pre-processing (t, ϵ)-ZK. We first note that, as already observed in [CLP15], weak ZK implies (t, ϵ)-ZK with non-uniform indistinguishability if we allow the simulator of (t, ϵ)-ZK to be non-uniform, i.e., if we only require that for each auxiliary input z_V to the verifier there exists an auxiliary input z_S to the simulator such that on input z_S, the simulator works for any (non-uniform) distinguisher. Now, the problem is of course that it is in general not possible to compute a "good" z_S from z_V efficiently. Thus, we give the simulator unbounded computing power so that it can compute a good z_S from z_V by brute force. To make sure that the simulator can compute a good z_S before receiving the statement, we further weaken the definition of (t, ϵ)-ZK and consider the distributional version of it, where the simulator is only required to work for random statements that are sampled from a certain distribution. Since it is now sufficient for the simulator to find a good z_S for random statements, the simulator can find it before obtaining the actual statement.

Step 2. Showing BB impossibility of pre-processing (t, ϵ)-ZK. We obtain a BB impossibility result about 2-round delayed-input pre-processing (t, ϵ)-ZK by appropriately modifying a proof that is given in [CLMP12, DJKL12] for the BB impossibility of 2-round *super-polynomial-simulation (SPS) ZK*, where the simulator is allowed to run in fixed super-polynomial time T.[8] To see how we modify the proof of [CLMP12, DJKL12], consider for example a step in the proof where it is shown that the simulator creates an accepting proof for a false statement. In [CLMP12, DJKL12], this property is shown by (i) first observing that the simulator creates an accepting proof for a true statement due to the indistinguishability of simulation (note that an honest prover does so with probability 1 by completeness) and then (2) observing that the simulator creates an accepting proof even for a false statement due to the indistinguishability between true and false statements (since the simulator runs in super-polynomial time T, it is assumed that true and false statements are indistinguishable in $\mathsf{poly}(T)$ time). Clearly, when the simulator is computationally unbounded, the second step of this argument fails

[8] In SPS ZK, the simulator is usually computationally bounded by a fixed moderate super-polynomial (e.g., a quasi-polynomial) but it can use its super-polynomial-time computing power arbitrarily. In pre-processing (t, ϵ)-ZK, the simulator is computationally unbounded but it can use its super-polynomial-time computing power only before receiving the statement.

since the simulator can distinguish true and false statements by brute force. Nevertheless, in the pre-processing model, we can still show the same property by relying on the non-uniform polynomial-time indistinguishability of true and false statements. To see this, observe that the non-uniform indistinguishability guarantees that no polynomial-time algorithm can distinguish true and false statements even when it is given any auxiliary input that is computed independently of the statement. This guarantee is clearly sufficient to show that when the main simulator in the pre-processing model creates an accepting proof for a true statement, it creates an accepting proof even for a false statement.

2.2 BB Impossibility of 2-Round Strong WI

We next explain how we obtain our BB impossibility results about 2-round strong WI. Recall that unlike our result about weak ZK, our results about strong WI require that BB reductions are used in the proof of prover privacy (i.e., strong WI). To explain the reason behind this difference, we note that when proving our result about weak ZK (or more precisely about (t, ϵ)-ZK), we first obtain a simulator by relying on the ZK property and then design a successful cheating prover against soundness by using it (the resultant cheating prover is then combined with the soundness BB reduction to show that the underlying assumption must be false). Clearly, this approach does not work for strong WI since simulators are not guaranteed to exist. Thus, we instead assume the existence of a strong WI reduction (which acts as a prover when interacting with a verifier through oracle queries) and use it to obtain a cheating prover. More details are given below.

Non-interactive strong WI. First, as a warm-up, we explain how we can obtain a BB impossibility result about non-interactive strong WI. In particular, we show that the strong WI of non-interactive arguments cannot be proven by oblivious BB reductions based on falsifiable assumptions.

At a high level, the proof proceeds as follows. Recall that an oblivious BB reduction R_{SWI} for strong WI has the following property: for any verifier V^* that breaks strong WI w.r.t. some distributions $\mathcal{D}^0, \mathcal{D}^1$ over statements (meaning that V^* can distinguish a proof π for statement $x \leftarrow \mathcal{D}^0$ and a proof π for statement $x \leftarrow \mathcal{D}^1$), the reduction $R_{\mathrm{SWI}}^{V^*}$ either breaks the underlying assumption (C, c) or distinguishes $\mathcal{D}^0$ and $\mathcal{D}^1$.[9] First, we observe that $R_{\mathrm{SWI}}^{V^*}$ breaks the assumption (C, c) rather than distinguishes $\mathcal{D}^0$ and $\mathcal{D}^1$. Assume for contradiction that $R_{\mathrm{SWI}}^{V^*}$ distinguishes $\mathcal{D}^0$ and $\mathcal{D}^1$, and assume without loss of generality that V^* aborts when it receives a proof that is not accepting. Now, intuitively, the assumption that $R_{\mathrm{SWI}}^{V^*}$ can distinguish $x \leftarrow \mathcal{D}^0$ and $x \leftarrow \mathcal{D}^1$ seems to imply that R_{SWI} sends x to V^* along with an accepting proof (since otherwise V^* seems useless); if so, we can use R_{SWI} to break soundness by arguing that even when x is a false statement, R_{SWI} still sends x to V^* along with an accepting proof. A problem is that R_{SWI} might distinguish $x \leftarrow \mathcal{D}^0$ and $x \leftarrow \mathcal{D}^1$ by sending a related statement x' to V^* without directly sending x. We solve this problem by designing a "non-malleable" language $\mathbf{L}$, which guarantees that R_{SWI} cannot distinguish $x \leftarrow \mathcal{D}^0$ and $x \leftarrow \mathcal{D}^1$ even when it sends a related statement x' to V^*. After showing $R_{\mathrm{SWI}}^{V^*}$ breaks the assumption (C, c), we conclude that the assumption (C, c) must be false by observing that we can design as V^* a specific cheating verifier that breaks strong WI w.r.t. $\mathcal{D}^0, \mathcal{D}^1$ efficiently.

More specifically, the proof proceeds as follows. Consider an NP language $\mathbf{L}$ that contains all the encryptions of 0 and 1 of a CCA-secure public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, i.e., $\mathbf{L} := \{(\mathsf{pk}, \mathsf{ct}) \mid \exists r \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, 0; r) \text{ or } \mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, 1; r)\}$. Also, for each public key pk of PKE and each $b \in \{0, 1\}$, consider the distribution $\mathcal{D}_{\mathsf{pk}}^b$ that outputs a random encryption of b under the public-key pk, i.e., $\mathcal{D}_{\mathsf{pk}}^b := \{(\mathsf{pk}, \mathsf{ct}) \mid \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, b)\}$. (We emphasize that $\mathcal{D}_{\mathsf{pk}}^b$ always outputs a ciphertext ct that is encrypted with the hardwired public key pk.) Assume that there exist a non-interactive argument (P, V) for $\mathbf{L}$ and an oblivious BB reduction R_{SWI} for showing the strong WI of (P, V) based on a falsifiable assumption (C, c). Note that this assumption implies that for any public key pk and any verifier V^* that breaks the strong WI of (P, V) w.r.t. $\mathcal{D}_{\mathsf{pk}}^0, \mathcal{D}_{\mathsf{pk}}^1$, the reduction $R_{\mathrm{SWI}}^{V^*}$ either breaks the assumption (C, c) or distinguishes $\mathcal{D}_{\mathsf{pk}}^0$ and $\mathcal{D}_{\mathsf{pk}}^1$. Now, our goal is to show that the assumption (C, c) is false. Toward this goal, for each public-key–secret-key pair $(\mathsf{pk}, \mathsf{sk})$, we consider the following verifier $V_{\mathrm{SWI}}^* = V_{\mathrm{SWI}}^*[\mathsf{pk}, \mathsf{sk}]$ against the strong WI of (P, V).

- **Verifier V_{SWI}^*:** Given a statement $(\mathsf{pk}', \mathsf{ct})$ and a proof π from the prover, return the decryption result $b \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ to the prover if $\mathsf{pk} = \mathsf{pk}'$ holds and π is an accepting proof for $(\mathsf{pk}', \mathsf{ct})$, and return a random bit otherwise.

[9] Formally, R_{SWI} also has oracle access to $\mathcal{D}^0$ and $\mathcal{D}^1$, but we ignore it for simplicity in this overview.

Note that for any $(\mathsf{pk}, \mathsf{sk})$, the verifier V^*_{SWI} breaks the strong WI w.r.t. $\mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}$ due to the correctness of PKE. Thus, for any $(\mathsf{pk}, \mathsf{sk})$, the reduction $R^{V^*_{\mathrm{SWI}}}_{\mathrm{SWI}}$ either breaks the assumption (C, c) or distinguishes $\mathcal{D}^0_{\mathsf{pk}}$ and $\mathcal{D}^1_{\mathsf{pk}}$. Now, we observe that the assumption (C, c) is false unless we can use the reduction $R^{V^*_{\mathrm{SWI}}}_{\mathrm{SWI}}$ to break either the CCA security of PKE or the soundness of (P, V). Consider the following three cases for random pk.

- **Case 1.** $R^{V^*_{\mathrm{SWI}}}_{\mathbf{SWI}}$ **breaks the assumption** (C, c)**.** In this case, it follows immediately that the assumption (C, c) is false since we can emulate V^*_{SWI} for R_{SWI} efficiently by using sk for random $(\mathsf{pk}, \mathsf{sk})$.
- **Case 2.** $R^{V^*_{\mathrm{SWI}}}_{\mathbf{SWI}}(\mathsf{pk}, \mathsf{ct})$ **distinguishes whether** $(\mathsf{pk}, \mathsf{ct}) \leftarrow \mathcal{D}^0_{\mathsf{pk}}$ **or** $(\mathsf{pk}, \mathsf{ct}) \leftarrow \mathcal{D}^1_{\mathsf{pk}}$**, and** $R_{\mathbf{SWI}}$ **does not send** $(\mathsf{pk}, \mathsf{ct})$ **to** $V^*_{\mathbf{SWI}}$ **along with an accepting proof** π**.** In this case, we can use $R^{V^*_{\mathrm{SWI}}}_{\mathrm{SWI}}$ to break the CCA security of PKE since we can efficiently emulate V^*_{SWI} for R_{SWI} in the CCA-security game (i.e., by using the decryption oracle).
- **Case 3.** $R^{V^*_{\mathrm{SWI}}}_{\mathbf{SWI}}(\mathsf{pk}, \mathsf{ct})$ **distinguishes whether** $(\mathsf{pk}, \mathsf{ct}) \leftarrow \mathcal{D}^0_{\mathsf{pk}}$ **or** $(\mathsf{pk}, \mathsf{ct}) \leftarrow \mathcal{D}^1_{\mathsf{pk}}$**, and** $R_{\mathbf{SWI}}$ **sends** $(\mathsf{pk}, \mathsf{ct})$ **to** $V^*_{\mathbf{SWI}}$ **along with an accepting proof** π**.** In this case, we can use R_{SWI} to break the soundness of (P, V). Indeed, the CCA security of PKE guarantees that even when ct is a false statement (e.g., a random encryption of 2), R_{SWI} still sends $(\mathsf{pk}, \mathsf{ct})$ to V^*_{SWI} along with an accepting proof. Thus, we can straightforwardly design an attacker against the soundness of (P, V) by efficiently emulating V^*_{SWI} for R_{SWI} by using sk for random $(\mathsf{pk}, \mathsf{sk})$.

Note that in the above argument, it is important that the reduction R_{SWI} is oblivious, i.e., is black-box about the distributions. This is because when we rely on the CCA security of PKE, we require that a single reduction works for every pk (rather than that each pk has its owns (not necessarily efficiently constructible) reduction).

2-round strong WI: non-delayed-input or publicly verifiable. Next, we explain the main difficulty that arises when we consider 2-round protocols. In general, when we consider a BB reduction for the strong WI of 2-round interactive arguments, we need to think that the reduction can "rewind" the given verifier V^*, i.e., it can control the randomness of V^* so that it can force V^* to reuse the same verifier message in multiple queries. In this case, the above argument for non-interactive strong WI fails when we try to use the reduction R_{SWI} to break the soundness of (P, V). To see this, note that the soundness attacker first receives a verifier message from the external verifier and needs to forward it to the internally emulated R_{SWI} as an oracle response from V^*_{SWI}. Now, if the reduction R_{SWI} can force V^*_{SWI} to reuse this verifier message in multiple queries (possibly for different statements when we consider the delayed-input setting), we can no longer efficiently emulate V^*_{SWI} for R_{SWI} since we cannot decide whether the reduction R_{SWI} creates an accepting proof or not.

We can easily avoid this difficulty if we consider the standard (non-delayed-input) strong WI and (possibly delayed-input) publicly verifiable strong WI. First, in the case of publicly verifiable strong WI, it is easy to see that the above argument for non-interactive strong WI still works with no modification since we can still emulate V^*_{SWI} for R_{SWI} efficiently even when the same first message is reused. Second, in the case of the standard strong WI, we can effectively prevent the reuse of verifier messages since we can consider a verifier that obtains all the randomness by applying PRF on the statement at the beginning.

Thus, it remains to consider privately verifiable delayed-input strong WI.

2-round strong WI: (possibly privately verifiable) delayed-input. In this case, we cannot obtain a BB impossibility result that is as strong as the one for non-interactive strong WI since there exists a positive result [JKKR17] whose strong WI is proven by a BB reduction based on a falsifiable assumption.[10] We thus consider a weaker form of BB impossibility result by assuming that soundness is also proven by a BB reduction based on a falsifiable assumption.

Our high-level strategy is to show that strong WI implies (a weak form of) weak ZK and then reuse our BB impossibility result about weak ZK. Toward showing that strong WI implies weak ZK, let us fix any verifier V^*_{WZK} and distinguisher D_{WZK} against the weak ZK of (P, V), and consider the following strong WI verifier $V^*_{\mathrm{SWI}} = V^*_{\mathrm{SWI}}[\mathsf{pk}, \mathsf{sk}, V^*_{\mathrm{WZK}}, D_{\mathrm{WZK}}]$ (which can be seen as a generalization of $V^*_{\mathrm{SWI}}[\mathsf{pk}, \mathsf{sk}]$, which we consider in the non-interactive case above).

[10] The soundness is proven based on quasi-polynomially hard assumptions.

Verifier V^*_{SWI}:

1. Invoke V^*_{WZK} and let it interact with the external prover. Let $(\mathsf{pk}', \mathsf{ct})$ denote the statement given from the prover and out_V denote the output of V^*_{WZK}.

2. If $\mathsf{pk} = \mathsf{pk}'$ holds and D_{WZK} is convinced by the external prover (i.e., D_{WZK} outputs 1 on $((\mathsf{pk}', \mathsf{ct}), \mathsf{out}_V)$), return the decryption result $b \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ to the prover. Otherwise, return a random bit.

Note that V^*_{SWI} returns a meaning response only when it receives a proof that convinces D_{WZK}. Now, at a high level, by arguing similarly to the case of non-interactive strong WI (with this new version of V^*_{SWI}), we show that the assumption (C, c) is false unless we can use the reduction $R^{V^*_{\mathrm{SWI}}}_{\mathrm{SWI}}$ either to break the CCA security of PKE or to obtain a weak ZK simulator that convinces D_{WZK}.

Unfortunately, although our strategy is intuitively simple, we need to overcome various problems because of subtle differences from the case of non-interactive strong WI (where we use R_{SWI} to break the soundness of (P, V) rather than to obtain a weak ZK simulator).

1. Unlike the case that we use the reduction R_{SWI} to break the soundness of (P, V) (where it suffices to construct a prover that obtains sk as auxiliary input to emulate V^*_{SWI} for R_{SWI} efficiently), we need to construct a weak ZK simulator that is not given sk and still is able to emulate V^*_{SWI} for R_{SWI}—this is because for our proof of weak ZK BB impossibility to go through, we need to make sure that the simulator cannot distinguish true statements (encryptions of 0 or 1) and false statements (encryptions of 2) so that we can show that the simulator creates an accepting proof for a false statement as mentioned at the end of Section 2.1. To overcome this problem, we assume that the CCA-secure encryption PKE in the definition of the language $\mathbf{L}$ is *puncturable* in the following sense: the CCA security holds even when the adversary is given a *punctured secret key* that can be used to emulate the decryption oracle unless the target ciphertext is queried. (It is easy to see that the classical CCA-secure encryption by Dolev et al. [DDN00] satisfies such a property.) Then, we consider a simulator that takes as auxiliary input a punctured secrete key $\mathsf{sk}_{\{\mathsf{ct}\}}$ that corresponds to the statement $(\mathsf{pk}, \mathsf{ct})$ (i.e., $\mathsf{sk}_{\{\mathsf{ct}\}}$ is a key that can be used to emulate the decryption oracle unless ct is queried). The simulator can now emulate V^*_{SWI} for R_{SWI} efficiently by using $\mathsf{sk}_{\{\mathsf{ct}\}}$ and yet it cannot distinguish true and false statements as required.

2. Unlike the case that we use the reduction R_{SWI} to break the soundness of (P, V) (where it suffices to show that we can use R_{SWI} to create a convincing proof for a single (false) statement), we need to show that we can use R_{SWI} to create a convincing proof (w.r.t. V^*_{WZK} and D_{WZK}) for any (true) statement. This is in general hard to show since R_{SWI} might work only for a non-negligible fraction of the statements (this is because the reduction R_{SWI} is only guaranteed to have non-negligible advantage even when it is combined with a verifier V^* that breaks strong WI with very high advantage). To overcome this problem, we consider a weaker definition of distributional weak ZK where (i) the simulator is given polynomially many statements that are sampled from a distribution over $\mathbf{L}$ and (ii) the simulator is only required to give a simulated proof for one of these statements. Now, by properly defining the distribution, we can show that if the simulator is given sufficiently many statements, with high probability the simulator can find a statement for which the reduction R_{SWI} works, so it can create a convincing proof for one of the statements. Furthermore, our BB impossibility of weak ZK can be easily extended to this distributional weak ZK setting.

3. Unlike the case that we use the reduction R_{SWI} to break the soundness of (P, V) (where it suffices to show that R_{SWI} creates a proof that is convincing with non-negligible probability), we need to show that R_{SWI} creates a proof that is convincing with probability as high as an honest proof. To overcome this problem, we modify V^*_{SWI} in such a way that (i) V^*_{SWI} approximates (by sampling) the probability that an honest prover convinces D_{WZK} for a random statement, and also approximates the probability that the external prover convinces D_{WZK}, and (ii) V^*_{SWI} returns the decryption result $b \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ only when the latter is sufficiently high compared with the former. Now, we can show that R_{SWI} creates a proof that convinces D_{WZK} with probability as high as an honest proof since otherwise R_{SWI} cannot obtain meaningful responses from V^*_{SWI}.

2.3 Outline of the Rest of the Paper

In Section 3, we introduce notations and definitions. From Section 4 to Section 6, we prove the lemmas that are necessary to obtain our results: in Section 4, we show that 2-round delayed-input strong WI with BB reductions implies a weak form of weak ZK; in Section 5, we show that such a weak form of weak ZK implies a weak form of pre-processing (t, ϵ)-ZK; in Section 6, we show a BB impossibility result about such a weak form of pre-processing (t, ϵ)-ZK. In Section 7, we explain how we obtain our main results.

3 Preliminaries

3.1 Notations

We denote the security parameter by n. For any $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, \dots, n\}$. For any random variable X, we use $\mathrm{Supp}(X)$ to denote the support of X. We use poly to denote an unspecified polynomial, use negl to denote an unspecified negligible function, and PPT as an abbreviation of "probabilistic polynomial-time." For any NP language $\mathbf{L}$, we use $\mathbf{R_L}$ to denote its witness relation (i.e., $\mathbf{R_L}$ is the set of all the instance-witness pairs of $\mathbf{L}$). For any pair of (possibly probabilistic) interactive Turing machines (P, V), we use $\langle P(w), V(z)\rangle(x)$ for any $x, w, z \in \{0,1\}^*$ to denote the random variable representing the output of V in an interaction between $P(x, w)$ and $V(x, z)$. Specifically, since we only consider such P and V that participate in a 2-round interaction where V starts the interaction, $\langle P(w), V(z)\rangle(x)$ represents the value out_V that is generated in the following process: $m_1 \leftarrow V(x, z)$; $m_2 \leftarrow P(x, w, m_1)$; $\mathsf{out}_V \leftarrow V(m_2)$.[11]

We assume that readers are familiar with the definitions of computational indistinguishability and basic cryptographic primitives, where unless explicitly stated, we assume that cryptographic primitives are secure against non-uniform adversaries. Following the standard convention, we think that a Turing machine runs in polynomial time if its running time is polynomially bounded in the length of its first input (which is often implicitly the security parameter). For any two sequences of random variables (or distributions) $\mathcal{X} = \{X_i\}_{i\in\mathbb{N}}, \mathcal{Y} = \{Y_i\}_{i\in\mathbb{N}}$, we use $\mathcal{X} \approx \mathcal{Y}$ to denote that $\mathcal{X}$ and $\mathcal{Y}$ are computationally indistinguishable.

3.2 (δ, γ)-Approximation

Definition 1. *For any $p, \delta, \gamma \in [0,1]$, a probabilistic algorithm Algo is said to give a* (δ, γ)-approximation *of p if the output $\tilde{p}$ of Algo satisfies $\Pr[|\tilde{p} - p| \le \delta] \ge 1 - \gamma$.* ◇

It is easy to see (using a Chernoff Bound) that for any $\delta, \gamma \in [0,1]$ and any distribution $\mathcal{D}$ over $\{0,1\}$, a (δ, γ)-approximation of $p := \Pr[b = 1 \mid b \leftarrow \mathcal{D}]$ can be obtained by taking $k := \Theta(\delta^{-2} \log \gamma^{-1})$ samples from $\mathcal{D}$ and computing the relative frequency in which 1 is sampled. (See any standard textbook, e.g., [MU17, Section 4.2.3].)

3.3 2-Round Interactive Argument

3.3.1 Basic Definitions

Let us recall the definitions of interactive arguments [GMR89, BCC88] and their delayed-input version [JKKR17], focusing on 2-round ones.

Definition 2 (Interactive argument)**.** *For any NP language $\mathbf{L}$, a pair of interactive Turing machines (P, V) is called* a 2-round interactive argument for $\mathbf{L}$ *if it satisfies the following.*

- ***Completeness.*** *There exists a negligible function negl such that for every $(x, w) \in \boldsymbol{R_L}$,*

$$\Pr[\langle P(w), V\rangle(x) = 1] \ge 1 - \mathsf{negl}(|x|) \ .$$

- ***Soundness.*** *For every PPT interactive Turing machine P^*, there exists a negligible function negl such that for every $x \in \{0,1\}^* \setminus \boldsymbol{L}$ and $z \in \{0,1\}^*$,*

$$\Pr[\langle P^*(z), V\rangle(x) = 1] \le \mathsf{negl}(|x|) \ .$$

[11] It should be understood that the secret state that is generated in the first invocation of V is implicitly inherited by the second invocation of V.

◇

Definition 3 (Delayed-input interactive argument)**.** *A 2-round interactive argument (P, V) for an NP language $\boldsymbol{L}$ is called* delayed-input *if it satisfies the following.*

- ***Completeness.*** *There exists a negligible function* negl *such that for every $(x, w) \in \boldsymbol{R}_{\boldsymbol{L}}$,*

$$\Pr\left[\mathsf{out} = 1 \;\middle|\; m_1 \leftarrow V(1^{|x|});\ m_2 \leftarrow P(x, w, m_1);\ \mathsf{out} \leftarrow V(x, m_2)\right] \geq 1 - \mathsf{negl}(|x|) \ .$$

- ***Adaptive soundness.*** *For every PPT interactive Turing machine P^*, there exists a negligible function* negl *such that for every $n \in \mathbb{N}$ and $z \in \{0,1\}^*$,*

$$\Pr\left[\mathsf{out} = 1 \wedge x \in \{0,1\}^n \setminus \boldsymbol{L} \;\middle|\; m_1 \leftarrow V(1^n);\ (x, m_2) \leftarrow P^*(1^n, z, m_1);\ \mathsf{out} \leftarrow V(x, m_2)\right] \leq \mathsf{negl}(n) \ .$$

◇

Notation. For a 2-round delayed-input interactive argument (P, V) for an NP language $\mathbf{L}$, an interactive Turing machine V^* is called a *delayed-input verifier* if for any $(x, w) \in \mathbf{R}_{\mathbf{L}}$, it interacts with $P(x, w)$ in behalf of V in the manner defined in the definition of the correctness above (i.e., in the manner that V^* receives x in the last round of the interaction). For a delayed-input verifier V^*, the notation $\langle P(w), V^*(z)\rangle(x)$ is overloaded naturally, i.e., it denotes the value out_V that is generated in the following process: $m_1 \leftarrow V^*(1^{|x|}, z)$; $m_2 \leftarrow P(x, w, m_1)$; $\mathsf{out}_V \leftarrow V^*(x, m_2)$.

3.3.2 Strong Witness Indistinguishability

Next, let us recall the definition of strong witness indistinguishability (strong WI) [Gol01], where we also introduce its straightforward extension to the delayed-input setting. Since we focus on negative results, we give a definition that is slightly weaker than the one given in [Gol01, Definition 4.6.2].

Definition 4 ((delayed-input) strong WI)**.** *An interactive argument (resp., a delayed-input interactive argument) (P, V) for an NP language $\boldsymbol{L}$ is called* strongly witness indistinguishable *(resp.,* delayed-input strongly witness indistinguishable*) if the following holds: for every $\{(\mathcal{X}_n^0, \mathcal{W}_n^0)\}_{n\in\mathbb{N}}, \{(\mathcal{X}_n^1, \mathcal{W}_n^1)\}_{n\in\mathbb{N}}$ and $\{z_n\}_{n\in\mathbb{N}}$ where each $(\mathcal{X}_n^b, \mathcal{W}_n^b)$ is a joint distribution that ranges over $\boldsymbol{R}_{\boldsymbol{L}} \cap (\{0,1\}^n \times \{0,1\}^*)$ and each z_n is a string in $\{0,1\}^*$, if it holds*

$$\{\mathcal{X}_n^0\}_{n\in\mathbb{N}} \approx \{\mathcal{X}_n^1\}_{n\in\mathbb{N}} \ ,$$

then for every PPT verifier (resp. PPT delayed-input verifier) V^* there exists a negligible function negl *such that for every $n \in \mathbb{N}$,*

$$\left| \begin{array}{l} \Pr\left[\langle P(w), V^*(z_n)\rangle(x) = 1 \mid (x, w) \leftarrow (\mathcal{X}_n^0, \mathcal{W}_n^0)\right] \\ -\Pr\left[\langle P(w), V^*(z_n)\rangle(x) = 1 \mid (x, w) \leftarrow (\mathcal{X}_n^1, \mathcal{W}_n^1)\right] \end{array} \right| \leq \mathsf{negl}(n) \ .$$

◇

3.3.3 Delayed-Input Weak Zero-Knowledge

Next, let us recall the definition of weak zero-knowledge (weak ZK) [DNRS03, CLP15], focusing on the delayed-input version of it while considering non-uniform indistinguishability. Since we focus on negative results, we give a weaker, distributional (t, ϵ) version of the definition [CLP15, JKKR17].

Definition 5 (delayed-input distributional weak (t, ϵ)-zero-knowledge)**.** *Let $\boldsymbol{L}$ be an NP language, t be a polynomial, and ϵ be an inverse polynomial. Then, a delayed-input interactive argument (P, V) for $\boldsymbol{L}$ is said to be* delayed-input distributional weak (t, ϵ)-zero-knowledge *if for every sequence of joint distributions $\mathcal{D}_{xw} = \{(\mathcal{X}_n, \mathcal{W}_n)\}_{n\in\mathbb{N}}$ such that each $(\mathcal{X}_n, \mathcal{W}_n)$ ranges over $\boldsymbol{R}_{\boldsymbol{L}} \cap (\{0,1\}^n \times \{0,1\}^*)$, every PPT delayed-input verifier V^*, and every probabilistic t-time distinguisher D, there exists a PPT simulator S and an $n_0 \in \mathbb{N}$ such that for every $n > n_0$, $z_V \in \{0,1\}^*$, and $z_D \in \{0,1\}^*$, it holds*

$$\left|\Pr\left[D(x, z_D, \langle P(w), V^*(z_V)\rangle(x)) = 1 \mid (x, w) \leftarrow (\mathcal{X}_n, \mathcal{W}_n)\right] - \Pr\left[D(x, z_D, S(x, z_V, z_D)) = 1 \mid x \leftarrow \mathcal{X}_n\right]\right| \leq \epsilon(n) \ .$$

◇

3.3.4 Special-Purpose (Weak) Zero-Knowledge

Next, let us introduce two new prover privacy notions for interactive arguments, where one is a weaker version of ZK and the other is a weaker version of weak ZK. We note that these nations should be viewed just as useful tools for our negative results; they are not intended to give any intuitively meaningful security.

First, we introduce special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional pre-processing (t, ϵ)-zero-knowledge, which is roughly speaking weaker than the standard delayed-input (t, ϵ)-ZK in the following sense.

1. The honest prover takes a random statement-witness pair that is sampled from a distribution $\mathcal{D}_{xwz}$. In contrast, the simulator takes N random statement–auxiliary-input pairs $\{x_i, z_{x,i}\}_{i\in[N]}$ that are sampled one by one from $\mathcal{D}_{xwz}$, and it only does the simulation for one of the statements.
2. The simulator works in a pre-processing model where after receiving an auxiliary input z (which is independent of the statements), the simulator is computationally unbounded and computes short trapdoor information before receiving the statements.

The formal definition is given below. For editorial simplicity, we focus on deterministic verifiers below.

Definition 6 (special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional pre-processing (t, ϵ)-zero-knowledge)**.** *Let $\boldsymbol{L}$ be an NP language, N, t be polynomials, ϵ be an inverse polynomial, and $\mathcal{D}_{xwz} = \{(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)\}_{n\in\mathbb{N}}$ be a sequence of joint distributions such that each $(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)$ ranges over $(\boldsymbol{R}_{\boldsymbol{L}} \times \{0,1\}^*) \cap (\{0,1\}^n \times \{0,1\}^* \times \{0,1\}^*)$. Then, a 2-round delayed-input interactive argument (P, V) for $\boldsymbol{L}$ is said to be* special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional pre-processing (t, ϵ)-zero-knowledge *if for every deterministic polynomial-time delayed-input verifier V^*, there exists a simulator $S = (S_{\mathrm{pre}}, S_{\mathrm{main}})$ such that (i) S_{pre} is computationally unbounded and S_{main} is PPT and (ii) for every probabilistic t-time distinguisher D, there exists an $n_0 \in \mathbb{N}$ such that for every $n > n_0$, $z_V \in \{0,1\}^*$, and $z_D \in \{0,1\}^*$, it holds*

$$\left| \begin{array}{l} \Pr\left[D(x, z_D, \langle P(w), V^*(z_V)\rangle(x)) = 1 \mid (x,w) \leftarrow (\mathcal{X}_n, \mathcal{W}_n)\right] \\ -\Pr\left[D(x_{i^*}, z_D, v) = 1 \;\middle|\; \begin{array}{l} \mathsf{st}_S \leftarrow S_{\mathrm{pre}}(1^n, z_V) \\ (x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N_n] \\ (i^*, v) \leftarrow S_{\mathrm{main}}(\{x_i, z_{x,i}\}_{i\in[N_n]}, \mathsf{st}_S) \end{array}\right] \end{array} \right| \leq \epsilon(n) \ ,$$

where $N_n := N(n, 1/\epsilon(n))$. ◇

We note that although the simulator is given some extra information $z_{x,i}$ about each x_i in the above definition, we will only consider the setting where $z_{x,i}$ does not contain much information about a witness for x_i. In particular, the distribution $(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)$ that we will consider has a related distribution $(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)$ over $(\{0,1\}^n \setminus L) \times \{0,1\}^*$ such that $(\mathcal{X}_n, \mathcal{Z}_n)$ and $(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)$ are computationally indistinguishable.

Next, we introduce special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional super-weak (t, ϵ)-zero-knowledge, which is roughly speaking weaker than the standard delayed-input weak (t, ϵ)-ZK in the following sense.

- The definition is $(\mathcal{D}_{xwz}, N)$-distributional in the same sense as above.
- The definition is super-weak ZK [CLP15] in the sense that the simulator is only required to convince the distinguisher with probability as high as an honest prover (i.e., only "one-sided" indistinguishability is required).

The formal definition is given below.

Definition 7 (special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional super-weak (t, ϵ)-zero-knowledge)**.** *Let $\boldsymbol{L}$ be an NP language, N, t be polynomials, ϵ be an inverse polynomial, and $\mathcal{D}_{xwz} = \{(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)\}_{n\in\mathbb{N}}$ be a sequence of joint distributions such that each $(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)$ ranges over $(\boldsymbol{R}_{\boldsymbol{L}} \times \{0,1\}^*) \cap (\{0,1\}^n \times \{0,1\}^* \times \{0,1\}^*)$. Then, a 2-round delayed-input interactive argument (P, V) for $\boldsymbol{L}$ is said to be* special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional super-weak (t, ϵ)-zero-knowledge *if for every deterministic polynomial-time delayed-input verifier V^* and every probabilistic t-time distinguisher D, there exists a PPT simulator S and an $n_0 \in \mathbb{N}$ such that for every $n > n_0$, $z_V \in \{0,1\}^*$, and $z_D \in \{0,1\}^*$, it holds*

$$\begin{aligned} &\Pr\left[D(x_{i^*}, z_D, v) = 1 \;\middle|\; \begin{array}{l} (x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N_n] \\ (i^*, v) \leftarrow S(\{x_i, z_{x,i}\}_{i\in[N_n]}, z_V, z_D) \end{array}\right] \\ &\geq \Pr\left[D(x, z_D, \langle P(w), V^*(z_V)\rangle(x)) = 1 \mid (x,w) \leftarrow (\mathcal{X}_n, \mathcal{W}_n)\right] - \epsilon(n) \ , \end{aligned}$$

where $N_n := N(n, 1/\epsilon(n))$. ◇

Remark 1 (Non-uniform indistinguishability). In both Definition 6 and Definition 7, the indistinguishability between a real proof and simulation holds against non-uniform distinguisher since the distinguisher takes its own auxiliary input z_D (which can contain z_V if necessary). Note that in Definition 7, the simulator also takes z_D since we consider the weak ZK setting. ◇

3.4 Falsifiable Assumption and Black-Box Reduction

3.4.1 Falsifiable Assumption

First, let us recall the definition of falsifiable assumptions from [Nao03, GW11].

Definition 8 (Falsifiable assumption)**.** *A* falsifiable cryptographic assumption *consists of a* PPT *interactive Turing machine* C *and a constant* $c \in [0, 1)$*, where* C *is called the* challenger*. On security parameter* n*, the challenger* $C(1^n)$ *interacts with an interactive Turing machine* $\mathcal{A}(1^n, z)$ *for some* $z \in \{0,1\}^*$ *and* C *outputs a bit* $b \in \{0,1\}$ *at the end of the interaction;* $\mathcal{A}$ *is called the* adversary*, and when* $b = 1$*, it is said that* $\mathcal{A}(1^n, z)$ wins $C(1^n)$*. The assumption associated with the tuple* (C, c) *states that for every* PPT *adversary* $\mathcal{A}$ *there exists a negligible function* negl *such that for every* $n \in \mathbb{N}$ *and* $z \in \{0,1\}^*$*, it holds* $\Pr[\langle \mathcal{A}(z), C\rangle(1^n) = 1] \leq c + \mathsf{negl}(n)$. ◇

For any polynomial p and security parameter n, we say that an (possibly inefficient) adversary $\mathcal{A}$ *breaks* a falsifiable assumption (C, c) on n with *advantage* $1/p(n)$ if there exists $z \in \{0,1\}^*$ such that it holds $\Pr[\langle \mathcal{A}(z), C\rangle(1^n) = 1] \geq c + 1/p(n)$. We say that an (possibly inefficient) adversary $\mathcal{A}$ breaks a falsifiable assumption (C, c) if there exists a polynomial p such that for infinitely many $n \in \mathbb{N}$, $\mathcal{A}$ breaks (C, c) on n with advantage $1/p(n)$.

3.4.2 Black-Box Reduction

Next, we introduce the definitions of black-box (BB) reductions. We consider BB reductions for adaptive soundness and BB reductions for strong WI. The former is defined as in [GW11, Wic13] and the latter is defined similarly to "oblivious" BB reductions for witness hiding [HRS09].

Definition 9 (BB reduction for adaptive soundness)**.** *Let* (P, V) *be a pair of interactive Turing machines that satisfies the correctness of a delayed-input 2-round interactive argument for an NP language* $\boldsymbol{L}$*. Then, a* PPT *oracle Turing machine* R *is said to be a* black-box reduction for showing the adaptive soundness of (P, V) based on a falsifiable assumption (C, c) *if there exists a polynomial* p *such that for every (possibly inefficient) interactive Turing machine* P^* *and every sufficiently large* $n \in \mathbb{N}$*, if there exists* $z \in \{0,1\}^*$ *such that*

$$\Pr\left[\mathsf{out} = 1 \wedge x \in \{0,1\}^n \setminus \boldsymbol{L} \;\middle|\; m_1 \leftarrow V(1^n);\ (x, m_2) \leftarrow P^*(1^n, z, m_1);\ \mathsf{out} \leftarrow V(x, m_2)\right] \geq \frac{1}{2}\ ,$$

then the machine $R^{P^*_z}$ *breaks the assumption* (C, c) *on* n *with advantage* $1/p(n)$ *(where* P^*_z *is the same as* P^* *except that* z *is hardwired as its auxiliary input).* ◇

Definition 10 (Oblivious BB reduction for (delayed-input) strong WI)**.** *Let* (P, V) *be a pair of interactive Turing machines that satisfies the correctness of 2-round interactive argument (resp., delayed-input interactive argument) for an NP language* $\boldsymbol{L}$*. Then, a* PPT *oracle Turing machine* R *is said to be an* oblivious black-box reduction for showing the strong WI (resp., delayed-input strong WI) of (P, V) based on a falsifiable assumption (C, c) *if for every polynomial* p*, there exists a polynomial* p' *such that for every (possibly inefficient) verifier (resp., delayed-input verifier)* V^**, every sufficiently large* $n \in \mathbb{N}$*, every two joint distributions* $\mathcal{D}^0_n = (\mathcal{X}^0_n, \mathcal{W}^0_n), \mathcal{D}^1_n = (\mathcal{X}^1_n, \mathcal{W}^1_n)$ *such that each* $(\mathcal{X}^b_n, \mathcal{W}^b_n)$ *ranges over* $\boldsymbol{R_L} \cap (\{0,1\}^n \times \{0,1\}^*)$*, and every* $z \in \{0,1\}^*$*, if*

$$\Pr\left[\langle P(w), V^*(z)\rangle(x) = b \;\middle|\; b \leftarrow \{0,1\};\ (x, w) \leftarrow (\mathcal{X}^b_n, \mathcal{W}^b_n)\right] \geq \frac{1}{2} + \frac{1}{p(n)}\ ,$$

then either (i) $R^{V^*_z, \mathcal{D}^0_n, \mathcal{D}^1_n}(1^n, 1^{p(n)})$ *breaks the assumption* (C, c) *on* n *with advantage* $1/p'(n)$ *or (ii)* $R^{V^*_z, \mathcal{D}^0_n, \mathcal{D}^1_n}(1^n, 1^{p(n)})$ *distinguishes* $\mathcal{X}^0_n$ *and* $\mathcal{X}^1_n$ *with advantage* $1/p'(n)$*, i.e., it holds*

$$\left|\Pr\left[R^{V^*_z, \mathcal{D}^0_n, \mathcal{D}^1_n}(1^n, 1^{p(n)}, x) = 1 \;\middle|\; x \leftarrow \mathcal{X}^0_n\right] - \Pr\left[R^{V^*_z, \mathcal{D}^0_n, \mathcal{D}^1_n}(1^n, 1^{p(n)}, x) = 1 \;\middle|\; x \leftarrow \mathcal{X}^1_n\right]\right| \geq \frac{1}{p'(n)}\ ,$$

where V^*_z *is the same as* V^* *except that* z *is hardwired as its auxiliary input.* ◇

Remark 2. As is [CLMP12, Wic13], we assume that given security parameter n, BB reductions make queries to the adversary with the same security parameter n. Also, we note that in Definition 9, the reduction R is given access to an adversary P^* that strongly breaks soundness (in the sense that the success probability is $1/2$ rather than non-negligible). Since we consider negative results (which essentially show the nonexistence of BB reductions), focusing on reductions that have access to such an adversary makes our results stronger. ◇

Conventions. Note that in Definition 9 and Definition 10, BB reductions are given access to probabilistic interactive Turing machines. When an oracle machine R is given oracle access to a probabilistic interactive Turing machine $\mathcal{A}$, we follow the following conventions (see, e.g., [BMO90, Gol01]), which are (to the best of our knowledge) general enough to capture the existing BB reductions.

- What R actually makes queries to is the next-message function of $\mathcal{A}$, i.e., a function $\mathcal{A}_r$ for some randomness r such that for any input x and a (possibly empty) list of messages $\vec{m}$, $\mathcal{A}_r(x, \vec{m})$ returns the message that $\mathcal{A}(x; r)$ will send after receiving messages $\vec{m}$ (or it returns the output of $\mathcal{A}$ if the interaction reaches the last round after $\mathcal{A}(x; r)$ receives $\vec{m}$).
- The randomness for $\mathcal{A}$ is set uniformly randomly, and in each query R can choose whether $\mathcal{A}$ should reuse the current randomness or it should use new (uniformly random) randomness.

3.5 Puncturable (CCA-Secure) Public-Key Encryption

Let us first recall the definition of CCA-secure public-key encryption [NY90, RS92].

Definition 11. *A* CCA-secure public-key encryption scheme *(PKE) consists of three* PPT *algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *that satisfy the following.*

- ***Correctness.*** *For every $n \in \mathbb{N}$ and $m \in \{0,1\}^n$,*

$$\Pr\left[\mathsf{Dec}(\mathsf{sk}, c) = m \;\middle|\; (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n);\ c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)\right] = 1 \ .$$

- ***CCA security.*** *For every pair of* PPT *Turing machines $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists a negligible function* negl *such that for every $n \in \mathbb{N}$ and $z \in \{0,1\}^*$,*

$$\Pr\left[b = b' \;\middle|\; \begin{array}{l}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)\\ (m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1^{\mathsf{Dec}(\mathsf{sk},\cdot)}(1^n, \mathsf{pk}, z) \text{ where } |m_0| = |m_1|\\ b \leftarrow \{0,1\};\ c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b);\ b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}'(\mathsf{sk},\cdot)}(\mathsf{st}, c)\end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(n),$$

  where the oracle $\mathsf{Dec}'(\mathsf{sk}, \cdot)$ *is the same as* $\mathsf{Dec}(\mathsf{sk}, \cdot)$ *except that it returns $\perp$ when $\mathcal{A}_2$ queries the challenge ciphertext c to it.*

◇

Next, we introduce a new type of PKE schemes that we call puncturable public-key encryption.[12]

Definition 12. *A public-key encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is called* puncturable *if there exist two* PPT *algorithms* $(\mathsf{PuncGen}, \mathsf{PuncDec})$ *that satisfy the following.*

- ***Correctness of punctured keys.*** *For every pair of* PPT *Turing machines $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the outputs of the following two probabilistic experiments are computationally indistinguishable for every $n \in \mathbb{N}$ and $z \in \{0,1\}^*$.*
  - ***Experiment 1.***
    1. *Run* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$, $(m, \mathsf{st}) \leftarrow \mathcal{A}_1(1^n, \mathsf{pk}, z)$, $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$, $\mathsf{sk}_{\{c\}} \leftarrow \mathsf{PuncGen}(\mathsf{sk}, c)$, *and* $\mathsf{out} \leftarrow \mathcal{A}_2^{\mathsf{Dec}(\mathsf{sk},\cdot)}(\mathsf{st}, c, \mathsf{sk}_{\{c\}})$.
    2. *If $\mathcal{A}_2$ queried c to* Dec *in the previous step, the output of the experiment is $\perp$. Otherwise, the output is* out.

[12] Our definition of puncturable PKE is related to but is much simpler than the one that is proposed in [GM15].

– ***Experiment 2.***

1. *Run* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$, $(m, \mathsf{st}) \leftarrow \mathcal{A}_1(1^n, \mathsf{pk}, z)$, $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m)$, $\mathsf{sk}_{\{c\}} \leftarrow \mathsf{PuncGen}(\mathsf{sk}, c)$, *and* $\mathsf{out} \leftarrow \mathcal{A}_2^{\mathsf{PuncDec}(\mathsf{sk}_{\{c\}}, \cdot)}(\mathsf{st}, c, \mathsf{sk}_{\{c\}})$.
2. *If* $\mathcal{A}_2$ *queried* c *to* PuncDec *in the previous step, the output of the experiment is* $\bot$. *Otherwise, the output is* out.

- ***Security of punctured keys.*** *For every pair of* PPT *Turing machines* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, *there exists a negligible function* negl *such that for every* $n \in \mathbb{N}$ *and* $z \in \{0,1\}^*$,

$$\Pr\left[\mathcal{A}_2(\mathsf{st}, c, \mathsf{sk}_{\{c\}}) = b \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n) \\ (m_0, m_1, \mathsf{st}) \leftarrow \mathcal{A}_1(1^n, \mathsf{pk}, z), \text{ where } |m_0| = |m_1| \\ b \leftarrow \{0,1\};\ c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b);\ \mathsf{sk}_{\{c\}} \leftarrow \mathsf{PuncGen}(\mathsf{sk}, c) \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(n) \ .$$

◇

It is easy to verify that the classical CCA-secure PKE of Dolev et al. [DDN00] is puncturable. (Indeed, their proof of CCA security relies on the very fact that we can create a key with which we can emulate the decryption oracle without disturbing the security of the challenge ciphertext; see Appendix A.) Thus, we have the following lemma.

Lemma 1. *Assume the existence of trapdoor permutations. Then, there exists a puncturable CCA-secure public-key encryption scheme.*

4 From 2-Round Delayed-Input Strong WI to 2-Round Special-Purpose Weak ZK

In this section, we show that 2-round delayed-input strong WI arguments satisfy a weak form of delayed-input weak ZK if their strong WI is proven by oblivious BB reductions.

Lemma 2. *Assume the existence of puncturable CCA-secure public-key encryption schemes. Then, there exists an NP language* $\mathbf{L}$ *such that if there exist*

- *a 2-round delayed-input interactive argument* (P, V) *for* $\mathbf{L}$ *and*
- *an oblivious black-box reduction* R_{swi} *for showing the delayed-input strong WI of* (P, V) *based on a falsifiable assumption* (C, c),

then either (i) the assumption (C, c) *is false or (ii)* (P, V) *is special-purpose delayed-input* $(\mathcal{D}_{xwz}, N)$*-distributional super-weak* (t, ϵ)*-zero-knowledge for every polynomial* t *and every inverse polynomial* ϵ, *where* N *is a polynomial and* $\mathcal{D}_{xwz} = \{(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)\}_{n\in\mathbb{N}}$ *is a sequence of efficient joint distributions such that each* $(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)$ *ranges over* $(\boldsymbol{R}_{\boldsymbol{L}} \times \{0,1\}^*) \cap (\{0,1\}^n \times \{0,1\}^* \times \{0,1\}^*)$. *Furthermore, there exists a sequence of joint distributions* $\overline{\mathcal{D}}_{xz} = \{(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)\}_{n\in\mathbb{N}}$ *such that each* $(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)$ *ranges over* $(\{0,1\}^n \setminus L) \times \{0,1\}^*$ *and* $\overline{\mathcal{D}}_{xz}$ *is computationally indistinguishable from* $\mathcal{D}_{xz} := \{(\mathcal{X}_n, \mathcal{Z}_n)\}_{n\in\mathbb{N}}$.

We note that the proof of this lemma is a little involved, and a related proof that is simpler than this one can be found in Appendix B (where we prove a BB impossibility of 2-round non-delayed-input strong WI).

Proof. Let $\mathsf{PuncPKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{PuncGen}, \mathsf{PuncDec})$ be a puncturable CCA-secure PKE and $\mathbf{L}$ be the NP language that consists of all the public-key–ciphertext pairs of PuncPKE such that either 0 or 1 is encrypted (the public key is not necessarily honestly generated), i.e.,

$$\mathbf{L} := \left\{(\mathsf{pk}, \mathsf{ct}) \;\middle|\; \exists b \in \{0,1\}, r \in \{0,1\}^{\mathsf{poly}(n)} \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, b; r)\right\} \ .$$

Assume, as stated in the statement of the lemma, the existence of a 2-round delayed-input interactive argument (P, V) and an oblivious black-box reduction R_{swi} for showing the delayed-input strong WI of (P, V) based on a falsifiable assumption (C, c). For any inverse polynomial ϵ', let $Q_{\epsilon'}$ denote a polynomial such that for every

delayed-input verifier V^*, every $n \in \mathbb{N}$, every two joint distributions $\mathcal{D}^0_n = (\mathcal{X}^0_n, \mathcal{W}^0_n)$ and $\mathcal{D}^1_n = (\mathcal{X}^1_n, \mathcal{W}^1_n)$ over $\mathbf{R_L} \cap (\{0,1\}^n \times \{0,1\}^*)$, and every $z \in \{0,1\}^*$, if it holds

$$\Pr\left[\langle P(w), V^*(z)\rangle(x) = b \;\middle|\; b \leftarrow \{0,1\};\ (x,w) \leftarrow (\mathcal{X}^b_n, \mathcal{W}^b_n)\ \right] \geq \frac{1}{2} + \epsilon'(n)\ ,$$

then either (i) $R_{\mathrm{SWI}}^{V^*_z, \mathcal{D}^0_n, \mathcal{D}^1_n}(1^n, 1^{1/\epsilon'(n)})$ breaks the assumption (C, c) on n with advantage $1/Q_{\epsilon'}(n)$ or (ii) $R_{\mathrm{SWI}}^{V^*_z, \mathcal{D}^0_n, \mathcal{D}^1_n}(1^n, 1^{1/\epsilon'(n)})$ distinguishes $\mathcal{X}^0_n$ and $\mathcal{X}^1_n$ with advantage $1/Q_{\epsilon'}(n)$. (Such a polynomial is guaranteed to exist because of our assumption on R_{SWI}.) Fix any polynomial t and inverse polynomial ϵ. Our goal is to show that either the assumption (C, c) is false or (P, V) is special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional super-weak (t, ϵ)-zero-knowledge for a distribution $\mathcal{D}_{xwz}$ and a polynomial N.

At a high level, the proof proceeds as outlined in Section 2.2. Specifically, for any verifier and distinguisher against the weak ZK of (P, V), we first define a cheating verifier V^*_{SWI} against the strong WI of (P, V). Then, we proceed with case analysis about the behavior of $R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}}$, where in the first case, we show that we can efficiently break the assumption (C, c) by using R_{SWI}, and in the second case, we show that we can obtain a simulator for weak ZK by using R_{SWI}. We note that in what follows, we use several constants that are chosen rather arbitrarily so that the proof works.

We first introduce distributions over $\mathbf{R_L}$ and a delayed-input verifier against the strong WI of (P, V). For any $n \in \mathbb{N}$, let Keys_n be the set of all the keys that can be output by $\mathsf{Gen}(1^n)$, i.e., $\mathsf{Keys}_n := \{(\mathsf{pk}, \mathsf{sk}) \mid \exists r \in \{0,1\}^* \text{s.t. } (\mathsf{pk}, \mathsf{sk}) = \mathsf{Gen}(1^n; r)\}$. Then, for any $n \in \mathbb{N}$ and any $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Keys}_n$, let $\mathcal{D}^0_{\mathsf{pk}}$ and $\mathcal{D}^1_{\mathsf{pk}}$ be the distributions that are defined over $\mathbf{R_L}$ as follows: $\forall b \in \{0,1\}$,

$$\mathcal{D}^b_{\mathsf{pk}} := \left\{((\mathsf{pk}, \mathsf{ct}), (b, r)) \;\middle|\; r \leftarrow \{0,1\}^{\mathsf{poly}(n)};\ \mathsf{ct} := \mathsf{Enc}(\mathsf{pk}, b; r)\ \right\}\ ,$$

i.e., the first part of $\mathcal{D}^b_{\mathsf{pk}}$ outputs pk and a random encryption of b, and the second part outputs b and the randomness of the encryption. We use $(\mathcal{X}^b_{\mathsf{pk}}, \mathcal{W}^b_{\mathsf{pk}})$ to denote the joint distributions such that $\mathcal{X}^b_{\mathsf{pk}}$ denotes the first part of $\mathcal{D}^b_{\mathsf{pk}}$ and $\mathcal{W}^b_{\mathsf{pk}}$ denotes the second part of $\mathcal{D}^b_{\mathsf{pk}}$. Next, for any $n \in \mathbb{N}$, any $z = z_V \| z_D \in \{0,1\}^*$, any $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Keys}_n$, and any pair of a (deterministic) delayed-input verifier V^*_{WZK} and a (probabilistic) distinguisher D_{WZK} against the weak zero-knowledge property of (P, V), let $V^*_{\mathrm{SWI}}[n, z, \mathsf{pk}, \mathsf{sk}, V^*_{\mathrm{WZK}}, D_{\mathrm{WZK}}]$ be the delayed-input verifier described in Algorithm 1. Note that due to the correctness of $\mathsf{PuncPKE}$, our verifier $V^*_{\mathrm{SWI}}[n, z, \mathsf{pk}, \mathsf{sk}, V^*_{\mathrm{WZK}}, D_{\mathrm{WZK}}]$ distinguishes $\mathcal{D}^0_{\mathsf{pk}}$ and $\mathcal{D}^1_{\mathsf{pk}}$ with probability 1 when it interacts with a prover that passes the test in the last step of $V^*_{\mathrm{SWI}}[n, z, \mathsf{pk}, \mathsf{sk}, V^*_{\mathrm{WZK}}, D_{\mathrm{WZK}}]$. In the following, we usually write $V^*_{\mathrm{SWI}}[n, z, \mathsf{pk}, \mathsf{sk}, V^*_{\mathrm{WZK}}, D_{\mathrm{WZK}}]$ as V^*_{SWI} for editorial simplicity.

We proceed with case analysis about the behavior of the strong WI reduction R_{SWI} in the setting where R_{SWI} is combined with our strong WI verifier V^*_{SWI}. Specifically, we consider the following two cases.

- **Case 1.** There exist a deterministic polynomial-time delayed-input verifier V^*_{WZK}, a probabilistic t-time distinguisher D_{WZK}, and polynomials p_1, p_2 such that for infinitely many $n \in \mathbb{N}$, there exist $z \in \{0,1\}^*$ and $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Keys}_n$ such that $R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}}(1^n, 1^{p_1(n)})$ breaks the assumption (C, c) on n with advantage $1/p_2(n)$, i.e.,

$$\Pr\left[\langle R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}(1^{p_1(n)}), C\rangle(1^n) = 1\right] \geq c + \frac{1}{p_2(n)}\ . \tag{1}$$

- **Case 2.** The condition of Case 1 does not hold.

We analyze each case below.

Analysis of Case 1. We show that R_{SWI} can be used to break the assumption (C, c). Fix any V^*_{WZK}, D_{WZK}, p_1, p_2, n, z, and $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Keys}_n$ such that we have (1). Consider the following adversary $\mathcal{A}$ against (C, c).

1. Given V^*_{WZK}, D_{WZK}, z, and $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Keys}_n$ as auxiliary inputs, $\mathcal{A}$ lets $R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}(1^n, 1^{p_1(n)})$ interact with the challenger C, where sk is used to emulate V^*_{SWI} for R_{SWI} efficiently.

Clearly, $\mathcal{A}$ runs in polynomial time. Also, from (1) it follows immediately that $\mathcal{A}$ breaks the assumption (C, c) on n with advantage $1/p_2(n)$. We thus conclude that the assumption (C, c) is false in this case.

Algorithm 1 Delayed-input strong WI verifier $V^*_{\mathrm{sWI}}[n, z, \mathsf{pk}, \mathsf{sk}, V^*_{\mathrm{WZK}}, D_{\mathrm{WZK}}]$, where $z = z_V \| z_D$.

1. On input 1^n, invoke $V^*_{\mathrm{WZK}}(1^n, z_V)$ and let it interact with the external prover. Let $x^\star = (\mathsf{pk}^\star, \mathsf{ct}^\star)$ denote the statement that is obtained in the last round of the interaction and $\mathsf{out}^\star$ denote the output of V^*_{WZK}. If $\mathsf{pk}^\star \neq \mathsf{pk}$, output a random bit and abort.

2. Sample a key key for a pseudorandom function PRF. In the following, whenever new randomness is required, it is obtained by applying $\mathsf{PRF}(\mathsf{key}, \cdot)$ on the transcript that is exchanged with the prover in the previous step. (The previous step does not require randomness since V^*_{WZK} is assumed to be deterministic.)

3. **(Approximation of honest prover's success probability.)** Obtain a $(\epsilon(n)/16, \mathsf{negl}(n))$-approximation $\tilde{p}$ of

$$p := \Pr\left[D_{\mathrm{WZK}}(x, z_D, \langle P(w), V^*_{\mathrm{WZK}}(z_V)\rangle(x)) = 1 \;\middle|\; (\mathsf{pk}', \mathsf{sk}') \leftarrow \mathsf{Gen}(1^n);\ b \leftarrow \{0,1\};\ (x, w) \leftarrow \mathcal{D}^b_{\mathsf{pk}'}\right] .$$

4. **(Approximation of external prover's success probability.)** Obtain a $(\epsilon(n)/16, \mathsf{negl}(n))$-approximation $\tilde{p}^\star$ of

$$p^\star := \Pr\left[D_{\mathrm{WZK}}(x^\star, z_D, \mathsf{out}^\star) = 1\right] .$$

5. Output a random bit and abort if $\tilde{p}^\star < \tilde{p} - \epsilon(n)/2$ (which suggests that the external prover with the given statement $x^\star$ is not likely to convince D_{WZK} with probability as high as an honest prover with a random statement). Otherwise, run $b \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct}^\star)$ and output b.

Analysis of Case 2. We show that R_{sWI} can be used to construct a simulator for the special-purpose distributional super-weak (ϵ, t)-zero-knowledge property of (P, V) (unless it can be used to break the CCA security of $\mathsf{PuncPKE}$). Toward this end, we split Case 2 into two sub-cases based on the behavior of R_{sWI} in the setting where $R^{V^*_{\mathrm{sWI}}}_{\mathrm{sWI}}$ is used as a distinguisher against $\mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}$ for randomly chosen $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$. Let us first introduce the following notations about $(\mathsf{pk}, \mathsf{sk})$ of $\mathsf{PuncPKE}$. For any n, z, $(\mathsf{pk}, \mathsf{sk})$, V^*_{WZK}, and D_{WZK}:

- $(\mathsf{pk}, \mathsf{sk})$ is called *interesting (w.r.t. $(n, z, V^*_{WZK}, D_{WZK})$)* if it satisfies the following.

$$\Pr\left[\langle P(w), V^*_{\mathrm{sWI}}\rangle(x) = b \;\middle|\; b \leftarrow \{0,1\};\ (x, w) \leftarrow \mathcal{D}^b_{\mathsf{pk}}\right] \geq \frac{1}{2} + \frac{\epsilon(n)}{18} . \tag{2}$$

  Intuitively, $(\mathsf{pk}, \mathsf{sk})$ is interesting if $V^*_{\mathrm{sWI}}[n, z, \mathsf{pk}, \mathsf{sk}, V^*_{\mathrm{WZK}}, D_{\mathrm{WZK}}]$ breaks the strong WI of (P, V) w.r.t. $\mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}$ with high advantage (which implies that R_{sWI} either breaks (C, c) or distinguishes $\mathcal{X}^0_{\mathsf{pk}}$ and $\mathcal{X}^1_{\mathsf{pk}}$ given V^*_{sWI}).

- $(\mathsf{pk}, \mathsf{sk})$ is called *type-1 interesting* if it is interesting and in addition satisfies the following.

$$\Pr\left[\mathsf{INTERESTING\text{-}QUERY} \;\middle|\; \begin{array}{l} b \leftarrow \{0,1\};\ (x, w) \leftarrow \mathcal{D}^b_{\mathsf{pk}} \\ b' \leftarrow R^{V^*_{\mathrm{sWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}_{\mathrm{sWI}}(1^n, 1^{36/\epsilon(n)}, x) \end{array}\right] \leq \frac{1}{4Q_{\epsilon/36}(n)} ,$$

  where (i) $Q_{\epsilon/36}$ is the polynomial that is introduced at the beginning of the proof and (ii) $\mathsf{INTERESTING\text{-}QUERY}$ is the event that is defined as follows: through oracle queries to V^*_{sWI}, the reduction $R_{\mathrm{sWI}}(1^n, 1^{36/\epsilon(n)}, x)$ invokes an execution of (P, V) in which R_{sWI} forwards the statement x to V^*_{sWI} along with an accepting prover message (i.e., a message that passes the test in the last step of V^*_{sWI}). Note that by the construction of V^*_{sWI}, $\mathsf{INTERESTING\text{-}QUERY}$ implies that R_{sWI} produces a prover message that convinces D_{WZK} with high probability on the statement x—thus, intuitively, $(\mathsf{pk}, \mathsf{sk})$ is type-1 interesting if R_{sWI} can either break (C, c) or distinguish $\mathcal{X}^0_{\mathsf{pk}}$ and $\mathcal{X}^1_{\mathsf{pk}}$ without producing such a prover message.

- $(\mathsf{pk}, \mathsf{sk})$ is called *type-2 interesting* if it is interesting but is not type-1 interesting.

Now, we consider the following two sub-cases.

- **Case 2-1.** There exist a deterministic polynomial-time delayed-input verifier V^*_{WZK} and a probabilistic t-time distinguisher D_{WZK} such that for infinitely many $n \in \mathbb{N}$ there exists $z \in \{0,1\}^*$ such that

$$\Pr\left[(\mathsf{pk},\mathsf{sk}) \text{ is type-1 interesting} \mid (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)\right] \geq \frac{\epsilon(n)}{8} \ . \tag{3}$$

- **Case 2-2.** The condition of Case 1 does not hold.

We analyze each sub-case below.

Analysis of Case 2-1. We show that R_{SWI} can be used to break the CCA security of $\mathsf{PuncPKE}$. Fix any V^*_{WZK}, D_{WZK}, n, and z such that (i) we have (3) and (ii) for every $(\mathsf{pk},\mathsf{sk}) \in \mathsf{Keys}_n$, we we have

$$\Pr\left[\langle R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}},\mathcal{D}^0_{\mathsf{pk}},\mathcal{D}^1_{\mathsf{pk}}}(1^{36/\epsilon(n)}), C\rangle(1^n) = 1\right] < c + \frac{1}{Q_{\epsilon/36}(n)} \ . \tag{4}$$

(Such V^*_{WZK}, D_{WZK}, n, and z are guaranteed to exist since in Case 2, it is assumed that the condition of Case 1 does not hold.) Then, consider the following adversary $\mathcal{A}_{\mathrm{CCA}}$ against the CCA security of $\mathsf{PuncPKE}$.

1. On input $(1^n, \mathsf{pk}, z)$, the adversary $\mathcal{A}_{\mathrm{CCA}}$ sends $m_0 := 0$ and $m_1 := 1$ to the challenger as the challenge plaintexts.
2. On receiving the challenge ciphertext ct, the adversary $\mathcal{A}_{\mathrm{CCA}}$ first does the following to check whether or not the key pair $(\mathsf{pk},\mathsf{sk})$ that the challenger has is likely to be type-1 interesting.
   (a) Obtain a $(1/4Q_{\epsilon/36}(n), \mathsf{negl}(n))$-approximation $\tilde{p}_1$ of
   $$p_1 := \Pr\left[\langle P(w), V^*_{\mathrm{SWI}}\rangle(x) = b \;\middle|\; b \leftarrow \{0,1\};\ (x,w) \leftarrow \mathcal{D}^b_{\mathsf{pk}}\right] \ ,$$
   where during the approximation, the decryption oracle $\mathsf{Dec}(\mathsf{sk},\cdot)$ is used to emulate V^*_{SWI} efficiently without knowing sk. (Since the definition of p_1 is independent of ct, the probability that ct need to be queried to $\mathsf{Dec}(\mathsf{sk},\cdot)$ is negligible.)
   (b) Obtain a $(1/4Q_{\epsilon/36}(n), \mathsf{negl}(n))$-approximation $\tilde{p}_2$ of
   $$p_2 := \Pr\left[\mathsf{INTERESTING\text{-}QUERY} \;\middle|\; \begin{array}{l} b \leftarrow \{0,1\};\ (x,w) \leftarrow \mathcal{D}^b_{\mathsf{pk}} \\ b' \leftarrow R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}},\mathcal{D}^0_{\mathsf{pk}},\mathcal{D}^1_{\mathsf{pk}}}(1^n, 1^{36/\epsilon(n)}, x) \end{array}\right] \ ,$$
   where as above the decryption oracle $\mathsf{Dec}(\mathsf{sk},\cdot)$ is used to emulate V^*_{SWI} during the approximation.
   (c) If $\tilde{p}_1 < 1/2 + \epsilon(n)/18 - 1/4Q_{\epsilon/36}(n)$ or $\tilde{p}_2 > 1/2Q_{\epsilon/36}(n)$ (which suggests that $(\mathsf{pk},\mathsf{sk})$ is unlikely to be type-1 interesting), output a random bit and abort.
3. Finally, the adversary $\mathcal{A}_{\mathrm{CCA}}$ lets $x^\star := (\mathsf{pk},\mathsf{ct})$ and runs $b^\star \leftarrow R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}},\mathcal{D}^0_{\mathsf{pk}},\mathcal{D}^1_{\mathsf{pk}}}(1^n, 1^{36/\epsilon(n)}, x^\star)$, where as above the decryption oracle $\mathsf{Dec}(\mathsf{sk},\cdot)$ is used to emulate V^*_{SWI}. If $\mathsf{INTERESTING\text{-}QUERY}$ occurs during the execution of R_{SWI}, the adversary $\mathcal{A}_{\mathrm{CCA}}$ outputs a random bit. Otherwise, it outputs $b^\star$.

We now analyze $\mathcal{A}_{\mathrm{CCA}}$. Let ABORT be the event that $\mathcal{A}_{\mathrm{CCA}}$ aborts, and $\mathsf{APPROX\text{-}FAIL}$ be the event that the approximation of any of $\tilde{p}_1, \tilde{p}_2$ fails, i.e., $\max(|p_1 - \tilde{p}_1|, |p_2 - \tilde{p}_2|) > 1/4Q_{\epsilon/36}(n)$. From the union bound, we have $\Pr[\mathsf{APPROX\text{-}FAIL}] \leq \mathsf{negl}(n)$. Also, we have $\Pr[\neg\mathsf{ABORT}] \geq \epsilon(n)/8 - \mathsf{negl}(n)$ due to (3) since $\mathcal{A}_{\mathrm{CCA}}$ does not abort when pk is the public key of a type-1 interesting $(\mathsf{pk},\mathsf{sk})$ and $\mathsf{APPROX\text{-}FAIL}$ does not occur. Now, under the condition that neither $\mathsf{APPROX\text{-}FAIL}$ nor ABORT occurs, we have

$$p_1 \geq \tilde{p}_1 - \frac{1}{4Q_{\epsilon/36}(n)} \geq \frac{1}{2} + \frac{\epsilon(n)}{18} - \frac{1}{2Q_{\epsilon/36}(n)} \geq \frac{1}{2} + \frac{\epsilon(n)}{36} \ , \text{ and} \tag{5}$$

$$p_2 \leq \tilde{p}_2 + \frac{1}{4Q_{\epsilon/36}(n)} \leq \frac{3}{4Q_{\epsilon/36}(n)} \ , \tag{6}$$

where the last inequality in (5) follows since we can assume without loss of generality that $Q_{\epsilon/36}(n)$ is sufficiently large and satisfies $1/Q_{\epsilon/36}(n) \leq \epsilon(n)/18$. Note that when we have (5) and (4) (where the former means that V^*_{SWI}

breaks the strong WI of (P, V) w.r.t. $\mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}$ with advantage $\epsilon(n)/36$ while the latter means that $R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}$ does not break (C, c) with advantage $1/Q_{\epsilon/36}(n)$), it is guaranteed that $R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}$ distinguishes $\mathcal{X}^0_{\mathsf{pk}}$ and $\mathcal{X}^1_{\mathsf{pk}}$ with advantage $1/Q_{\epsilon/36}(n)$ due to the definition of $Q_{\epsilon/36}$. Thus, by additionally using (6) and recalling the definitions of $\mathcal{X}^0_{\mathsf{pk}}$ and $\mathcal{X}^1_{\mathsf{pk}}$ (i.e., that $\mathcal{X}^b_{\mathsf{pk}}$ outputs pk and a random encryption of b), we conclude that $\mathcal{A}_{\mathrm{CCA}}$ wins with advantage at least

$$\left(\frac{1}{Q_{\epsilon/36}(n)} - \Pr\left[\textsf{INTERESTING-QUERY} \text{ occurs in Step 3 of } \mathcal{A}_{\mathrm{CCA}}\right]\right) \times \Pr\left[\neg\textsf{ABORT}\right] - \Pr\left[\textsf{APPROX-FAIL}\right]$$
$$\geq \frac{1}{4Q_{\epsilon/36}(n)} \times \left(\frac{\epsilon(n)}{8} - \mathsf{negl}(n)\right) - \mathsf{negl}(n) = \frac{1}{\mathsf{poly}(n)} \quad .$$

Since this is a contradiction, we conclude that we never have Case 2-1.

Analysis of Case 2-2. We show that R_{SWI} can be used to construct a simulator for the special-purpose distributional super-weak (ϵ, t)-zero-knowledge property of (P, V). For each $n \in \mathbb{N}$, let $(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)$ be the following joint distributions.

$$(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n) := \left\{ ((\mathsf{pk}, \mathsf{ct}), (b, r), \mathsf{sk}_{\{\mathsf{ct}\}}) \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n);\ b \leftarrow \{0,1\};\ r \leftarrow \{0,1\}^{\mathsf{poly}(n)} \\ \mathsf{ct} := \mathsf{Enc}(\mathsf{pk}, b; r);\ \mathsf{sk}_{\{\mathsf{ct}\}} \leftarrow \mathsf{PuncGen}(\mathsf{sk}, \mathsf{ct}) \end{array} \right\} .$$

(Note that $(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)$ indeed ranges over $(\mathbf{R}_{\mathbf{L}} \times \{0,1\}^*) \cap (\{0,1\}^n \times \{0,1\}^* \times \{0,1\}^*)$ as required.[13] Also, note that $(\mathcal{X}_n, \mathcal{W}_n)$ is identically distributed with $\{(x, w) \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n); b \leftarrow \{0,1\}; (x, w) \leftarrow \mathcal{D}^b_{\mathsf{pk}}\}$.) Let N be the polynomial such that $N(n, 1/\epsilon(n)) := 320Q_{\epsilon/36}(n)/\epsilon(n)^2$.

For any deterministic polynomial-time delayed-input verifier V^*_{WZK} and a probabilistic t-time distinguisher D_{WZK}, we consider the simulator S described in Algorithm 2.

Algorithm 2 Weak zero-knowledge simulator S.

Input: $\{x_i, z_{x,i}\}_{i \in [N_n]}$ and $z_V, z_D \in \{0,1\}^*$, where $N_n := N(n, 1/\epsilon(n))$ and each $(x_i, z_{x,i})$ is sampled from $(\mathcal{X}_n, \mathcal{Z}_n)$.
Hardwired information: the descriptions of the verifier V^*_{WZK} and the distinguisher D_{WZK}.

1. Let $z := z_V \| z_D$. Then, for each $i \in [N_n]$, do the following.
   (a) Parse $(x_i, z_{x,i})$ as $((\mathsf{pk}, \mathsf{ct}), \mathsf{sk}_{\{\mathsf{ct}\}})$, and run $R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}(1^n, 1^{36/\epsilon(n)}, x_i)$ as a distinguisher for $\mathcal{D}^0_{\mathsf{pk}}$ and $\mathcal{D}^1_{\mathsf{pk}}$ to see whether $\textsf{INTERESTING-QUERY}$ occurs, where the punctured secret key $\mathsf{sk}_{\{\mathsf{ct}\}}$ is used to emulate V^*_{SWI} efficiently for R_{SWI} until $\textsf{INTERESTING-QUERY}$ occurs. (Recall that $\textsf{INTERESTING-QUERY}$ occurs if R_{SWI} makes a query (to V^*_{SWI}) that contains x_i and an accepting prover message.)
   (b) If $\textsf{INTERESTING-QUERY}$ occurs, let $i^\star := i$, and let $\mathsf{out}^\star$ denote the output of V^*_{WZK} that is computed inside V^*_{SWI} when the query that causes $\textsf{INTERESTING-QUERY}$ is made; then, exit the loop and go to the next step.
2. If $(i^\star, \mathsf{out}^\star)$ is not defined in the above step, abort. Otherwise, output $(i^\star, \mathsf{out}^\star)$.

We now proceed with the analysis of S. Fix any V^*_{WZK} and D_{WZK}. Since it is assumed that the condition of Case 2-1 does not hold, we have that for every sufficiently large $n \in \mathbb{N}$ and every $z = z_V \| z_D \in \{0,1\}^*$,

$$\Pr\left[(\mathsf{pk}, \mathsf{sk}) \text{ is type-1 interesting} \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)\right] < \frac{\epsilon(n)}{8} \quad . \tag{7}$$

Fix any such n and $z = z_V \| z_D$. Let p be defined by

$$p := \Pr\left[D_{\mathrm{WZK}}(x, z_D, \langle P(w), V^*_{\mathrm{WZK}}(z_V)\rangle(x)) = 1 \;\middle|\; (x, w) \leftarrow (\mathcal{X}_n, \mathcal{W}_n)\right] \quad . \tag{8}$$

(Note that p is identically defined with the one in the description of V^*_{SWI} in Algorithm 1.)

[13] We assume without loss of generality that on security parameter 1^n, Gen and Enc generate $(\mathsf{pk}, \mathsf{ct})$ such that $|(\mathsf{pk}, \mathsf{ct})| = n$.

We first make a simplifying assumption. First, note that S runs the reduction R_{SWI} with our (probabilistic) verifier V^*_{SWI}. Following the conventions stated in Section 3.4.2, in general the reduction R_{SWI} can make V^*_{SWI} reuse the same randomness multiple times when it makes queries to V^*_{SWI}. However, since V^*_{SWI} obtains randomness by applying PRF on the transcript exchanged with the prover (where the prover message is actually given by R_{SWI}), we can safely think, by assuming without loss of generality that R_{SWI} never makes the same query twice to V^*_{SWI} while making V^*_{SWI} reuse the same randomness, as if V^*_{SWI} always uses new true randomness in each invocation during the execution of S.[14] Second, note that S uses the punctured secret key $\mathsf{sk}_{\{\mathsf{ct}\}}$ to emulate V^*_{SWI} for R_{SWI}. We can however safely think as if S uses the real secret key sk to perfectly emulate V^*_{SWI} since the correctness of punctured keys of $\mathsf{PuncPKE}$ guarantees that the output of R_{SWI} (and hence that of S) is indistinguishable in these two cases. (Note that by the definition of $\mathsf{INTERESTING\text{-}QUERY}$, decrypting ct is not required for the emulation of V^*_{SWI} unless $\mathsf{INTERESTING\text{-}QUERY}$ occurs.)

Next, we bound the probability that S aborts. Toward this end, it suffices to show that we have

$$\Pr\left[(\mathsf{pk},\mathsf{sk}) \text{ is type-2 interesting} \mid (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)\right] \geq \frac{\epsilon(n)}{8} \quad . \tag{9}$$

Indeed, by combining (9) with the definition of type-2 interesting keys, we obtain

$$\Pr\left[\mathsf{INTERESTING\text{-}QUERY} \;\middle|\; \begin{array}{l} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^n) \\ b \leftarrow \{0,1\};\ (x,w) \leftarrow \mathcal{D}^b_{\mathsf{pk}} \\ b' \leftarrow R^{V^*_{\mathrm{SWI}},\mathcal{D}^0_{\mathsf{pk}},\mathcal{D}^1_{\mathsf{pk}}}_{\mathrm{SWI}}(1^n, 1^{36/\epsilon(n)}, x) \end{array}\right] \geq \frac{\epsilon(n)}{8} \cdot \frac{1}{4Q_{\epsilon/36}(n)} = \frac{\epsilon(n)}{32Q_{\epsilon/36}(n)} \quad ,$$

and thus, by using Markov's inequality, we can bound the probability that S aborts as follows.

$$\Pr\left[S(\{x_i, z_{x,i}\}_{i\in[N_n]}, z_V, z_D) \text{ aborts} \mid (x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N_n]\right] \leq \frac{1}{N_n} \cdot \frac{32Q_{\epsilon/36}(n)}{\epsilon(n)} = \frac{\epsilon(n)}{10} \quad . \tag{10}$$

So, we focus on showing (9). Observe that from (8) and an average argument, it follows that with probability at least $\epsilon(n)/4$ over the choice of $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$, we obtain $(\mathsf{pk},\mathsf{sk})$ such that

$$\Pr\left[D_{\mathrm{WZK}}(x, z_D, \langle P(w), V^*_{\mathrm{WZK}}(z_V)\rangle(x)) = 1 \;\middle|\; b \leftarrow \{0,1\};\ (x,w) \leftarrow \mathcal{D}^b_{\mathsf{pk}}\right] \geq p - \frac{\epsilon(n)}{4} \quad . \tag{11}$$

For any such $(\mathsf{pk},\mathsf{sk})$, it follows from (11) and an average argument that with probability at least $\epsilon(n)/8$ over the choice of $b \leftarrow \{0,1\}$, $(x,w) \leftarrow \mathcal{D}^b_{\mathsf{pk}}$, and $\mathsf{out} \leftarrow \langle P(w), V^*_{\mathrm{WZK}}(z_V)\rangle(x)$, we obtain out such that

$$\Pr\left[D_{\mathrm{WZK}}(x, z_D, \mathsf{out}) = 1\right] \geq p - \frac{3\epsilon(n)}{8} \quad . \tag{12}$$

Now, for any $(\mathsf{pk},\mathsf{sk})$ such that we have (11), we have

$$\begin{aligned} \Pr\left[\langle P(w), V^*_{\mathrm{SWI}}(z)\rangle(x) = b \;\middle|\; b \leftarrow \{0,1\};\ (x,w) \leftarrow \mathcal{D}^b_{\mathsf{pk}}\right] &= \frac{1}{2} \cdot \Pr\left[V^*_{\mathrm{SWI}} \text{ aborts}\right] + 1 \cdot \Pr\left[V^*_{\mathrm{SWI}} \text{ does not abort}\right] \\ &= \frac{1}{2} + \frac{1}{2} \cdot \Pr\left[V^*_{\mathrm{SWI}} \text{ does not abort}\right] \\ &\geq \frac{1}{2} + \frac{1}{2}\left(\frac{\epsilon(n)}{8} - \mathsf{negl}(n)\right) \\ &\geq \frac{1}{2} + \frac{\epsilon(n)}{18} \quad , \end{aligned} \tag{13}$$

where to see the first inequality, observe that we have

$$\Pr\left[V^*_{\mathrm{SWI}} \text{ does not abort}\right] \geq \frac{\epsilon(n)}{8} - \mathsf{negl}(n)$$

since if the output out of V^*_{WZK} that is computed in the first step of V^*_{SWI} satisfies (12), we have $\tilde{p}^\star \geq p^\star - \epsilon(n)/16 \geq p - \epsilon(n)/16 - 3\epsilon(n)/8 \geq \tilde{p} - \epsilon(n)/16 - 3\epsilon(n)/8 - \epsilon(n)/16 = \tilde{p} - \epsilon(n)/2$ in V^*_{SWI} unless the approximations of p

[14] Formally, we need to consider a hybrid simulator $\widetilde{S}$ that emulates V^*_{SWI} for R_{SWI} in such a way that V^*_{SWI} uses true randomness instead of pseudorandomness. Due to the security of PRF, the output of $\widetilde{S}$ is indistinguishable from that of S.

and $p^\star$ fails (the second inequality follows from (12)). Thus, by (13) and the definition of interesting keys, any $(\mathsf{pk}, \mathsf{sk})$ such that we have (11) is interesting, so we have

$$\Pr\left[(\mathsf{pk}, \mathsf{sk}) \text{ is interesting} \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)\right] \geq \frac{\epsilon(n)}{4} \ . \tag{14}$$

Combining (7) and (14), we obtain (9).

Next, we analyze the behavior of S under the condition that it does not abort. Since S makes at most polynomially many queries to V^*_{SWI}, it follows from a union bound that with overwhelming probability, in each query the approximations of p and $p^\star$ by V^*_{SWI} are correct, i.e., $\max(|p-\tilde{p}|, |p^\star - \tilde{p}^\star|) \leq \epsilon(n)/16$. Thus, under the condition that S does not abort, with overwhelming probability the output $(i^\star, \mathsf{out}^\star)$ of $S(\{x_i, z_{x,i}\}_{i\in[N_n]}, z_V, z_D)$ satisfies

$$\Pr\left[D_{\mathrm{WZK}}(x_{i^\star}, z_D, \mathsf{out}^\star) = 1\right] \geq \tilde{p} - \frac{\epsilon(n)}{2} - \frac{\epsilon(n)}{16} \geq p - \frac{\epsilon(n)}{2} - \frac{\epsilon(n)}{8} = p - \frac{5\epsilon(n)}{8} \ . \tag{15}$$

Finally, by combining (10) and (15), we obtain

$$\begin{aligned}
&\Pr\left[D_{\mathrm{WZK}}(x_{i^\star}, z_D, \mathsf{out}^\star) = 1 \;\middle|\; \begin{array}{l}(x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N_n] \\ (i^\star, \mathsf{out}^\star) \leftarrow S(\{x_i, z_{x,i}\}_{i\in[N_n]}, z_V, z_D)\end{array}\right] \\
&\geq p - \frac{5\epsilon(n)}{8} - \frac{\epsilon(n)}{10} - \mathsf{negl}(n) \\
&\geq \Pr\left[D_{\mathrm{WZK}}(x, z_D, \langle P(w), V^*_{\mathrm{WZK}}(z_V)\rangle(x)) = 1 \mid (x, w) \leftarrow (\mathcal{X}_n, \mathcal{W}_n)\right] - \epsilon(n)
\end{aligned}$$

as required. Thus, (P, V) is special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional super-weak (t, ϵ)-zero-knowledge in this case.

Completing the proof of the first part of Lemma 2. Combining the analyses of Case 1 and Case 2, we conclude that for any $t, \epsilon, V^*_{\mathrm{WZK}}, D_{\mathrm{WZK}}$, either the assumption (C, c) is false or S is a good simulator for the delayed-input special-purpose $(\mathcal{D}_{xwz}, N)$-distributional super-weak (t, ϵ)-zero-knowledge property of (P, V), where $\mathcal{D}_{xwz}$ and N are defined as above. This completes the proof of the first part of Lemma 2.

Proof of the furthermore part. We define $\overline{\mathcal{D}}_{xz} = \{(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)\}_{n\in\mathbb{N}}$ by

$$(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n) := \left\{((\mathsf{pk}, \mathsf{ct}), \mathsf{sk}_{\{\mathsf{ct}\}}) \;\middle|\; (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n);\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, 2);\ \mathsf{sk}_{\{\mathsf{ct}\}} \leftarrow \mathsf{PuncGen}(\mathsf{sk}, \mathsf{ct})\right\} \ .$$

Due to the (perfect) correctness of PuncPKE, each $(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)$ indeed ranges over $(\{0,1\}^n \setminus L) \times \{0,1\}^*$. Also, $\overline{\mathcal{D}}_{xz}$ is computationally indistinguishable from $\mathcal{D}_{xz} = \{(\mathcal{X}_n, \mathcal{Z}_n)\}_{n\in\mathbb{N}}$ because of the security of PuncPKE. This completes the proof of Lemma 2. □

5 From Special-Purpose Weak ZK to Special-Purpose Pre-Processing ZK

In this section, we show that special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional super-weak (t, ϵ)-zero-knowledge implies special-purpose delayed-input $(\mathcal{D}_{xwz}, N')$-distributional pre-processing (t', ϵ')-zero-knowledge for some N', t', ϵ'.

Lemma 3. *Let (P, V) be a 2-round delayed-input interactive argument for an NP language $\boldsymbol{L}$ and $\mathcal{D}_{xwz} = \{(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)\}_{n\in\mathbb{N}}$ be a sequence of joint distributions such that each $(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)$ ranges over $(\boldsymbol{R}_{\boldsymbol{L}} \times \{0,1\}^*) \cap (\{0,1\}^n \times \{0,1\}^* \times \{0,1\}^*)$. Then, if there exists a polynomial N such that (P, V) is special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional super-weak (t, ϵ)-zero-knowledge for every polynomial t and inverse polynomial ϵ, there also exists a polynomial N' such that (P, V) is special-purpose delayed-input $(\mathcal{D}_{xwz}, N')$-distributional pre-processing (t', ϵ')-zero-knowledge for every polynomial t' and inverse polynomial ϵ'.*

Before giving the proof, let us first give a high-level idea of the proof. Essentially, the proof is obtained by slightly modifying the proof of [CLP15, Theorem 9] (where it is shown that a certain version of weak ZK implies a certain version of ZK as in our lemma). In particular, we prove the lemma by using von Neumann's minimax

theorem as in [CLP15]. Recall that, at a high level, the minimax theorem guarantees that in any finite two-player zero-sum game, if for every strategy for Player 1 there exists a strategy for Player 2 such that Player 2's payoff is v, then there exists a (universal) strategy for Player 2 such that for any strategy for Player 1, Player 2's payoff is v. Now, consider a game where Player 1 chooses a distinguisher, Player 2 chooses a simulator, and Player 2's payoff is defined to be high when the simulator makes the distinguisher output 1 with probability as high as an honest prover. The weak ZK property of (P, V) guarantees that for each distinguisher that Player 1 chooses, there exists a simulator that Player 2 can choose so that Player 2's payoff is high. Thus, intuitively, we can use the minimax theorem to show that Player 2 can choose a (universal) simulator such that for any distinguisher that Player 1 chooses, Player 2's payoff is guaranteed to be high. A subtlety is that to use the minimax theorem, we need to allow Player 1 to choose a distribution over distinguishers (rather than a single distinguisher); thus, we cannot use the weak ZK property directly. We solve this problem as in [CLP15], namely by considering a polynomial-size circuit that approximates a distribution over distinguishers. Another subtlety is that the universal strategy that is guaranteed to exist by the minimax theorem is a distribution of simulators (rather than a single simulator); thus, if we approximate it by a polynomial-size circuit, we only obtain a non-uniform simulator. We solve this problem by considering ZK in the pre-processing model, where we can consider a pre-processing simulator that finds a good non-uniform simulator by using its unbounded computing power.

Now, we give the proof of Lemma 3. We note that much text in the proof is taken from [CLP15].

Proof of Lemma 3. Suppose (P, V) is special-purpose $(\mathcal{D}_{xwz}, N)$-distributional super-weak (t, ϵ)-zero-knowledge for every polynomial t and inverse polynomial ϵ for a polynomial N. Let t' be any polynomial and ϵ' be any inverse polynomial. Let V^* be any PPT verifier and T_{V^*} be any polynomial that bounds the running time of V^*. Without loss of generality, we can assume that the auxiliary inputs $z_V, z_D \in \{0,1\}^*$ in the definition of special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional pre-processing (t', ϵ')-zero-knowledge (Definition 6) satisfy $|z_V| = T_{V^*}(n)$ and $|z_D| = t'(n)$, and we can also remove the absolute value $|\cdot|$ when considering the difference of the probabilities.[15] Thus, it suffices to construct a polynomial N' and a PPT simulator $S = (S_{\mathrm{pre}}, S_{\mathrm{main}})$ such that for every t'-time distinguisher D, there exists an $n_0 \in \mathbb{N}$ such that for every $n > n_0$ and $z_V, z_D \in \{0,1\}^*$ with $|z_V| = T_{V^*}(n)$ and $|z_D| = t'(n)$ and for $N'_n := N'(n, 1/\epsilon'(n))$, it holds

$$\Pr\left[D(x_{i^*}, z_D, v) = 1 \;\middle|\; \begin{array}{l} \mathsf{st}_S \leftarrow S_{\mathrm{pre}}(1^n, z_V) \\ (x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N'_n] \\ (i^*, v) \leftarrow S_{\mathrm{main}}(\{x_i, z_{x,i}\}_{i\in[N_n]}, \mathsf{st}_S) \end{array}\right]$$
$$\geq \Pr\left[D(x, z_D, \langle P(w), V^*(z_V)\rangle(x)) = 1 \mid (x, w) \leftarrow (\mathcal{X}_n, \mathcal{W}_n)\right] - \epsilon'(n) \ .$$

We let N' be a polynomial such that $N'(n, 1/\epsilon'(n)) = N(n, 2/\epsilon'(n))$ for every $n \in \mathbb{N}$. (The reason for this choice will become clear later.)

First, let us describe a useful lemma given in [CLP15]. Roughly speaking, the lemma states that any distribution over circuits can be approximated by a small randomized circuit.

Lemma 4 ([CLP15])**.** *Let X and A be finite sets, let Y be any random variable with finite support, let C be any distribution over s-size randomized circuits of the form $C : X \times \mathrm{Supp}(Y) \to A$, and let U be any finite set of randomized circuits of the form $u : X \times \mathrm{Supp}(Y) \times A \to \{0,1\}$. Then, for every $\delta > 0$, there exists a randomized circuit $\widehat{C}$ of size $T = O(s \cdot (\log|X| + \log|U|)/\delta^2)$ such that for every $u \in U$ and $x \in X$, we have*

$$\left|\mathop{\mathbb{E}}_{C\leftarrow C}[u(x, Y, C(x, Y))] - \mathbb{E}\left[u(x, Y, \widehat{C}(x, Y))\right]\right| \leq \delta \ .$$

Using Lemma 4, we obtain two corollaries, where the first one will be used to approximate a distribution over distinguishers and the second one will be used to approximate a distribution over simulators.

Corollary 1. *Fix any $n \in \mathbb{N}$. Let C be any distribution over s-size randomized circuits of the form $D : \mathrm{Supp}(\mathcal{X}_n) \times \{0,1\}^{t'(n)} \to \{0,1\}$. Then, for every $\delta > 0$, there exists a randomized circuit $\widehat{D}$ of size $T = O(s \cdot (n + t'(n))/\delta^2)$ such that for every $x \in \mathrm{Supp}(\mathcal{X}_n)$ and $v \in \{0,1\}^{t'(n)}$, we have*

$$\left|\Pr_{D\leftarrow C}[D(x, v) = 1] - \Pr[\widehat{D}(x, v) = 1]\right| \leq \delta \ .$$

[15] Specifically, if we have (t', ϵ')-zero-knowledge holds in this version of definition, we have $(t' - 1, \epsilon')$-zero-knowledge in the original version of definition, where we assume for simplicity that we can negate the output of each distinguisher in one step.

Proof. In the statement of Lemma 4, let $X = \mathrm{Supp}(\mathcal{X}_n) \times \{0,1\}^{t'(n)}$, $A = \{0,1\}$, $Y = 0$, and U be the set that only contains the circuit $(x, y, a) \mapsto a$. □

Corollary 2. *Fix any $n \in \mathbb{N}$ and let $N'_n := N'(n, 1/\epsilon(n))$. Let C be any distribution over s-size randomized circuits of the form $S : (\mathrm{Supp}(\mathcal{X}_n) \times \mathrm{Supp}(\mathcal{Z}_n))^{N'_n} \to [N'_n] \times \{0,1\}^{t'(n)}$, and let U_D be any finite set of randomized circuits of the form $D : \mathrm{Supp}(\mathcal{X}_n) \times \{0,1\}^{t'(n)} \to \{0,1\}$. Then, for every $\delta > 0$, there exists a randomized circuit $\widehat{S}$ of size $T = O(s \cdot (\log|U_D|)/\delta^2)$ such that for every $D \in U_D$, we have*

$$\left| \Pr\left[D(x_{i^*}, v) = 1 \;\middle|\; \begin{array}{l} S \leftarrow C \\ (x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N'_n] \\ (i^*, v) \leftarrow S(\{x_i, z_{x,i}\}_{i \in [N'_n]}) \end{array} \right] - \Pr\left[D(x_{i^*}, v) = 1 \;\middle|\; \begin{array}{l} (x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N'_n] \\ (i^*, v) \leftarrow \widehat{S}(\{x_i, z_{x,i}\}_{i \in [N'_n]}) \end{array} \right] \right| \leq \delta$$

Proof. In the statement of Lemma 4, let $X = \emptyset$, $A = [N'_n] \times \{0,1\}^{t'(n)}$, $Y = (\mathcal{X}_n, \mathcal{Z}_n)^{N'_n}$, and U is the set that contains, for each $D \in U_D$, a circuit such that on input of the form $(\{x_i, z_{x,i}\}_{i \in [N'_n]}, (i^*, v))$, it runs $D(x_{i^*}, v)$. □

Next, to obtain a simulator S, for each $n \in \mathbb{N}$ and $z_V \in \{0,1\}^{T_{V^*}(n)}$ we define a two-player zero-sum game between a "simulator player" P_S and a "distinguisher player" P_D. Let $D_1, D_2, D_3, \ldots$ be an enumeration of the set of all (uniform) distinguishers, and let $D'_1, D'_2, D'_3, \ldots$ be the corresponding sequence where D'_j is the same as D_j except that after $t'(n)$ steps, D'_j stops and outputs 0. (Note that each fixed t'-time distinguisher D will eventually appear in the set $\{D'_1, \ldots, D'_n\}$ as n gets larger.) Let $Z_D := \{0,1\}^{t'(n)}$ be the set of all length-$t'(n)$ binary strings. Then, the strategies for P_S and P_D are defined as follows.

- The set Strat_D of pure strategies for P_D is $\{D'_i(\cdot, z_D, \cdot)\}_{i \in [n], z_D \in Z_D}$, i.e., the set that is obtained by hardwiring z_D on D'_i for each $z_D \in Z_D$ and $D'_i \in \{D'_1, \ldots, D'_n\}$.
- The set Strat_S of pure strategies for P_S is, roughly speaking, the set of the distinguisher-dependent simulators that we obtain by first considering circuits that approximate distributions over $\{D'_i(\cdot, z_D, \cdot)\}_{i \in [n], z_D \in Z_D}$ and then using the weak ZK property of (P, V) for them. Formally, let $T_{\widehat{D}}$ be the size of the circuit that we obtain by using Corollary 1 on a distribution[16] over $\{D'_i(\cdot, z_D, \cdot)\}_{i \in [n], z_D \in Z_D}$ with $\delta := \epsilon'(n)/8$ (each $D'_i(\cdot, z_D, \cdot)$ is viewed as a circuit). Let D_U be a Turing machine such that on input of the form (x, C, v) for a $T_{\widehat{D}}$-size circuit C, it runs $C(x, v)$. Let T_{D_U} be the running time of D_U, and let S_{D_U} be the simulator that is guaranteed to exist for the distinguisher D_U by the special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional super-weak $(T_{D_U}, \epsilon'/2)$-zero-knowledge property of (P, V). Then, the set Strat_S of pure strategies for P_S is the set that contains $S_{D_U}(\cdot, z_V, C)$ for every $T_{\widehat{D}}$-size circuit C.[17]

For each $S' \in \mathsf{Strat}_S$ and $D' \in \mathsf{Strat}_D$, let the "payoff" of P_S (w.r.t. V^*, n, and z_V) is defined by

$$\begin{aligned} \mu_{n,z_V}(S', D') := \Pr\left[D'(x_{i^*}, v) = 1 \;\middle|\; \begin{array}{l} (x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N'_n] \\ (i^*, v) \leftarrow S'(\{x_i, z_{x,i}\}_{i \in [N'_n]}) \end{array} \right] \\ - \Pr\left[D'(x, \langle P(w), V^*(z_V) \rangle(x)) = 1 \mid (x, w) \leftarrow (\mathcal{X}_n, \mathcal{W}_n) \right] \ , \end{aligned}$$

where $N'_n := N'(n, 1/\epsilon'(n)) = N(n, 2/\epsilon'(n))$. Also, for any mixed strategies (i.e., distributions) $\mathcal{S}$ over Strat_S and $\mathcal{D}$ over Strat_D, the expected payoff of P_S is defined by

$$\mu_{n,z_V}(\mathcal{S}, \mathcal{D}) := \mathop{\mathbb{E}}_{S' \leftarrow \mathcal{S}, D' \leftarrow \mathcal{D}} \left[\mu_{n,z_V}(S', D') \right] = \sum_{S', D'} \Pr_{S \leftarrow \mathcal{S}, D \leftarrow \mathcal{D}} \left[S = S' \wedge D = D' \right] \cdot \mu_{n,z_V}(S', D') \ .$$

Jumping ahead, below we show that there exists a (non-uniform) simulator $\widehat{S}$ such that for any $D \in \mathsf{Strat}_D$, we have $\mu_{n,z_V}(\widehat{S}, D) \geq -\epsilon'(n)$.

Now, by using the above game, we obtain a simulator S. Roughly speaking, we proceed in four steps (more details will be given shortly).

[16] Note that in Corollary 1, the size T of the randomized circuit is independent of the distribution C.

[17] Here, S_{D_U} is given z_V as the auxiliary input to the verifier V^* and is given C as the auxiliary input to the distinguisher D_U (cf. Definition 7).

Step 1. We first show that for any sufficiently large $n \in \mathbb{N}$, any $z_V \in \{0,1\}^{T_{V^*}(n)}$, and any mixed strategy $\mathcal{D}$ for P_D (i.e., any distribution over Strat_D), there exists a simulator $S_{\mathcal{D}} \in \mathsf{Strat}_S$ such that $\mu_{n,z_V}(S_{\mathcal{D}}, \mathcal{D}) \geq -3\epsilon'(n)/4$. Toward this end, we first use Corollary 1 to show that we can approximate $\mathcal{D}$ by a $T_{\widehat{D}}$-size distinguisher $\widehat{D}$, and then use the special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional super-weak $(T_{D_U}, \epsilon'/2)$-zero-knowledge property of (P, V) to obtain a simulator $S_{\widehat{D}}$ for $\widehat{D}$ such that $\mu_{n,z_V}(S_{\widehat{D}}, \widehat{D}) \geq -\epsilon'(n)/2$. Finally, by recalling that $\widehat{D}$ approximates $\mathcal{D}$, we show $\mu_{n,z_V}(S_{\widehat{D}}, \mathcal{D}) \geq -3\epsilon'(n)/4$.

Step 2. We now apply the minimax theorem to the result of Step 1 to obtain that for any sufficiently large $n \in \mathbb{N}$ and any $z_V \in \{0,1\}^{T_{V^*}(n)}$, there exists a mixed strategy $\mathcal{S}$ for P_S (i.e., a distribution over Strat_S) such that for every distinguisher $D \in \mathsf{Strat}_D$, we have $\mu_{n,z_V}(\mathcal{S}, D) \geq -3\epsilon'(n)/4$.

Step 3. We next use Corollary 2 to show that for any sufficiently large $n \in \mathbb{N}$ and any $z_V \in \{0,1\}^{T_{V^*}(n)}$, we can approximate $\mathcal{S}$ (from Step 2) by a polynomial-size circuit $\widehat{S}$ so that $\mu_{n,z_V}(\widehat{S}, D) \geq -\epsilon'(n)$ for every distinguisher $D \in \mathsf{Strat}_D$.

Step 4. Finally, we obtain $S = (S_{\mathrm{pre}}, S_{\mathrm{main}})$ as follows. First, on input $(1^n, z_V)$, the pre-processing simulator S_{pre} finds the polynomial-size circuit $\widehat{S}$ (from Step 3) by brute force and outputs it as the intermediate state. Next, on input $\{x_i, z_{x,i}\}_{i\in[N_n]}$ and $\widehat{S}$, the main simulator S_{main} simply runs $\widehat{S}$ on $\{x_i, z_{x,i}\}_{i\in[N_n]}$.

Details of Step 1, Step 3, and Step 4 are given below.

Details of Step 1. Let n be sufficiently large (in particular, n should be large enough that the distinguisher-dependent simulator S_{D_U} works for the universal distinguisher D_U, where D_U and S_{D_U} are defined in the definition of Strat_S), and fix any $z_V \in \{0,1\}^{T_{V^*}(n)}$ and any mixed strategy $\mathcal{D}$ for P_D. Let $\widehat{D}$ be the $T_{\widehat{D}}$-size randomized circuit that we obtain by using Corollary 1 on the distribution $\mathcal{D}$ with $\delta := \epsilon'(n)/8$ so that for every $x \in \mathrm{Supp}(\mathcal{X}_n)$ and $v \in \{0,1\}^{t'(n)}$, we have

$$\left|\Pr_{D\leftarrow\mathcal{D}}[D(x,v)=1] - \Pr[\widehat{D}(x,v)=1]\right| \leq \frac{\epsilon'(n)}{8} \quad . \tag{16}$$

(Recall that $T_{\widehat{D}}$ is defined in the definition of Strat_S above.) First, we observe that there exists $S_{\mathcal{D}} \in \mathsf{Strat}_S$ such that $\mu_{n,z_V}(S_{\mathcal{D}}, \widehat{D}) \geq -\epsilon'(n)/2$. To see this, observe that for $S_{\mathcal{D}} := S_{D_U}(\cdot, z_V, \widehat{D}) \in \mathsf{Strat}_S$, we have

$$\begin{aligned}
&\Pr\left[\widehat{D}(x_{i^*}, v) = 1 \;\middle|\; \begin{array}{l}(x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N'_n] \\ (i^*, v) \leftarrow S_{\mathcal{D}}(\{x_i, z_{x,i}\}_{i\in[N_n]})\end{array}\right] \\
&= \Pr\left[D_U(x_{i^*}, \widehat{D}, v) = 1 \;\middle|\; \begin{array}{l}(x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N'_n] \\ (i^*, v) \leftarrow S_{D_U}(\{x_i, z_{x,i}\}_{i\in[N_n]}, z_V, \widehat{D})\end{array}\right] && \text{(due to the definitions of } D_U, S_{\mathcal{D}}) \\
&\geq \Pr\left[D_U(x, \widehat{D}, \langle P(w), V^*(z_V)\rangle(x)) = 1 \mid (x,w) \leftarrow (\mathcal{X}_n, \mathcal{W}_n)\right] - \frac{\epsilon'(n)}{2} \\
&= \Pr\left[\widehat{D}(x, \langle P(w), V^*(z_V)\rangle(x)) = 1 \mid (x,w) \leftarrow (\mathcal{X}_n, \mathcal{W}_n)\right] - \frac{\epsilon'(n)}{2} \quad , && \text{(due to the definition of } D_U)
\end{aligned}$$

where the inequality holds since the simulator S_{D_U} is obtained by the special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional super-weak $(T_{D_U}, \epsilon'/2)$-zero-knowledge of (P, V) for the universal distinguisher D_U. Then, since we have $|\mu_{n,z_V}(S_{\mathcal{D}}, \mathcal{D}) - \mu_{n,z_V}(S_{\mathcal{D}}, \widehat{D})| \leq \epsilon'(n)/4$ due to (16), we obtain $\mu_{n,z_V}(S_{\mathcal{D}}, \mathcal{D}) \geq -3\epsilon'(n)/4$ as required.

Details of Step 3. By Corollary 2, there exists a polynomial-size circuit $\widehat{S}$ such that for every distinguisher $D \in \mathsf{Strat}_D$ and for the distribution $\mathcal{S}$ that is obtained in Step 2, we have

$$\left|\Pr\left[D(x_{i^*}, v) = 1 \;\middle|\; \begin{array}{l}S \leftarrow \mathcal{S} \\ (x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N'_n] \\ (i^*, v) \leftarrow S(\{x_i, z_{x,i}\}_{i\in[N'_n]})\end{array}\right] - \Pr\left[D(x_{i^*}, v) = 1 \;\middle|\; \begin{array}{l}(x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N'_n] \\ (i^*, v) \leftarrow \widehat{S}(\{x_i, z_{x,i}\}_{i\in[N'_n]})\end{array}\right]\right| \leq \frac{\epsilon'(n)}{4} \quad .$$

In other words, for $\widehat{S}$, we have $|\mu_{n,z_V}(\mathcal{S}, D) - \mu_{n,z_V}(\widehat{S}, D)| \leq \epsilon'(n)/4$ for every $D \in \mathsf{Strat}_D$. Thus, by the result of Step 2, we have $\mu_{n,z_V}(\widehat{S}, D) \geq -\epsilon'(n)$ for every $D \in \mathsf{Strat}_D$.

Details of Step 4. Consider the following simulator $S = (S_{\text{pre}}, S_{\text{main}})$. On input $(1^n, z_V)$, the pre-processing simulator S_{pre} uses its unbounded computing power to find a polynomial-size circuit $\widehat{S}$ that satisfies $\mu_{n,z_V}(\widehat{S}, D) \geq -\epsilon'(n)$ for every $D \in \mathsf{Strat}_D$. (Note that $\mu_{n,z_V}(\widehat{S}, D)$ can be computed given $(1^n, z_V)$, so S_{pre} can indeed check whether a circuit $\widehat{S}$ satisfies $\mu_{n,z_V}(\widehat{S}, D) \geq -\epsilon'(n)$.) The output of S_{pre} is $\mathsf{st}_S := \widehat{S}$. Then, on input $(\{x_i, z_{x,i}\}_{i\in[N_n]}, \mathsf{st}_S)$, the main simulator S_{main} parses st_S as $\widehat{S}$ and outputs whatever $\widehat{S}(\{x_i, z_{x,i}\}_{i\in[N_n]})$ outputs.

We now analyze S. Fix any probabilistic t-time distinguisher D, and fix any sufficiently large $n \in \mathbb{N}$ (in particular, n should be large enough that D appears in $\{D'_1, \ldots, D'_n\}$ and that the distinguisher-dependent simulator S_{D_U} works for the universal distinguisher D_U). Fix any $z_V \in \{0,1\}^{T_{V^*}(n)}$ and $z_D \in Z_{D,n} = \{0,1\}^{t'(n)}$. Note that by the definition of Strat_D, we have $D(\cdot, z_D, \cdot) \in \mathsf{Strat}_D$. Thus, by the analysis of Step 3, the pre-processing simulator S_{pre} can always find a polynomial-size circuit $\widehat{S}$ such that $\mu_{n,z_V}(\widehat{S}, D) \geq -\epsilon'(n)$, i.e.,

$$\begin{aligned}
&\Pr\left[D(x_{i^*}, z_D, v) = 1 \;\middle|\; \begin{array}{l} \widehat{S} \leftarrow S_{\text{pre}}(1^n, z_V) \\ (x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N'_n] \\ (i^*, v) \leftarrow \widehat{S}(\{x_i, z_{x,i}\}_{i\in[N'_n]}) \end{array}\right] \\
&\geq \Pr\left[D(x, z_D, \langle P(w), V^*(z_V)\rangle(x)) = 1 \mid (x, w) \leftarrow (\mathcal{X}_n, \mathcal{W}_n)\right] - \epsilon'(n) \ .
\end{aligned}$$

Thus, by the construction of $S = (S_{\text{pre}}, S_{\text{main}})$, we obtain

$$\begin{aligned}
&\Pr\left[D(x_{i^*}, z_D, v) = 1 \;\middle|\; \begin{array}{l} \mathsf{st}_S \leftarrow S_{\text{pre}}(1^n, z_V) \\ (x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N'_n] \\ (i^*, v) \leftarrow S_{\text{main}}(\{x_i, z_{x,i}\}_{i\in[N_n]}, \mathsf{st}_S) \end{array}\right] \\
&= \Pr\left[D(x_{i^*}, z_D, v) = 1 \;\middle|\; \begin{array}{l} \widehat{S} \leftarrow S_{\text{pre}}(1^n, z_V) \\ (x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n) \text{ for } \forall i \in [N'_n] \\ (i^*, v) \leftarrow \widehat{S}(\{x_i, z_{x,i}\}_{i\in[N_n]}) \end{array}\right] \\
&\geq \Pr\left[D(x, z_D, \langle P(w), V^*(z_V)\rangle(x)) = 1 \mid (x, w) \leftarrow (\mathcal{X}_n, \mathcal{W}_n)\right] - \epsilon'(n)
\end{aligned}$$

as required. This completes the proof of Lemma 3. □

6 BB Impossibility of 2-Round Special-Purpose Pre-Processing ZK

In this section, we give a BB impossibility result about special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional pre-processing (t, ϵ)-zero-knowledge.

Lemma 5. *Let $\mathbf{L}$ be an NP language and $\mathcal{D}_{xwz} = \{(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)\}_{n\in\mathbb{N}}$ be a sequence of efficient joint distributions such that (i) each $(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)$ ranges over $(\boldsymbol{R}_{\mathbf{L}} \times \{0,1\}^*) \cap (\{0,1\}^n \times \{0,1\}^* \times \{0,1\}^*)$ and (ii) there exists a sequence of joint distributions $\overline{\mathcal{D}}_{xz} = \{(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)\}_{n\in\mathbb{N}}$ such that $\overline{\mathcal{D}}_{xz}$ is computationally indistinguishable from $\mathcal{D}_{xz} := \{(\mathcal{X}_n, \mathcal{Z}_n)\}_{n\in\mathbb{N}}$ and each $(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)$ ranges over $(\{0,1\}^n \setminus L) \times \{0,1\}^*$. Then, if there exists a 2-round delayed-input interactive argument (P, V) for $\mathbf{L}$ such that*

- *there exists a polynomial N such that (P, V) is special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional pre-processing (t, ϵ)-zero-knowledge for every polynomial t and every inverse polynomial ϵ, and*
- *there exists a black-box reduction R for showing the adaptive soundness of (P, V) based on a falsifiable assumption (C, c),*

then, the assumption (C, c) is false.

As mentioned in Section 2.2, the proof of this lemma closely follows the proof of [CLMP12, Theorem 2]. We note that much text in the proof is taken from [CLMP12].

Proof. Assume that there exist a 2-round delayed-input interactive argument (P, V) for $\mathbf{L}$, a polynomial N, a black-box reduction R, and a falsifiable assumption (C, c) that satisfy the conditions stated in the lemma. Following the "meta-reduction" paradigm by Boneh and Venkatesan [BV98] (which is also used in, e.g., [Pas11, GW11, CLMP12, Pas13]) we will use the reduction R to efficiently break the assumption (C, c). More formally,

we construct an inefficient cheating prover that breaks the soundness of (P, V) with overwhelming probability (which implies that the reduction R breaks the assumption (C, c) when it is combined with this cheating prover), and we next show how to emulate this cheating prover for R efficiently without disturbing R's interaction with the challenger C.

We first describe an inefficient cheating prover $\mathcal{A}$ that breaks the adaptive soundness of (P, V). More precisely (as in [Pas11, CLMP12]), we define a class of deterministic provers $\mathcal{A}^f$, parameterized by a function $f : \{0,1\}^* \to \{0,1\}^\infty$. Given that $\mathcal{A}^f$ is deterministic, we can assume without loss of generality that R never asks its oracle the same query twice. Let V^* be the delayed-input verifier such that on input $(1^n, z)$, it sends z to the prover as the first message of (P, V), and after receiving a response a from the prover, it simply outputs a. Let $S = (S_{\text{pre}}, S_{\text{main}})$ be the simulator that is guaranteed to exist for V^* by the special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional pre-processing (t, ϵ)-zero-knowledge property of (P, V), where t and ϵ are the parameters that we will specify later (see (17) and (18)). Note that for any $n \in \mathbb{N}$, $z \in \{0,1\}^*$, $\mathsf{st}_S \leftarrow S_{\text{pre}}(1^n, z)$, and $\{(x_i, z_{x,i})\}_{i\in[N_n]}$ (where $N_n := N(n, 1/\epsilon(n))$), the main simulator $S_{\text{main}}(\{x_i, z_{x,i}\}_{i\in[N_n]}, \mathsf{st}_S)$ outputs some (i^*, v) such that v is the same format as the output of V^*, i.e., v can be viewed as a prover message w.r.t. the verifier message z and the statement x_{i^*}. Now, on input the security parameter 1^n and a verifier message q of (P, V), the cheating prover $\mathcal{A}^f$ does the following by using $f(1^n \| q)$ as randomness: (i) compute $\mathsf{st}_S \leftarrow S_{\text{pre}}(1^n, q)$, (ii) sample $(x_i, z_{x,i}) \leftarrow (\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)$ for each $i \in [N_n]$, (iii) compute $(i^*, a) \leftarrow S_{\text{main}}(\{x_i, z_{x,i}\}_{i\in[N_n]}, \mathsf{st}_S)$ and send (x_{i^*}, a) to the prover.

Let us see that $\mathcal{A}$ indeed breaks the adaptive soundness of (P, V). Let $\mathsf{RO} : \{0,1\}^* \to \{0,1\}^\infty$ be a uniformly distributed random oracle. Our claim is that $\mathcal{A}^{\mathsf{RO}}$ breaks the adaptive soundness of (P, V) with overwhelming probability. First note that with probability 1 (over the choice of RO), $\mathcal{A}^{\mathsf{RO}}$ selects a false statement $x_{i^*} \notin \mathbf{L}$ due to our assumption on $\overline{\mathcal{D}}_{xz}$. Now, consider an alternative cheating prover $\mathcal{A}^f_{\text{ALT}}$ that is identical with $\mathcal{A}^f$ except that $\mathcal{A}^f_{\text{ALT}}$ samples $(x_i, z_{x,i}) \leftarrow (\mathcal{X}_n, \mathcal{Z}_n)$ for each $i \in [N_n]$ (again by using $f(1^n \| q)$ as the randomness); that is, the difference from $\mathcal{A}^f$ is that $\mathcal{A}^f_{\text{ALT}}$ samples true statements rather than false statements. It follows from the (non-uniform) indistinguishability property of the simulator S and the completeness of (P, V) that $\mathcal{A}^{\mathsf{RO}}_{\text{ALT}}$ convinces an honest verifier for a true statement $x_{i^*} \in \mathbf{L}$ with overwhelming probability.[18] Now, it follows from the (non-uniform) indistinguishability between $\mathcal{D}_{xz}$ and $\overline{\mathcal{D}}_{xz}$ that $\mathcal{A}^{\mathsf{RO}}$ convinces the honest verifier for a false statement $x_{i^*} \notin \mathbf{L}$ with overwhelming probability.[19] That is, $\mathcal{A}^{\mathsf{RO}}$ breaks the soundness of (P, V) with probability $\mu(\cdot) = 1 - \nu(\cdot)$ for a negligible function ν.

Next, let us see that the reduction R breaks the assumption (C, c) when it is combined with $\mathcal{A}$. By an averaging argument, with probability at least $1 - 10\nu(n)$ over the choice of a random oracle $f \leftarrow \mathsf{RO}$, the cheating prover $\mathcal{A}^f$ breaks the adaptive soundness of (P, V) with probability at least $9/10$. Therefore, there exists a polynomial Q (independent of S) such that for each such "good" choice of f, the reduction $R^{\mathcal{A}^f}$ breaks the assumption (C, c) with non-negligible advantage $1/Q(n)$. By a union bound, it follows that $R^{\mathcal{A}^{\mathsf{RO}}}(1^n)$ breaks the assumption (C, c) with advantage $1/2Q(n)$ for sufficiently large n.

We now construct a PPT cheating prover $\widetilde{\mathcal{A}}$ that emulates $\mathcal{A}$. The cheating prover $\widetilde{\mathcal{A}}$, on input the security parameter 1^n and a verifier message q, uniformly samples $(x, w) \leftarrow (\mathcal{X}_n, \mathcal{W}_n)$, runs the honest prover strategy $P(x, w)$ on input the message q, and outputs x and whatever P outputs.

We now show the following claim, which concludes the proof of Lemma 5.

Claim 1. *For sufficiently large n, the reduction $R^{\widetilde{\mathcal{A}}}$ breaks the assumption (C, c) with advantage at least $1/4Q(n)$ on common input 1^n.*

Proof. Let $m(n)$ be an upper bound on the running time of R, and define a sequence of $m(n) + 1$ hybrids $H_0, \ldots, H_{m(n)}$ as follows.

- In the hybrid H_i, the challenger C interacts with the reduction $R^{(\cdot)}$ where the first i oracle responses are

[18] In particular, we use the indistinguishability w.r.t. $V^*(1^n, z)$ where the auxiliary input z is an honest verifier message q. We require non-uniform indistinguishability since we use indistinguishability against a distinguisher that takes as additional auxiliary input the randomness that is used to generate q. (The randomness is used by the distinguisher to check whether the given prover message is accepting or not.)

[19] We require non-uniform indistinguishability since we use the indistinguishability against a distinguisher that takes as auxiliary input (i) an honest verifier message q and its randomness and (ii) the pre-processing simulator's output $\mathsf{st}_S \leftarrow S_{\text{pre}}(1^n, q)$. (These auxiliary inputs are used by the distinguisher to check whether the main simulator S_{main} outputs an accepting prover message on the given statement $(x_i, z_{x,i})$.)

simulated (i.e., answered by $\mathcal{A}^{\mathsf{RO}}$) and the remaining queries are answered honestly (i.e., answered by $\widetilde{\mathcal{A}}$).[20] The output of the hybrid H_i is that of the challenger C.

Note that H_0 is the output of C after interacting with $R^{\widetilde{\mathcal{A}}}$ and $H_{m(n)}$ is the output of C after interacting with $R^{\mathcal{A}^{\mathsf{RO}}}$. Roughly speaking, we show the indistinguishability between each two consecutive hybrids H_i and H_{i+1} by combining the indistinguishability of the simulation, the indistinguishability between $\mathcal{D}_{xz}$ and $\overline{\mathcal{D}}_{xz}$, and the fact that oracle responses for all $j > i+1$ can be generated in polynomial time. Formally, let us consider the following intermediate hybrid H_i^+.

- The hybrid H_i^+ is identical with H_i except that the $(i+1)$-th oracle response is answered by the alternative cheating prover $\mathcal{A}^{\mathsf{RO}}_{\mathrm{ALT}}$.

We first show that H_i and H_i^+ are distinguishable with advantage at most $1/5m(n)Q(n)$ by relying on the (non-uniform) indistinguishability of the simulation. (Recall that H_i^+ differs from H_i in that $\mathcal{A}^{\mathsf{RO}}_{\mathrm{ALT}}$ is used instead of $\widetilde{\mathcal{A}}$ in the $(i+1)$-th oracle response, where $\mathcal{A}^{\mathsf{RO}}_{\mathrm{ALT}}$ differs from $\widetilde{\mathcal{A}}$ in that $\mathcal{A}^{\mathsf{RO}}_{\mathrm{ALT}}$ generates a simulated proof for a true statement whereas $\widetilde{\mathcal{A}}$ generates an honest proof for a true statement.)

> **Details:** Set the parameter t and ϵ for the special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional pre-processing (t,ϵ)-zero-knowledge property of (P,V) as
>
> $$t(n) \text{ is a polynomial that upper bounds the joint running time of } C \text{ and } R^{\tilde{\mathcal{A}}}. \tag{17}$$
>
> $$\epsilon(n) = \frac{1}{5m(n)Q(n)} \ . \tag{18}$$
>
> (Note that both t and ϵ are independent of i and S.) Assume for contradiction that H_i and H_i^+ are distinguishable with advantage greater than $1/5m(n)Q(n)$. By an average argument, we can fix the execution of H_i up until the $(i+1)$-th query (inclusive) in such a way that H_i and H_i^+ are still distinguishable with advantage greater than $1/5m(n)Q(n)$. Let q_{i+1} and $\mathsf{st}_{R,C}$ be the $(i+1)$-th query and the joint internal state of R and C at the point of the $(i+1)$-th query of this fixed execution, respectively. Now, it is easy to see that we can break the special-purpose delayed-input $(\mathcal{D}_{xwz}, N)$-distributional pre-processing (t,ϵ)-zero-knowledge property of (P,V) (w.r.t. the cheating verifier V^* that is defined at the beginning of the proof of Lemma 5)—namely, we let q_{i+1} be the auxiliary input to the verifier and $\mathsf{st}_{R,C}$ be the auxiliary input to the distinguisher, and we consider a distinguisher that continues the execution of H_i from the $(i+1)$-th oracle response in such a way that the statement and the output of V^* (which is viewed as a prover's response) are used as the $(i+1)$-th oracle response. (Note that the running time of such a distinguisher is indeed bounded by t.)

We next show that H_i^+ and H_{i+1} are indistinguishable by relying on the (non-uniform) indistinguishability between $\mathcal{D}_{xz}$ and $\overline{\mathcal{D}}_{xz}$. (Recall that H_{i+1} differs from H_i^+ in that $\mathcal{A}^{\mathsf{RO}}$ is used instead of $\mathcal{A}^{\mathsf{RO}}_{\mathrm{ALT}}$ in the $(i+1)$-th oracle response, where $\mathcal{A}^{\mathsf{RO}}$ differs from $\mathcal{A}^{\mathsf{RO}}_{\mathrm{ALT}}$ in that $\mathcal{A}^{\mathsf{RO}}_{\mathrm{ALT}}$ generates a simulated proof for a true statement whereas $\mathcal{A}^{\mathsf{RO}}$ generates a simulated proof for a false statement.)

> **Details:** Assume for contradiction that H_i^+ and H_{i+1} are distinguishable with non-negligible advantage, and fix the execution of H_i^+ until the point that $\mathsf{st}_S \leftarrow S_{\mathrm{pre}}(1^n, q_{i+1})$ is computed during the $(i+1)$-th oracle response (inclusive) in such a way that H_i^+ and H_{i+1} are still distinguishable with non-negligible advantage. Now it is easy to see that we can break the (multiple version of the) indistinguishability between $\mathcal{D}_{xz}$ and $\overline{\mathcal{D}}_{xz}$ by considering a distinguisher that continues the execution of H_i^+ in such a way that the given statements $\{x_i, z_{x,i}\}_{i\in[N_n]}$ is used as the input to S_{main} in the $(i+1)$-th oracle response.

From the above two, it follows that H_i and H_{i+1} are distinguishable with advantage at most $1/5m(n)Q(n) + \mathsf{negl}(n) \le 1/4m(n)Q(n)$ for sufficiently large n. Thus, H_0 and $H_{m(n)}$ are distinguishable with advantage at most $1/4Q(n)$. Since as shown above $R^{\mathcal{A}^{\mathsf{RO}}}$ breaks the assumption (C,c) with advantage $1/2Q(n)$ (and therefore the challenger C in $H_{m(n)}$ outputs 1 with probability at least $c + 1/2Q(n)$), we conclude that $R^{\widetilde{\mathcal{A}}}$ breaks the assumption (C,c) with advantage $1/2Q(n) - 1/4Q(n) = 1/4Q(n)$. This concludes the proof of Claim 1. □

[20] Since (P,V) is a 2-round protocol, we can assume without loss of generality that the reduction makes queries sequentially. Also, recall that it is guaranteed that R does not make the same query twice.

This concludes the proof of Lemma 5. □

7 Obtaining Main Results

In this section, we obtain our main results by using the lemmas given in the previous sections.

7.1 BB Impossibility of 2-Round Delayed-Input Weak ZK

By using Lemma 3 and Lemma 5, we obtain the following black-box impossibility result about 2-round delayed-input weak ZK.

Theorem 1. *Assume the existence of one-way functions. Then, there exists an NP language $\mathbf{L}$ such that if there exist*

- *a 2-round delayed-input interactive argument (P, V) for $\mathbf{L}$ that is delayed-input distributional weak (t, ϵ)-zero-knowledge for every polynomial t and inverse polynomial ϵ, and*
- *a black-box reduction R for showing the adaptive soundness of (P, V) based on a falsifiable assumption (C, c),*

then the assumption (C, c) is false.

Proof. Let PRG be any pseudorandom generator (which can be obtained from one-way functions [HILL99]) and $\mathbf{L}$ be the NP language that is defined by $\mathbf{L} := \{\mathsf{PRG}(s) \mid s \in \{0,1\}^*\}$, where we assume without loss of generality that PRG is length-doubling. For each $n \in \mathbb{N}$, consider the following joint distributions $(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)$ and $(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)$.

$$(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n) := \left\{(\mathsf{PRG}(s), s, \bot) \mid s \leftarrow \{0,1\}^{n/2}\right\}$$
$$(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n) := \{(r, \bot) \mid r \leftarrow \{0,1\}^n \setminus \mathbf{L}\}$$

It is easy to see that $\{(\mathcal{X}_n, \mathcal{Z}_n)\}_{n\in\mathbb{N}}$ and $\{(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)\}_{n\in\mathbb{N}}$ are computationally indistinguishable, and delayed-input distributional weak (t, ϵ)-zero-knowledge implies special-purpose delayed-input $(\mathcal{D}_{xwz}, 1)$-distributional super-weak (t, ϵ)-zero-knowledge, where $\mathcal{D}_{xwz} = \{(\mathcal{X}_n, \mathcal{W}_n, \mathcal{Z}_n)\}_{n\in\mathbb{N}}$. Now, the lemma follows from Lemma 3 and Lemma 5. □

7.2 BB Impossibility of 2-Round Delayed-Input Strong WI

By combining Lemma 1, Lemma 2, Lemma 3, and Lemma 5, we immediately obtain the following black-box impossibility result about 2-round delayed-input strong WI.

Theorem 2. *Assume the existence of trapdoor permutations. Then, there exists an NP language $\mathbf{L}$ such that if there exist*

- *a 2-round delayed-input interactive argument (P, V) for $\mathbf{L}$,*
- *an oblivious black-box reduction R for showing the delayed-input strong WI of (P, V) based on a falsifiable assumption (C, c), and*
- *a black-box reduction R' for showing the adaptive soundness of (P, V) based on a falsifiable assumption (C', c'),*

then either the assumption (C, c) is false or the assumption (C', c') is false.

7.3 BB Impossibility of 2-Round (Non-Delayed-Input) Strong WI

By adjusting the proof of Lemma 2, we obtain the following black-box impossibility result about 2-round (non-delayed-input) strong WI.

Theorem 3. *Assume the existence of CCA-secure public-key encryption schemes. Then, there exists an NP language $\boldsymbol{L}$ such that if there exist*

- *a 2-round interactive argument (P, V) for $\boldsymbol{L}$ and*
- *an oblivious black-box reduction R for showing the strong WI of (P, V) based on a falsifiable assumption (C, c),*

then the assumption (C, c) is false.

Since Theorem 3 can be proven by closely following the proof of Lemma 2, we give the proof in the appendix (Appendix B).

7.4 BB Impossibility of 2-Round Publicly Verifiable Delayed-Input Strong WI

By adjusting the proof of Lemma 2, we obtain the following black-box impossibility result about 2-round delayed-input publicly verifiable strong WI.

Theorem 4. *Assume the existence of CCA-secure public-key encryption schemes. Then, there exists an NP language $\boldsymbol{L}$ such that if there exist*

- *a 2-round delayed-input publicly verifiable interactive argument (P, V) for $\boldsymbol{L}$ and*
- *an oblivious black-box reduction R for showing the delayed-input strong WI of (P, V) based on a falsifiable assumption (C, c),*

then the assumption (C, c) is false.

Since Theorem 4 can be proven very similarly to Theorem 3 (as mentioned in Section 2), we omit the proof.

References

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.

[BCPR16] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the existence of extractable one-way functions. *SIAM Journal on Computing*, 45(5):1910–1952, 2016.

[BFJ+20] Saikrishna Badrinarayanan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. Statistical ZAP arguments. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 642–667. Springer, Heidelberg, May 2020.

[BGI+17] Saikrishna Badrinarayanan, Sanjam Garg, Yuval Ishai, Amit Sahai, and Akshay Wadia. Two-message witness indistinguishability and secure computation in the plain model from new assumptions. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 275–303. Springer, Heidelberg, December 2017.

[BKP19] Nir Bitansky, Dakshita Khurana, and Omer Paneth. Weak zero-knowledge beyond the black-box barrier. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1091–1102. ACM Press, June 2019.

[BMO90] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. The (true) complexity of statistical zero knowledge. In *22nd ACM STOC*, pages 494–502. ACM Press, May 1990.

[BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking RSA may not be equivalent to factoring. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 59–71. Springer, Heidelberg, May / June 1998.

[CLMP12] Kai-Min Chung, Edward Lui, Mohammad Mahmoody, and Rafael Pass. Unprovable security of two-message zero knowledge. Cryptology ePrint Archive, Report 2012/711, 2012. https://eprint.iacr.org/2012/711.

[CLP15] Kai-Min Chung, Edward Lui, and Rafael Pass. From weak to strong zero-knowledge and applications. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part I*, volume 9014 of *LNCS*, pages 66–92. Springer, Heidelberg, March 2015.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

[DJKL12] Dana Dachman-Soled, Abhishek Jain, Yael Tauman Kalai, and Adriana Lopez-Alt. On the (in)security of the Fiat-Shamir paradigm, revisited. Cryptology ePrint Archive, Report 2012/706, 2012. https://eprint.iacr.org/2012/706.

[DN07] Cynthia Dwork and Moni Naor. Zaps and their applications. *SIAM Journal on Computing*, 36(6):1513–1543, 2007.

[DNRS03] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. *Journal of the ACM*, 50(6):852–921, 2003.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM Journal on Computing*, 29(1):1–28, 1999.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *22nd ACM STOC*, pages 416–426. ACM Press, May 1990.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013.

[GJJM20] Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. Statistical zaps and new oblivious transfer protocols. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 668–699. Springer, Heidelberg, May 2020.

[GM15] Matthew D. Green and Ian Miers. Forward secure asynchronous messaging from puncturable encryption. In *2015 IEEE Symposium on Security and Privacy*, pages 305–320. IEEE Computer Society Press, May 2015.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, December 1994.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001.

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *Journal of the ACM*, 59(3), June 2012.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011.

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

[HRS09] Iftach Haitner, Alon Rosen, and Ronen Shaltiel. On the (im)possibility of Arthur-Merlin witness hiding protocols. In Omer Reingold, editor, *TCC 2009*, volume 5444 of *LNCS*, pages 220–237. Springer, Heidelberg, March 2009.

[JKKR17] Abhishek Jain, Yael Tauman Kalai, Dakshita Khurana, and Ron Rothblum. Distinguisher-dependent simulation in two rounds and its applications. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 158–189. Springer, Heidelberg, August 2017.

[KKS18] Yael Tauman Kalai, Dakshita Khurana, and Amit Sahai. Statistical witness indistinguishability (and more) in two messages. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 34–65. Springer, Heidelberg, April / May 2018.

[KS17] Dakshita Khurana and Amit Sahai. How to achieve non-malleability in one or two rounds. In Chris Umans, editor, *58th FOCS*, pages 564–575. IEEE Computer Society Press, October 2017.

[LVW20] Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. Statistical ZAPR arguments from bilinear maps. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 620–641. Springer, Heidelberg, May 2020.

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge University Press, 2017.

[Nao03] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, Heidelberg, August 2003.

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 160–176. Springer, Heidelberg, May 2003.

[Pas11] Rafael Pass. Limits of provable security from standard assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 109–118. ACM Press, June 2011.

[Pas13] Rafael Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 334–354. Springer, Heidelberg, March 2013.

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 433–444. Springer, Heidelberg, August 1992.

[Wic13] Daniel Wichs. Barriers in cryptography with weak, correlated and leaky sources. In Robert D. Kleinberg, editor, *ITCS 2013*, pages 111–126. ACM, January 2013.

A Instantiation of Puncturable CCA-Secure PKE

Recall that the CCA-secure PKE of Dolev et al. [DDN00] works as follows.

- **Building blocks.**
  - CPA-secure PKE scheme $\mathsf{PKE}_{\mathrm{CPA}} = (\mathsf{Gen}_{\mathrm{CPA}}, \mathsf{Enc}_{\mathrm{CPA}}, \mathsf{Dec}_{\mathrm{CPA}})$.
  - One-time signature scheme $\Pi_{\mathrm{OTS}} = (\mathsf{Gen}_{\mathrm{OTS}}, \mathsf{Sign}_{\mathrm{OTS}}, \mathsf{Verify}_{\mathrm{OTS}})$. For simplicity, we assume that on security parameter 1^n, the key generation algorithm $\mathsf{Gen}_{\mathrm{OTS}}$ outputs a verification key of length n.

– Non-interactive zero-knowledge system $\mathsf{NIZK} = (\mathsf{Gen}_{\mathrm{NIZK}}, \mathsf{Prove}_{\mathrm{NIZK}}, \mathsf{Verify}_{\mathrm{NIZK}})$.

- **Key Generation** $\mathsf{Gen}(1^n)$**:**
  1. For each $i \in [n], b \in \{0,1\}$, run $(\mathsf{pk}_{i,b}, \mathsf{sk}_{i,b}) \leftarrow \mathsf{Gen}_{\mathrm{CPA}}(1^n)$.
  2. Run $\mathsf{crs} \leftarrow \mathsf{Gen}_{\mathrm{NIZK}}(1^n)$.
  3. Output $\mathsf{pk} := (\{\mathsf{pk}_{i,b}\}_{i\in[n],b\in\{0,1\}}, \mathsf{crs})$ and $\mathsf{sk} := (\mathsf{pk}, \{\mathsf{sk}_{i,b}\}_{i\in[n],b\in\{0,1\}})$.
- **Encryption** $\mathsf{Enc}(\mathsf{pk}, m)$**:**
  1. Run $(\mathsf{vk}_{\mathrm{OTS}}, \mathsf{sk}_{\mathrm{OTS}}) \leftarrow \mathsf{Gen}_{\mathrm{OTS}}(1^n)$, and parse $\mathsf{vk}_{\mathrm{OTS}}$ as $\mathsf{vk}_{\mathrm{OTS}} = (v_1, \ldots, v_n) \in \{0,1\}^n$.
  2. For each $i \in [n]$, run $\mathsf{ct}_i \leftarrow \mathsf{Enc}_{\mathrm{CPA}}(\mathsf{pk}_{i,v_i}, m)$.
  3. Use $\mathsf{Prove}_{\mathrm{NIZK}}$ with crs to obtain a proof π for the statement $\{\mathsf{pk}_{i,v_i}, \mathsf{ct}_i\}_{i\in[n]}$ that says "$\exists m'$ s.t. $\forall i \in [n]$, ct_i is an encryption of m' under pk_{i,v_i}.
  4. Run $\sigma \leftarrow \mathsf{Sign}_{\mathrm{OTS}}(\mathsf{sk}_{\mathrm{OTS}}, \mathsf{ct}_1 \| \cdots \| \mathsf{ct}_n \| \pi)$.
  5. Output $\mathsf{ct} := (\mathsf{ct}_1, \ldots, \mathsf{ct}_n, \pi, \mathsf{vk}_{\mathrm{OTS}}, \sigma)$
- **Decryption** $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$**:**
  1. Check whether $\mathsf{Verify}_{\mathrm{OTS}}(\mathsf{vk}_{\mathrm{OTS}}, \mathsf{ct}_1 \| \cdots \| \mathsf{ct}_n \| \pi, \sigma) = 1$. If not, abort.
  2. Check whether π is an accepting proof for the statement $\{\mathsf{pk}_{i,v_i}, \mathsf{ct}_i\}_{i\in[n]}$. If not, abort.
  3. Run $\tilde{m} \leftarrow \mathsf{Dec}_{\mathrm{CPA}}(\mathsf{sk}_{1,v_1}, \mathsf{ct}_1)$.
  4. Output $\tilde{m}$.

Now, let us define $(\mathsf{PuncGen}, \mathsf{PuncDec})$ as follows.

- **Punctured key generation** $\mathsf{PuncGen}(\mathsf{sk}, \mathsf{ct})$**:**
  1. Parse sk as $\mathsf{sk} = (\mathsf{pk}, \{\mathsf{sk}_{i,b}\}_{i\in[n],b\in\{0,1\}})$, parse ct as $\mathsf{ct} = (\mathsf{ct}_1, \ldots, \mathsf{ct}_n, \pi, \mathsf{vk}_{\mathrm{OTS}}, \sigma)$, and parse $\mathsf{vk}_{\mathrm{OTS}}$ as $\mathsf{vk}_{\mathrm{OTS}} = (v_1, \ldots, v_n)$.
  2. Output $\mathsf{sk}_{\{\mathsf{ct}\}} := (\mathsf{pk}, \mathsf{vk}_{\mathrm{OTS}}, \{\mathsf{sk}_{i,1-v_i}\}_{i\in[n]})$.
- **Decryption with punctured key** $\mathsf{PuncDec}(\mathsf{sk}_{\{\mathsf{ct}\}}, \mathsf{ct}')$**:**
  1. Parse $\mathsf{sk}_{\{\mathsf{ct}\}}$ as $\mathsf{sk}_{\{\mathsf{ct}\}} = (\mathsf{pk}, \mathsf{vk}_{\mathrm{OTS}}, \{\mathsf{sk}_{i,1-v_i}\}_{i\in[n]})$, parse ct' as $\mathsf{ct}' = (\mathsf{ct}'_1, \ldots, \mathsf{ct}'_n, \pi', \mathsf{vk}'_{\mathrm{OTS}}, \sigma')$, parse pk as $\mathsf{pk} = (\{\mathsf{pk}_{i,b}\}_{i\in[n],b\in\{0,1\}}, \mathsf{crs})$ parse $\mathsf{vk}_{\mathrm{OTS}}$ as $\mathsf{vk}_{\mathrm{OTS}} = (v_1, \ldots, v_n)$, and parse $\mathsf{vk}'_{\mathrm{OTS}}$ as $\mathsf{vk}'_{\mathrm{OTS}} = (v'_1, \ldots, v'_n)$.
  2. Check whether $\mathsf{vk}_{\mathrm{OTS}} \neq \mathsf{vk}'_{\mathrm{OTS}}$ and $\mathsf{Verify}_{\mathrm{OTS}}(\mathsf{vk}'_{\mathrm{OTS}}, \mathsf{ct}'_1 \| \cdots \| \mathsf{ct}'_n \| \pi', \sigma') = 1$. If not, abort.
  3. Check whether π' is an accepting proof for the statement $\{\mathsf{pk}_{i,v'_i}, \mathsf{ct}'_i\}_{i\in[n]}$. If not, abort.
  4. Find any $i^* \in [n]$ such that $v_{i^*} \neq v'_{i^*}$, and run $\tilde{m}' \leftarrow \mathsf{Dec}_{\mathrm{CPA}}(\mathsf{sk}_{i^*,v'_{i^*}}, \mathsf{ct}'_{i^*})$.
  5. Output $\tilde{m}'$.

To see the correctness of punctured keys, observe that (i) the strong unforgeability of Π_{OTS} guarantees that during the security game, after receiving a ciphertext $\mathsf{ct} = (\mathsf{ct}_1, \ldots, \mathsf{ct}_n, \pi, \mathsf{vk}_{\mathrm{OTS}}, \sigma)$ from the challenger, the adversary creates $\mathsf{ct}' = (\mathsf{ct}'_1, \ldots, \mathsf{ct}'_n, \pi', \mathsf{vk}'_{\mathrm{OTS}}, \sigma')$ such that $\mathsf{ct} \neq \mathsf{ct}'$, $\mathsf{vk}_{\mathrm{OTS}} = \mathsf{vk}'_{\mathrm{OTS}}$, and $\mathsf{Verify}_{\mathrm{OTS}}(\mathsf{vk}'_{\mathrm{OTS}}, \mathsf{ct}'_1 \| \cdots \| \mathsf{ct}'_n \| \pi', \sigma') = 1$ only with negligible probability, and (ii) the soundness of NIZK guarantees that during the security game, the probability that the adversary creates $\mathsf{ct}' = (\mathsf{ct}'_1, \ldots, \mathsf{ct}'_n, \pi', \mathsf{vk}'_{\mathrm{OTS}}, \sigma')$ such that π' is accepting but $\mathsf{Dec}_{\mathrm{CPA}}(\mathsf{sk}_{1,v'_1}, \mathsf{ct}'_1) \neq \mathsf{Dec}_{\mathrm{CPA}}(\mathsf{sk}_{i^*,v'_{i^*}}, \mathsf{ct}'_{i^*})$ is negligible. Regarding the security of punctured keys, it follows immediately from the CPA security of $\mathsf{PKE}_{\mathrm{CPA}}$ and the security of NIZK.

B Proof of Theorem 3

Theorem 5 (restatement of Theorem 3)**.** *Assume the existence of CCA-secure public-key encryption schemes. Then, there exists an NP language* $\mathbf{L}$ *such that if there exist*

- *a 2-round interactive argument* (P, V) *for* $\mathbf{L}$ *and*
- *an oblivious black-box reduction* R_{SWI} *for showing the strong WI of* (P, V) *based on a falsifiable assumption* (C, c)*,*

then the assumption (C, c) *is false.*

Proof. Let $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a CCA-secure PKE and $\mathbf{L}$ be the NP language that consists of all the public-key–ciphertext pairs of PKE such that either 0 or 1 is encrypted (the public key is not necessarily honestly generated), i.e.,

$$\mathbf{L} := \left\{(\mathsf{pk}, \mathsf{ct}) \;\middle|\; \exists b \in \{0,1\}, r \in \{0,1\}^{\mathsf{poly}(n)} \text{ s.t. } \mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, b; r) \right\} .$$

Assume, as stated in the statement of the theorem, the existence of a 2-round (non-delayed-input) interactive argument (P, V) and an oblivious black-box reduction R_{SWI} for showing the strong WI of (P, V) based on a falsifiable assumption (C, c). Let Q denote a polynomial such that for every verifier V^*, every $n \in \mathbb{N}$, every two joint distributions $\mathcal{D}^0_n = (\mathcal{X}^0_n, \mathcal{W}^0_n)$ and $\mathcal{D}^1_n = (\mathcal{X}^1_n, \mathcal{W}^1_n)$ over $\mathbf{R}_{\mathbf{L}} \cap (\{0,1\}^n \times \{0,1\}^*)$, and every $z \in \{0,1\}^*$, if it holds

$$\Pr\left[\langle P(w), V^*(z)\rangle(x) = b \;\middle|\; b \leftarrow \{0,1\};\ (x, w) \leftarrow (\mathcal{X}^b_n, \mathcal{W}^b_n) \right] \geq \frac{3}{4} ,$$

then either (i) $R^{V^*_z, \mathcal{D}^0_n, \mathcal{D}^1_n}_{\mathrm{SWI}}(1^n)$ breaks the assumption (C, c) on n with advantage $1/Q(n)$ or (ii) $R^{V^*_z, \mathcal{D}^0_n, \mathcal{D}^1_n}_{\mathrm{SWI}}(1^n)$ distinguishes $\mathcal{X}^0_n$ and $\mathcal{X}^1_n$ with advantage $1/Q(n)$.[21] (Such a polynomial is guaranteed to exist because of our assumption on R_{SWI}.)

At a high level, the proof proceeds as outlined in Section 2.2. Specifically, we first define a cheating verifier V^*_{SWI} against the strong WI of (P, V). Then, we proceed with case analysis about the behavior of $R^{V^*_{\mathrm{SWI}}}_{\mathrm{SWI}}$, where in the first case, we show that we can efficiently break the assumption (C, c) by using R_{SWI}, and in the second case, we show that we can efficiently break the soundness of (P, V) by using R_{SWI}.

We first introduce distributions over $\mathbf{R}_{\mathbf{L}}$ and a verifier against the strong WI of (P, V). For any $n \in \mathbb{N}$, let Keys_n be the set of all the keys that can be output by $\mathsf{Gen}(1^n)$, i.e., $\mathsf{Keys}_n := \{(\mathsf{pk}, \mathsf{sk}) \mid \exists r \in \{0,1\}^* \text{s.t. } (\mathsf{pk}, \mathsf{sk}) = \mathsf{Gen}(1^n; r)\}$. Then, for any $n \in \mathbb{N}$ and any $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Keys}_n$, let $\mathcal{D}^0_{\mathsf{pk}}$ and $\mathcal{D}^1_{\mathsf{pk}}$ be the distributions that are defined over $\mathbf{R}_{\mathbf{L}}$ as follows: $\forall b \in \{0,1\}$,

$$\mathcal{D}^b_{\mathsf{pk}} := \left\{((\mathsf{pk}, \mathsf{ct}), (b, r)) \;\middle|\; r \leftarrow \{0,1\}^{\mathsf{poly}(n)};\ \mathsf{ct} := \mathsf{Enc}(\mathsf{pk}, b; r) \right\} ,$$

i.e., the first part of $\mathcal{D}^b_{\mathsf{pk}}$ outputs pk and a random encryption of b, and the second part outputs b and the randomness of the encryption. We use $(\mathcal{X}^b_{\mathsf{pk}}, \mathcal{W}^b_{\mathsf{pk}})$ to denote the joint distributions such that $\mathcal{X}^b_{\mathsf{pk}}$ denotes the first part of $\mathcal{D}^b_{\mathsf{pk}}$ and $\mathcal{W}^b_{\mathsf{pk}}$ denotes the second part of $\mathcal{D}^b_{\mathsf{pk}}$. Next, for any $n \in \mathbb{N}$ and $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Keys}_n$, let $V^*_{\mathrm{SWI}}[n, \mathsf{pk}, \mathsf{sk}]$ be the verifier described in Algorithm 3. Note that due to the completeness of (P, V) and the correctness of PKE, our verifier $V^*_{\mathrm{SWI}}[n, \mathsf{pk}, \mathsf{sk}]$ distinguishes $\mathcal{D}^0_{\mathsf{pk}}$ and $\mathcal{D}^1_{\mathsf{pk}}$ with probability $1 - \mathsf{negl}(n) > 3/4$. In the following, we usually write $V^*_{\mathrm{SWI}}[n, \mathsf{pk}, \mathsf{sk}]$ as V^*_{SWI} for editorial simplicity.

We proceed with case analysis about the behavior of the strong WI reduction R_{SWI} in the setting where R_{SWI} is combined with our strong WI verifier V^*_{SWI}. Specifically, we consider the following two cases.

- **Case 1.** There exists a polynomial poly such that for infinitely many $n \in \mathbb{N}$, there exists $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Keys}_n$ such that $R^{V^*_{\mathrm{SWI}}}_{\mathrm{SWI}}(1^n)$ breaks the assumption (C, c) on n with advantage $1/\mathsf{poly}(n)$, i.e.,

$$\Pr\left[\langle R^{V^*_{\mathrm{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}_{\mathrm{SWI}}, C\rangle(1^n) = 1\right] \geq c + \frac{1}{\mathsf{poly}(n)} . \tag{19}$$

- **Case 2.** The condition of Case 1 does not hold.

We analyze each case below.

[21] Formally, we also need to give the reduction R_{SWI} an input 1^4 to let it know that the advantage of V^*_z is $1/4$ (see Definition 10). We omit it in this proof for editorial simplicity.

Algorithm 3 Strong WI verifier $V^*_{\mathrm{SWI}}[n, \mathsf{pk}, \mathsf{sk}]$.

1. On input 1^n, sample a key key for a pseudorandom function PRF. In the following, whenever new randomness is required, it is obtained by applying $\mathsf{PRF}(\mathsf{key}, \cdot)$ on the transcript that is exchanged with the prover so far (including the statement).
2. Invoke the honest verifier algorithm V of (P, V) and let it interact with the external prover. Let $x^\star = (\mathsf{pk}^\star, \mathsf{ct}^\star)$ denote the statement that is obtained at the beginning of the interaction and $\mathsf{out}^\star$ denote the output of V. If $\mathsf{pk}^\star \neq \mathsf{pk}$, output a random bit and abort.
3. Output a random bit and abort if $\mathsf{out}^\star = 0$. Otherwise, run $b \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$ and output b.

Analysis of Case 1. We show that R_{SWI} can be used to break the assumption (C, c). Fix any poly, n, and $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Keys}_n$ such that we have (19). Consider the following adversary $\mathcal{A}$ against (C, c).

1. Given $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Keys}_n$ as auxiliary inputs, $\mathcal{A}$ lets $R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}(1^n)$ interact with the challenger C, where sk is used to emulate V^*_{SWI} for R_{SWI} efficiently.

Clearly, $\mathcal{A}$ runs in polynomial time. Also, from (19) it follows immediately that $\mathcal{A}$ breaks the assumption (C, c) on n with advantage $1/\mathsf{poly}(n)$. We thus conclude that the assumption (C, c) is false in this case.

Analysis of Case 2. We show that R_{SWI} can be used to break the soundness of (P, V) (unless it can be used to break the CCA security of PKE). Toward this end, we split Case 2 into two sub-cases based on the behavior of R_{SWI} in the setting where $R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}}$ is used as a distinguisher against $\mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}$ for randomly chosen $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$. Let us first introduce the following notations about $(\mathsf{pk}, \mathsf{sk})$ of PKE. For any n and $(\mathsf{pk}, \mathsf{sk})$:

- $(\mathsf{pk}, \mathsf{sk})$ is called *interesting (w.r.t. n)* if it satisfies the following.

$$\Pr\left[\langle P(w), V^*_{\mathrm{SWI}}\rangle(x) = b \;\middle|\; b \leftarrow \{0,1\};\ (x, w) \leftarrow \mathcal{D}^b_{\mathsf{pk}} \right] \geq \frac{3}{4} \ . \tag{20}$$

  Intuitively, $(\mathsf{pk}, \mathsf{sk})$ is interesting if $V^*_{\mathrm{SWI}}[n, \mathsf{pk}, \mathsf{sk}]$ breaks the strong WI of (P, V) w.r.t. $\mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}$ with high advantage (which implies that R_{SWI} either breaks (C, c) or distinguishes $\mathcal{X}^0_{\mathsf{pk}}$ and $\mathcal{X}^1_{\mathsf{pk}}$ given V^*_{SWI}).

- $(\mathsf{pk}, \mathsf{sk})$ is called *type-1 interesting* if it is interesting and in addition satisfies the following.

$$\Pr\left[\mathsf{INTERESTING\text{-}QUERY} \;\middle|\; \begin{array}{l} b \leftarrow \{0,1\};\ (x, w) \leftarrow \mathcal{D}^b_{\mathsf{pk}} \\ b' \leftarrow R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}(1^n, x) \end{array}\right] \leq \frac{1}{2Q(n)} \ , \tag{21}$$

  where (i) Q is the polynomial that is introduced at the beginning of the proof and (ii) $\mathsf{INTERESTING\text{-}QUERY}$ is the event that is defined as follows: through oracle queries to V^*_{SWI}, the reduction $R_{\mathrm{SWI}}(1^n, x)$ invokes an execution of (P, V) in which R_{SWI} forwards the statement x to V^*_{SWI} along with an accepting prover message (i.e., a message such that we have $\mathsf{out}^\star = 1$ in V^*_{SWI}). Thus, intuitively, $(\mathsf{pk}, \mathsf{sk})$ is type-1 interesting if R_{SWI} can either break (C, c) or distinguish $\mathcal{X}^0_{\mathsf{pk}}$ and $\mathcal{X}^1_{\mathsf{pk}}$ without producing an accepting prover message for the statement x.

- $(\mathsf{pk}, \mathsf{sk})$ is called *type-2 interesting* if it is interesting but is not type-1 interesting.

Now, we consider the following two sub-cases.

- **Case 2-1.** There exists a negligible function negl such that for every $n \in \mathbb{N}$,

$$\Pr\left[(\mathsf{pk}, \mathsf{sk}) \text{ is type-1 interesting} \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)\right] \geq 1 - \mathsf{negl}(n) \ . \tag{22}$$

- **Case 2-2.** The condition of Case 1 does not hold.

We analyze each sub-case below.

Analysis of Case 2-1. We show that R_{SWI} can be used to break the CCA security of PKE. We note that for every sufficiently large $n \in \mathbb{N}$, (i) we have (22) and (ii) for every $(\mathsf{pk}, \mathsf{sk}) \in \mathsf{Keys}_n$, we have

$$\Pr\left[\langle R_{\mathsf{SWI}}^{V^*_{\mathsf{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}, C\rangle(1^n) = 1\right] = c + \mathsf{negl}(n) \ . \tag{23}$$

(This is because it is assumed that the condition of Case 1 does not hold.) Fix any such n. Then, consider the following adversary $\mathcal{A}_{\mathrm{CCA}}$ against the CCA security of PKE.

1. On input $(1^n, \mathsf{pk}, z)$, the adversary $\mathcal{A}_{\mathrm{CCA}}$ sends $m_0 := 0$ and $m_1 := 1$ to the challenger as the challenge plaintexts.
2. On receiving the challenge ciphertext ct, the adversary $\mathcal{A}_{\mathrm{CCA}}$ lets $x^\star := (\mathsf{pk}, \mathsf{ct})$ and runs $b^\star \leftarrow R_{\mathsf{SWI}}^{V^*_{\mathsf{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}(1^n, x^\star)$, where the decryption oracle $\mathsf{Dec}(\mathsf{sk}, \cdot)$ is used to emulate V^*_{SWI} efficiently without knowing sk. If $\mathsf{INTERESTING\text{-}QUERY}$ occurs during the execution of R_{SWI}, the adversary $\mathcal{A}_{\mathrm{CCA}}$ outputs a random bit. Otherwise, it outputs $b^\star$. (Note that $\mathcal{A}_{\mathrm{CCA}}$ needs to query ct to $\mathsf{Dec}(\mathsf{sk}, \cdot)$ only when $\mathsf{INTERESTING\text{-}QUERY}$ occurs.)

We now analyze $\mathcal{A}_{\mathrm{CCA}}$. Note that when the challenger samples a type-1 interesting $(\mathsf{pk}, \mathsf{sk})$, we have (20), and thus, by combining it with (23), we have that $R_{\mathsf{SWI}}^{V^*_{\mathsf{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}$ distinguishes $\mathcal{X}^0_{\mathsf{pk}}$ and $\mathcal{X}^1_{\mathsf{pk}}$ with advantage $1/Q(n)$ due to the definition of Q. Thus, by additionally using (22) and (21) and recalling the definitions of $\mathcal{X}^0_{\mathsf{pk}}$ and $\mathcal{X}^1_{\mathsf{pk}}$ (i.e., that $\mathcal{X}^b_{\mathsf{pk}}$ outputs pk and a random encryption of b), we conclude that $\mathcal{A}_{\mathrm{CCA}}$ wins with advantage at least

$$\begin{aligned}
&\left(\frac{1}{Q(n)} - \Pr\left[\begin{array}{l}\mathsf{INTERESTING\text{-}QUERY} \text{ occurs in Step 2 of } \mathcal{A}_{\mathrm{CCA}} \\ \text{when } (\mathsf{pk}, \mathsf{sk}) \text{ is type-1 interesting}\end{array}\right]\right) \times \Pr\left[(\mathsf{pk}, \mathsf{sk}) \text{ is type-1 interesting}\right] \\
&- \Pr\left[(\mathsf{pk}, \mathsf{sk}) \text{ is not type-1 interesting}\right] \\
\geq &\frac{1}{2Q(n)} \times (1 - \mathsf{negl}(n)) - \mathsf{negl}(n) = \frac{1}{\mathsf{poly}(n)} \ .
\end{aligned}$$

Thus, we obtain a contradiction.

Analysis of Case 2-2. We show that R_{SWI} can be used to break the soundness of (P, V). For each $n \in \mathbb{N}$, let $(\mathcal{X}_n, \mathcal{W}_n)$ and $(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)$ be the following joint distributions.

$$(\mathcal{X}_n, \mathcal{W}_n) := \left\{((\mathsf{pk}, \mathsf{ct}), (b, r)) \;\middle|\; (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n);\ b \leftarrow \{0,1\};\ r \leftarrow \{0,1\}^{\mathsf{poly}(n)};\ \mathsf{ct} := \mathsf{Enc}(\mathsf{pk}, b; r)\right\} \ .$$
$$(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n) := \left\{((\mathsf{pk}, \mathsf{ct}), \mathsf{sk}) \;\middle|\; (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n);\ \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, 2)\right\} \ .$$

(Note that $(\mathcal{X}_n, \mathcal{W}_n)$ ranges over $\mathbf{R}_{\mathbf{L}} \cap (\{0,1\}^n \times \{0,1\}^*)$ and $(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}}_n)$ ranges over $(\{0,1\}^n \setminus \mathbf{L}) \times \{0,1\}^*$.[22] Also, note that $(\mathcal{X}_n, \mathcal{W}_n)$ is identically distributed with $\{(x, w) \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n); b \leftarrow \{0,1\}; (x, w) \leftarrow \mathcal{D}^b_{\mathsf{pk}}\}$.)

We first describe a cheating prover P^*. At a high level, given $x = (\mathsf{pk}, \mathsf{ct})$ as the statement, P^* first internally runs $R_{\mathsf{SWI}}^{V^*_{\mathsf{SWI}}}$ as a distinguisher for $\mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}$ while hoping that the event $\mathsf{INTERESTING\text{-}QUERY}$ occurs in a randomly selected query. (Recall that $\mathsf{INTERESTING\text{-}QUERY}$ occurs when the reduction R_{SWI}, on input a statement x, makes a query that contains x and an accepting prover message.) If indeed $\mathsf{INTERESTING\text{-}QUERY}$ occurs in the selected query, P^* simply forwards the prover message that is contained in the query to the external verifier. A problem is that P^* needs to forward the message m_V that it receives from the external prover to the internally emulated R_{SWI}, and in this case, P^* cannot verify whether R_{SWI}'s query contains an accepting prover message. Thus, to emulate V^*_{SWI}'s response for R_{SWI} until the randomly selected query is made, P^* simply guesses that only rejecting prover messages are queried about m_V until the randomly selected one is made, and therefore P^* simply returns a random bit until the randomly selected query is made. The formal description of P^* is given in Algorithm 4.

Now, we analyze the success probability of P^* as follows. Since it is assumed that the condition of Case 2-1 does not hold, there exists a polynomial poly such that for infinitely many $n \in \mathbb{N}$,

$$\Pr\left[(\mathsf{pk}, \mathsf{sk}) \text{ is type-1 interesting} \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)\right] < 1 - \frac{1}{\mathsf{poly}(n)} \ . \tag{24}$$

[22] We assume without loss of generality that on security parameter 1^n, Gen and Enc yield $(\mathsf{pk}, \mathsf{ct})$ such that $|(\mathsf{pk}, \mathsf{ct})| = n$.

Algorithm 4 Cheating Prover P^*.

Input: (x, z_x), which is sampled from $(\overline{\mathcal{X}}_n, \overline{\mathcal{Z}_n})$.

1. Obtain a verifier message m_V from the external verifier.
2. Sample random $i^\star \in [T_R]$, where T_R is an upper bound of the number of queries that R_{SWI} makes to V^*_{SWI}. Then, parse (x, z_x) as $((\mathsf{pk}, \mathsf{ct}), \mathsf{sk})$ and run $R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}(1^n, x)$ as a distinguisher for $\mathcal{D}^0_{\mathsf{pk}}$ and $\mathcal{D}^1_{\mathsf{pk}}$, where each query that R_{SWI} makes to V^*_{SWI} is answered as follows.
   - If the statement that is contained in the query is different from x, emulate the response of V^*_{SWI} perfectly by using sk.
   - If the statement that is contained in the query is x, return m_V as the verifier first-round message, and after receiving a prover response in a subsequent query, return a random bit unless it is the $i^\star$-th query. If it is the $i^\star$-th query, define m_P to be the prover response that is contained in the query, and abort the execution of R_{SWI}.
3. If m_P is not defined in the above step, abort. Otherwise, send m_P to the external verifier.

Fix any such n. Note that by the completeness of (P, V) and the correctness of PKE, we have

$$\Pr\left[(\mathsf{pk}, \mathsf{sk}) \text{ is interesting} \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)\right] = 1 - \mathsf{negl}(n) \ . \tag{25}$$

By combing (24) and (25), we obtain

$$\Pr\left[(\mathsf{pk}, \mathsf{sk}) \text{ is type-2 interesting} \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)\right] \geq \frac{1}{2\mathsf{poly}(n)} \ . \tag{26}$$

By combining (26) with the definition of type-2 interesting keys, we obtain

$$\Pr\left[\mathsf{INTERESTING\text{-}QUERY} \;\middle|\; \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n);\ b \leftarrow \{0,1\};\ (x, w) \leftarrow \mathcal{D}^b_{\mathsf{pk}} \\ b' \leftarrow R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}(1^n, x) \end{array}\right] \geq \frac{1}{4Q(n)\mathsf{poly}(n)} \ .$$

By additionally using the security of the CCA-secure PKE, we obtain

$$\Pr\left[\mathsf{INTERESTING\text{-}QUERY} \;\middle|\; \begin{array}{l} (x, z_x) \leftarrow (\overline{\mathcal{X}}_n, \overline{\mathcal{Z}_n}) \\ b' \leftarrow R_{\mathrm{SWI}}^{V^*_{\mathrm{SWI}}, \mathcal{D}^0_{\mathsf{pk}}, \mathcal{D}^1_{\mathsf{pk}}}(1^n, x) \end{array}\right] \geq \frac{1}{4Q(n)\mathsf{poly}(n)} - \mathsf{negl}(n) \ . \tag{27}$$

(To see this, observe that we can efficiently emulate V^*_{SWI} by using the decryption oracle in the CCA-security game until $\mathsf{INTERESTING\text{-}QUERY}$ occurs.) Now, note that P^* convinces the external verifier if during the execution of P^*, the event $\mathsf{INTERESTING\text{-}QUERY}$ occurs for the first time in the $i^\star$-th query. Thus, by (27) and the security of PRF in the first step of V^*_{SWI},[23] [24] we obtain

$$\Pr\left[\langle P^*(z_x), V\rangle(x) = 1 \;\middle|\; (x, z_x) \leftarrow (\overline{\mathcal{X}}_n, \overline{\mathcal{Z}_n})\right] \geq \frac{1}{\mathsf{poly}'(n)}$$

for some polynomial $\mathsf{poly}'(n)$. This in particular means that there exists $x \in \{0,1\}^n \setminus \mathbf{L}$ and $z \in \{0,1\}^*$ such that

$$\Pr\left[\langle P^*(z), V\rangle(x) = 1\right] \geq \frac{1}{\mathsf{poly}'(n)} \ .$$

Thus, P^* breaks the soundness of (P, V), and therefore we obtain a contradiction.

Completing the proof of Theorem 3. Combining the analyses of Case 1 and Case 2, we conclude that the assumption (C, c) is false. This completes the proof of Theorem 3. □

[23] The security of PRF is used to argue that the probability of $\mathsf{INTERESTING\text{-}QUERY}$ occurring in (27) decreases only negligibly when V^*_{SWI} uses true randomness.

[24] For simplicity, we implicitly assume that R_{SWI} requests V^*_{SWI} to reuse the same randomness in all the queries (cf. the conventions stated in Section 3.4.2). The general case can be handled straightforwardly by considering slightly more complex P^*.