Doctoral Thesis

A Study of Non-Interactive Zero-Knowledge Proof Systems in a Black-Box Framework

Supervisor: Masayuki Abe & Mehdi Tibouchi

Department of Social Informatics
Graduate School of Informatics
Kyoto University
Japan

Kyosuke Yamashita

A Study of Non-Interactive Zero-Knowledge Proof Systems in a Black-Box Framework

Kyosuke Yamashita

Abstract

This dissertation studies the limitation of *non-interactive zero-knowledge proof systems* (NIZKs) in a *black-box* framework.

A *black-box construction* is one of the most well-known methodology to investigate the essential characteristics of cryptographic primitives. In this methodology, we are given an oracle that implements a cryptographic primitive, and consider if we can construct another primitive by using the oracle. Because an underlying primitive is given as an oracle, if there is a black-box construction of a primitive Q based on a primitive P, it means that we can achieve Q from P no matter how P is implemented. However, if there is no such a black-box construction, it indicates that there is no universal "compiler" that compiles P to Q, and thus we might have to rely on algebraic structures to construct Q based on P.

An NIZK is a cryptographic primitive between two parties a *prover* and a *verifier* who share a statement of an NP-language. The prover, who is the only one possessing a secret (or a witness), convinces the verifier that he knows the secret by sending a single message. NIZKs are fundamental building blocks in cryptography, and are used in many applications such as electronic voting systems and blockchains. So far, NIZKs have been developed for each assumption such as the existence of a trapdoor permutation or pairing.

This dissertation investigates the limitation of NIZKs by treating them as oracles. When we use an NIZK as a building block in a certain application, it is often required that the NIZK supports a specific language that is convenient for the construction, which typically contains a conjunctive/disjunctive relation. Furthermore, we often attempt to extend a language that an NIZK proves, if we want to construct an NIZK for an NP-complete language. We consider the problem of extending a language that an NIZK proves to a language that includes a conjunctive/disjunctive relation in a black-box manner, and show negative results on it.

In the first result, we simplify the NIZK oracle in the existing black-box framework, and demonstrate that such a simplification does not affect the capability of the framework. An involved oracle in a black-box framework may cause complicated proofs. Therefore, the newly proposed oracle makes the framework easy to use.

In the second result, we consider the problem of constructing an NIZK that proves the secret equality behind two distinct NP statements, which is useful in theory and practice. Namely, we show that, given NIZKs for languages $\mathcal{L}$ and $\mathcal{L}'$, it is impossible to construct an NIZK that proves the equality of secrets of $\mathcal{L}$ and $\mathcal{L}'$ in a black-box manner. This result indicates that standard NIZKs are not sufficient to construct an NIZK for the secret equality.

In the final result, we investigate the limitation of an NIZK that employs the *commit-and-prove* methodology (CP-NIZK). CP-NIZKs are used in real world applications such as cryptocurrency, and it is known that we can construct an NIZK for the secret equality based on CP-NIZKs. However, we show that, even if we are given CP-NIZKs for certain languages, it is impossible to construct an NIZK for a language that includes a disjunctive relation (i.e., the OR relation) in a black-box manner. Therefore, we conclude that we should rely on algebraic structures if we want to enhance the power of NIZKs in terms of languages that they can prove.

These results indicate the hardness of using NIZKs in an abstract way. We conjecture that this is because the capability of an NIZK is determined by the assumption that the NIZK is based on, and the language that the NIZK proves. Therefore, NIZKs will develop for each assumption. In particular, if a new assumption is proposed in the future, we should construct an NIZK by exploiting the characteristic of the assumption.

Acknowledgements

First and foremost, the author would like to show his greatest appreciation to his supervisors, Professor Masayuki Abe and Professor Mehdi Tibouchi. None of the results in this dissertation was accomplished without their helpful and patient supports. The author would like to thank to his advisors, Professor Akinori Kawachi, Professor Masatoshi Yoshikawa and Professor Shigeo Matsubara for giving insightful comments. The author is grateful to Dr. Goichiro Hanaoka and Dr. Takahiro Matsuda, who gave him an opportunity to work in National Institute of Advanced Industrial Science and Technology as a research assistant.

Contents

Chapter 1

Introduction

1.1 Non-Interactive Zero-Knowledge Proof System

In our society, there are often conflicts between identity verification and privacy protection. For instance, when voting, voters should be authenticated, but individual must not be identified, and when one buys something with age limit, his age should be certificated, but it is not necessary to reveal his exact age. The same situation happens in information systems. There is a case that one wants to authenticate himself to use an online service by sending his password, but does not want to reveal the password itself. In either case, a sender, who possesses a private information, certifies it without disclosing the secret itself.

A zero-knowledge proof system (ZK) [1] and a non-interactive zero-knowledge proof system (NIZK) [2] are cryptographic primitives that realize the above mentioned requirements. They constitute two parties a *prover* and a *verifier* who share a certain problem (i.e., a statement of an NP-language). The prover in ZK, who is the only one possessing a secret (or a witness), proves his knowledge about the secret with interacting the verifier. The security requirements of a ZK are that a malicious prover cannot cheat a verifier on an invalid statement (which is called the *soundness*), and the interaction does not leak anything about the secret apart from the prover's knowledge about the secret (which is called the *zero-knowledge property*).

An NIZK is a variant of ZKs which requires only a single message from a prover. As the proof is done by a single message, NIZKs are considered more efficient and practical than ZKs. In the aforementioned scenario, the proof can be completed even if it is difficult for a sender (i.e., a prover) and a verifier to keep communicating online. That is, once the prover sends a proof, he can be offline and wait for the the result. Therefore, NIZKs could provide us more efficient and flexible applications.

The construction of an efficient NIZK was a long-standing open problem. However, Groth and Sahai [3] and Groth *et al.* [4] proposed a pairing-based technique to

construct an efficient and practical NIZK. Recently, more efficient NIZKs are proposed [5, 6], and are used in electronic votings [7, 8, 9], electronic auction [10, 11, 12], and cryptocurrencies such as ZCash [13] and Ethereum [14]. We introduce the following two examples as concrete applications of NIZKs.

Example 1: Contingent Payment One of the most notable applications of NIZKs in blockchain is the contingent payment [15, 16, 17], which follows the above scenario [17]: Alice wants to know the answer of a sudoku puzzle, and she broadcasts a message that she will pay whoever provides her the answer. Bob knows the answer, and wants to sell the solution. However, there is a problem; they do not trust each other. This problem can be resolved by using an NIZK as follows: First, Bob encrypts the solution m by a symmetric key encryption scheme to obtain a ciphertext $c = \mathsf{Enc}(sk, m)$ where sk is a symmetric key, and randomize the secret key by using a hash function H to obtain $h = H(sk)$. Then, Bob sends a proof that shows "the plaintext of c is the solution m" $\wedge$ "the preimage of h is the secret key that is used to obtain c." If Alice is convinced by the proof, then she sends a transaction that says she will pay whoever provides her the secret key sk to blockchain. In this scenario, the NIZK plays essential role to realize the fair trade between Alice and Bob. In fact, the NIZK proves the following language that shows the equality of secrets:

$$\mathcal{L} = \{c, h \,|\, \exists m, sk, sk' \text{ s.t. } c = \mathsf{Enc}(sk, m) \wedge h = H(sk') \wedge sk = sk'\}.$$

Example 2: Electronic Voting Another important example is an electronic voting that uses an NIZK and a homomorphic encryption scheme. A homomorphic encryption scheme is an encryption scheme that allows a computation on encrypted data, such as $\mathsf{Enc}(m) + \mathsf{Enc}(m') = \mathsf{Enc}(m + m')$ (note that an encryption scheme should take a key, but we here omit it for simplicity). Suppose that we are having a majority voting, where a voter sends 1 if yes, 0 otherwise, with encrypting the value by the encryption scheme. The voting organizer can calculate the some of votes due to the homomorphic property, but cannot detect whether a voter votes 0 or 1 due to the security of the encryption scheme. However, if a malicious voter encrypts, say, -100 and send the ciphertext, then it obviously cancels 100 yes votes. Therefore, it should be guaranteed that a vote is either 0 or 1, and an NIZK for the following language could be used for this purpose:

$$\mathcal{L} = \{c \,|\, c = \mathsf{Enc}(0) \vee \mathsf{Enc}(1)\}.$$

Comparison with Other Methodologies

We introduce existing methodologies that could substitute NIZKs, and discuss the superiority of NIZKs. One possible solution for the conflict between authentication and

privacy is a multi-party computation (MPC). An MPC is a cryptographic primitive where multiple parties possess their own secrets, and compute something (such as the summation of the secrets) based on the secrets without revealing them. This could provide a good solution for electronic votings and electronic auctions. However, it is often the case that an MPC becomes complicated one. That is, malicious parties could collude each other, and manipulate the computational result. Therefore, we should model such an involved adversary to prove the security, which is a cumbersome task. We note that, assuming the absence of adversaries might make the situation simpler, but it is far from realistic.

Recently, trusted execution environment (TEE) is gathering attention, such as Intel SGX [18] and Arm TrustZone [19]. TEE offers an isolated environment that an important process runs independent of other process, and guarantees the correct computation. As TEE guarantees secure computations and forces honest behavior, it provides many cryptographic applications. However, the problem is that, we must trust the hardware suppliers. Therefore, if we incorporate these hardwares into an information system, we should construct the system to be less dependent on this device.

Compared with these technologies, NIZKs provide simpler models and theoretical security. Note that NIZKs are modeled in several ways. It is known that we cannot construct an NIZK for a non-trivial language in the standard model [20]. Therefore, several models have been proposed such as the common reference string (CRS) model [21], the hidden bit model [22], and the random oracle model [23, 24]. The CRS model is a model that a prover and a verifier have access to a common reference string (CRS) generated by a trusted third party, and the proof and the verification are done by using the CRS. It is known that we can construct NIZKs for NP-complete language in the CRS model [4, 5, 25, 26, 27], where all constructions are based on certain assumptions. The CRS model is the most practical one, while it requires a trusted third party. As pointed out by Bellare and Naor [25], it is necessary to use a doubly enhanced trapdoor permutation to realize the hidden bit model, which is a slightly stronger primitive. When we implement an NIZK in the random oracle model, we should use a hash function instead of a random oracle. However, we do not know if there exists a hash function that perfectly simulates a random oracle (note that the existence of such a hash function immediately indicates $\mathrm{P} \neq \mathrm{NP}$). Therefore, the CRS model is the most widely used model. Note that the CRS model can be classified in terms of the way of the CRS generation such as the preprocessing model [28], the designated verifier/ prover model [29], the quasi-adaptive NIZK [30], the bare public-key model [31, 32], the helper model [33], the multi-string model [34], and the subversion resilience NIZK [35]. While there are variety of NIZKs, we only treat NIZKs in the standard CRS model in this dissertation.

We argue that, by using an NIZK, we can prove the security of an information system easily, and reduce the dependence of the security on specific hardware. Suppose that we want to construct an NIZK that a CRS is reusable (i.e., a multi-theorem

NIZK). In such a construction, the CRS generation is a once-for-all task. Therefore, we can use MPCs or TEE to generate a CRS, and after that we can prove something by relying only on the property of an NIZK whose security is theoretically guaranteed. Therefore, NIZKs could provide a better solution for the conflict between authentication and privacy, while we cannot eliminate the dependence on a trusted third party.

1.2 Black-Box Construction

One of the central goal in theoretical cryptography is to uncover relationships between cryptographic primitives. The seminal work by Impagliazzo and Rudich [36] formalized the problem of constructing a "high-level" primitive with assuming the existence of a "low-level" primitive. That is, if there exists a "compiler" that compiles a primitive P to another primitive Q, it is said that there exists a *black-box construction* of Q based on P. In a black-box construction, an underlying primitive P is given as an oracle, which satisfies syntactical and security definitions of P. Therefore, a black-box construction of Q from P indicates that P is sufficient for Q regardless of the assumption that P is based on (e.g., integer factoring or lattice based assumption). This line of research has been successful and we know many positive results for black-box constructions. For instance, it is known that a one-way function can be used to construct the following primitives: a signature scheme [37], a pseudorandom generator [38], and a commitment scheme [39] (in fact, these primitives are equivalent to a one-way function). Further, more involved primitives have been studied in a black-box manner. For instance, a trapdoor permutation can be used to construct a public key encryption scheme [40], or a private information retrieval [41].

On the other hands, showing the absence of a black-box construction is also an important direction. We can expand our knowledge of the conditions for a primitive to exist if we show the impossibility of a black-box construction of the primitive based on another primitive. The work [36] showed that there is no (fully) black-box construction of a key agreement protocol based on a one-way functions. They demonstrated an adversary in the random oracle model that breaks the security of a key agreement protocol with polynomial many queries to the random oracle. This is sufficient to separate a one-way function and a key agreement protocol, as a one-way function exists in the random oracle model. This line of research has been successful, and a large number of follow-up works are seen in the literature. For instance, we know that the following black-box impossibilities; a collision resistant hash function cannot be based on a one-way permutation [42], an oblivious transfer cannot be based on a public key encryption scheme (and vice versa) [43], and a virtual black-box obfuscation cannot be based on a one-way function [44].

In fact, there are many types of black-box constructions and techniques to show

the impossibility of black-box constructions. In Chapter 2.2, we present these taxonomy and make a short survey of them.

We remark that the black-box researches helps us to understand what the world we live in is. We now believe that public key primitives are separated from symmetric key primitives due to the result in [36]. In fact, Impagliazzo [45] conjectured that the world we live in is one of the following:

Algorithmica: P = NP.

Heuristica: NP-problems are hard in worst case, but easy on average.

Pessiland: NP-problems are hard on average, but one-way functions do not exist.

Minicrypt: One-way functions exist.

Cryptomania: Public key cryptography is possible.

We cannot deny the possibility that we live in Cryptomania, as long as we believe that public key encryption schemes exist. However, we do not know even one-way functions exist (note that the existence of a one-way function implies $P \neq NP$). Therefore, the black-box research could help us to understand what the world should be.

1.3 Related Work

We introduce previous works that deeply relate to the results of this dissertation. Overall, they treat NIZKs in a black-box manner.

We have mentioned the black-box researches thus far in this dissertation. However, another line of research has been developed. That is, the construction of primitives that uses the internal structure of underlying primitives, such as algebraic structures, named non-black-box construction. Black-box construction says nothing about non-black-box techniques. For instance, it is known that a chosen-ciphertext attack secure public key encryption scheme (CCA-PKE) cannot be based on a chosen-piaintext attack secure public key encryption scheme (CPA-PKE) in a black-box manner (in slightly restricted model) [46], we can realize such a construction if we employ non-black-box techniques [47, 48]. In particular, every efficient construction of an NIZK for an NP-complete language is constructed in a non-black-box manner.

The Augmented Black-Box Framework

Brakerski *et al.* [49] proposed a black-box framework that encompasses the power of non-black-box techniques. That is, they initiated the work of utilizing NIZKs in a black-box framework, named the augmented black-box framework. They introduced

an oracle that implements an NIZK for an NP-complete language, and showed following results: It is possible to construct a CCA-PKE based on a CPA-PKE, and it is impossible to construct a key-agreement scheme based on a one-way function. Note that it is not surprising we can construct a CCA-PKE in a black-box manner because the NIZK oracle proves an NP-complete language. In chapter 3, we simplify their oracle so that it becomes easy to use.

Black-Box Language Extension

One of the ultimate goal in the construction of NIZKs is to deal with an NP-complete language. When an NIZK is proposed based on some new assumption, it is often the case that it proves only a restricted language. Then, cryptographers consider how to expand the language that the NIZK proves, or reductions to NP-complete languages. Further, the expansion is done by combining languages by a binary operator $\diamond \in \{\wedge, \vee\}$. However, we do not know whether such a language expansion can be done in general.

Abe *et al.* [50] initiated the study of a black-box language extension. They showed that, given simulation-sound NIZKs (SS-NIZKs) [51] for a language $\mathcal{L}$, it is impossible to construct a (standard) NIZK for $\mathcal{L} \vee \mathcal{L}'$, where $\mathcal{L}'$ is some NP language. (Note that, given NIZKs for $\mathcal{L}$ and $\mathcal{L}'$, we can trivially construct a standard NIZK for $\mathcal{L} \wedge \mathcal{L}'$ by executing the given NIZKs in parallel.) This result suggests that we should use non-black-box techniques when we construct an NIZK for an NP-complete language. As another contribution, they introduced a technique named swapping technique in their proof. Chapter 4 and 5 rely on the technique, and thus we introduce the technique in Chapter 4.

1.4 Summary of Contributions

As practical NIZKs have been already proposed, NIZKs have become building blocks for practical cryptographic applications. We remark that NIZKs have been studied in a black-box manner [25, 52, 53, 54]. However, most of them treat NIZKs as "target" primitives of the constructions, and focus on a specific variant of an NIZK. To the best of our knowledge, there are few results that treat NIZKs as an underlying primitive in a black-box manner, apart from the works [49, 50].

The study of NIZKs has been developed for each assumptions, and many NIZKs have been proposed based on several assumptions such as the existence of a trapdoor permutation [25, 21], pairing [3, 4, 5], the existence of an indistinguishability obfuscation [26, 27], Diffie-Hellman assumption [55, 56, 57], and lattice based assumption [58, 59]. However, this could take time and effort for cryptographers. If some new assumption is proposed in the future, we should construct NIZKs based on

the assumption from scratch, and consider how to use the NIZK in practice, which are cumbersome tasks. Therefore, studying a black-box construction of a primitive based on an NIZK might help us to find an efficient way to use an NIZK as a building block.

This dissertation studies NIZKs in a black-box framework. We first study how to implement an NIZK as an oracle. As already mentioned, there exists a black-box framework that treats an NIZK oracle [49]. However, their instantiation is slightly complicated, and thus we simplify the oracle so that it becomes easier to use. Then, we consider the problem of black-box language extensions. Namely, we focus on a language and a technique that are employed in real world applications of NIZKs. We consider the construction of an NIZK for the witness equality, and show that such an NIZK cannot be obtained based on NIZKs that proves smaller languages in a black-box manner (recall that an NIZK for the witness equality is one of the most notable application of NIZKs). However, if we employ a technique named the commit-and-prove, we can trivially construct an NIZK for the witness equality. In fact, many real world NIZK use this technique. Hence, we investigate the limitation of such an NIZK, i.e., we consider the black-box language extension based on the specific NIZK.

This dissertation is composed as follows: Chapter 2 introduces basic notations. Then, following chapters presents our main results:

Chapter 3 The black-box framework that treats an NIZK oracle was initiated by Brakerski *et al.* [49]. They proposed an oracle that implements an NIZK and showed both positive and negative results on black-box constructions in their framework. However, their oracle instantiation was complicated so that it becomes difficult to use it. In this chapter we simplify the oracle in [49] and obtain the same results for both construction and separation. Such a simplification makes the framework easy to treat, and helps the framework to be known widely.

Chapter 4 One of the most typical usage of NIZKs is to prove the witness equality behind two different NP-statements (e.g., a plaintext of a ciphertext and a preimage of a hash function), and such an NIZK is used in real world applications such as blockchain systems. In this chapter, we show that there is no (fully) black-box construction of an NIZK for the witness equality based on NIZKs for each languages. This result suggests that if we want to construct an NIZK for such a practical language, we should rely on certain mathematical structures that relate to underlying assumptions.

Chapter 5 Regardless of the negative result in Chapter 4, we can construct an NIZK for the witness equality if we employ a specific technique named the commit-and-prove methodology [60, 61, 62]. Therefore, an NIZK which uses the

commit-and-prove methodology (CP-NIZK) is powerful enough to break the barrier demonstrated in Chapter 4, and is actually used in many practical applications. In this chapter, we uncover the limitation of CP-NIZKs in a black-box manner. That is, given CP-NIZKs for certain languages $\mathcal{L}$ and $\mathcal{L}'$ respectively, we show that there is no (fully) black-box construction of a (standard) NIZK for a language $\mathcal{L} \vee \mathcal{L}'$, where such an NIZK has been paid attention in cryptography. Hence, there is no generic methodology to expand languages that NIZKs prove even though we have practical NIZKs that rely on certain assumption.

Finally, Chapter 6 concludes this dissertation.

Caveat. We stress that the existence of CP-NIZK does not mean that the analysis in Chapter 4 is wasted. First, the NIZK modeled in Chapter 4 is of the most abstract (i.e., the most standard) form. Such an NIZK encompasses every variant of NIZKs, including a CP-NIZK. Therefore, Chapter 4 investigates the possibility to construct an NIZK for the witness equality from any NIZK. Second, it is not trivial if we can construct an NIZK for the witness equality from standard NIZKs. One of the main purpose of the black-box research is to investigate the theoretical limitation of the capability of a primitive. Furthermore, it sometimes conflicts to our intuition. For instance, the black-box construction of a signature scheme based on a one-way function [37] is surprising, since a signature schemes is a public key flavor primitive, while one-way function is a symmetric key flavor one. Therefore, the result in Chapter 4 is meaningful to understand the limitation of standard NIZKs, even if we can break such a barrier by employing a specific technique.

Chapter 2

Preliminary

2.1 Basic Notation

We denote by $n \in \mathbb{N}$ a security parameter throughout this paper. We often use the notion of a "negligible function," which is defined as below:

Definition 1 (Negligible Function) *A function $f : \mathbb{N} \to \mathbb{R}$ is* negligible *if for every polynomial function* poly*, there exists an integer N s.t. for all $n > N$, it holds that $f(n) <$ poly(n) .*

A polynomial function and a negligible function are denoted by poly and negl, respectively. For a finite set X, the notation $x \leftarrow X$ represents a sampling of an instance $x \in X$ with a uniform distribution over X. Similarly, for an algorithm A, the computation that A takes x as input and outputs y is denoted by $y \leftarrow A(x)$. A probabilistic polynomial-time Turing machine is denoted by PPT. A Turing machine M that has access to an oracle O is called an oracle Turing machine, denoted by M^O. For an NP language $\mathcal{L}$, the NP relation is denoted by $R_{\mathcal{L}}$, and we let $\mathcal{L}_n := \mathcal{L} \cap \{0,1\}^n$ and $R_n := \{(x,w) \,|\, (x,w) \in R_{\mathcal{L}} \wedge x \in \mathcal{L}_n\}$. For a function f, we denote the inverse function by f^{-1}. When y has no preimage, we write $f^{-1}(y) = \perp$. For a function $f : \{0,1\}^{n_1} \to \{0,1\}^{n_2}$, where $n_1 < n_2$, we say $y \in \{0,1\}^{n_2}$ is *legitimate* with respect to f if y has a preimage x s.t. $f(x) = y$. We say that a query to an oracle is *successful* if it has a result other than $\perp$.

The notation $y \leftarrow O(x)$ represents that a query to an oracle O on x results in y. We use oracles and algorithms that implement several functionalities. We denote by $\mathsf{M}(\mathsf{func}, x)$ an algorithm or an oracle M that works as a functionality func on input x. If the input is not important in the context, we write $\mathsf{M}.\mathsf{func}$ to denote the functionality func implemented by M. We regard an oracle O as a set of entries $(\mathsf{func}, x; y)$ where func is a function implemented by O, x is an input of func and y is an output s.t. $y \leftarrow O(\mathsf{func}, x)$. We denote by $O(\mathsf{func}, x, y)$ such an entry.

We use bracket notation $[\cdot]$ to represent a variable that matches any value; for instance, $y \leftarrow O([x])$ is a query that results in y, and we refer to the input value as x thereafter. When the matched value is not important in the context, we write $y \leftarrow O(*)$.

A partial oracle S of an oracle O is a set that is defined on only some subset of inputs of O, and S is *consistent* with O if there exists another set S' s.t. $S \cup S' = O$. We sometimes denote an oracle S by $S = S_1||S_2||\cdots$, where S_i are partial oracles and S works as follows: Given a query on x, it first searches for a matching entry $S_1(x, [y])$ and returns y if such a query exists, otherwise it searches S_2 and so on.

2.2 Black-Box Construction and Separation

Variants of Black-Box Construction

We formally introduce the notion of black-box construction by following the taxonomy in [63, 64]. Before that, we define cryptographic primitives as follows.

Definition 2 (Cryptographic Primitive) *A primitive P is a pair (F_P, R_P) of a set of functions $f : \{0,1\}^* \to \{0,1\}^*$ and a relation over pairs (f, M) where $f \in F_P$ and M is a (possibly inefficient) Turing machine.*

We say *f implements P* or *f* is an *implementation* of P if $f \in F_P$. A Turing machine M *P-breaks* the security of P if there exists an implementation $f \in F_P$ s.t. $(f, \mathsf{M}) \in R_P$. Thus, we say $f \in F_P$ is a *secure implementation* of P if there exists no PPT M s.t. $(f, \mathsf{M}) \in R_P$.

Now, we formally define what a black-box construction means as follows:

Definition 3 *There exists a* fully black-box construction *of a primitive Q from a primitive P if there exist PPT oracle machines G and S s.t.*

- *For any implementation f of P, G^f implements Q.*
- *For any implementation f of P and any oracle Turing machine M, if M^f Q-breaks the security of Q, then $S^{f,\mathsf{M}}$ P-breaks the security of P.*

We say G is a construction, and S is a reduction. The definition of a fully black-box construction requires a "universal compiler" that lifts an underlying primitive P up to a target primitive Q. That is, G^f implements Q for any $f \in F_P$, and $S^{f,\mathsf{M}}$ P-breaks the security of P for any $f \in F_P$ and any adversary M that Q-breaks the security of Q.

A more relaxed variant of a black-box construction is a semi black-box construction that is formally defined as follows:

Definition 4 *There exists a* semi black-box construction *of a primitive Q from a primitive P if there exist PPT oracle machines G s.t.*

- *For any implementation f of P, G^f implements Q.*
- *For any implementation f of P, if there exists a PPT oracle machine M s.t. M^f Q-breaks the security of Q, then there exists a PPT oracle machine S s.t. S^f P-breaks the security of P.*

A construction G in a semi black-box construction is also required to be universal. The key difference between a fully black-box construction and a semi black-box construction is that a reduction S is allowed oracle access to an adversary M or not. That is, a reduction S in a semi black-box construction does not have oracle access to M, meaning that S could depend on the internal structure of M. Therefore, a semi black-box construction is more relaxed variant than a fully black-box construction.

Other more relaxed variant of a black-box construction is a $\forall\exists$semi black-box construction. In this construction, a construction is no longer required to be universal.

Definition 5 *There exists a* $\forall\exists$semi black-box construction *of a primitive Q from a primitive P if for any implementation $f \in F_P$, there exists a PPT oracle machine G s.t.*

- *G^f implements Q.*
- *If there exists a PPT oracle machine M s.t. M^f Q-breaks the security of Q, then there exists a PPT oracle machine S s.t. S^f P-breaks the security of P.*

Finally, we introduce a relativizing construction, which is useful when we prove the absence of a black-box construction. We first define the existence of a primitive relative to an oracle, and then define a relativizing construction.

Definition 6 *Let O be an oracle. Then,*

- *O implements a primitive P if there exists an implementation $f \in F_P$ that is computable by a PPT oracle machine that has oracle access to O.*
- *An implementation f is secure relative to O if there is no PPT oracle machine M s.t. M^O P-breaks f.*
- *A primitive P exists relative to O if there exists a secure implementation f of P relative to O.*

Definition 7 *There exists a* relativizing construction *of a primitive Q from P, if for any oracle O, if P exists relative to O then so does Q.*

The black-box constructions we have introduced so far have the following implication:

Theorem 1 ([63]) *If there exists a fully black-box construction of Q from P, then there exists a semi black-box construction and a relativizing construction of Q from P. If there exists either a semi black-box construction or a relativizing construction of Q from P, then there exists a $\forall\exists$semi black-box construction of Q from P.*

Black-Box Separation Techniques

This time we introduce known techniques to show black-box impossibility. Recall that we focus on negative results on black-box constructions in this dissertation. We say that there is fully (resp, semi, $\forall\exists$semi and relativizing) black-box separation if there is no fully (resp, semi, $\forall\exists$semi and relativizing) black-box construction. If we simply say there is no black-box construction, it indicates the absence of a fully black-box construction.

One-Oracle Technique

This technique originates from the seminal work by Impagliazzo and Rudich [36]. To prove a separation between a primitive P and a primitive Q by following this technique, we first demonstrate the existence of the underlying primitive P relative to an oracle O. Then, we demonstrate a (possibly inefficient) adversary M^O that Q-breaks any implementation G^O of Q. In fact, Impagliazzo and Rudich showed the following theorem, which is sufficient, combined with Theorem 1, to rule out a fully black-box construction of a key agreement protocol based on a one-way function.

Theorem 2 *There is no relativizing black-box construction of a key agreement protocol from a one-way function.*

The one-oracle technique is most commonly used to show black-box separations. In fact, many well-known results are proven by this technique, such as the impossibility of a collision-resistant hash function from a one-way permutation [65], an identity-based encryption scheme from a trapdoor permutation [66], and the mutual impossibility between an oblivious transfer and a public key encryption scheme [43].

We remark that the work by Brakerski *et al.* [49] also uses this technique, while it seems that they introduce multiple oracles. They introduced an oracle that constitutes an NIZK for a language that is relativized with another oracle. However, in their separation proof, these oracles are given to both the construction and the adversary. Thus, we can regard these oracles as a single oracle. Further, the follow-up works [50, 67, 68] also employ this technique.

Two-Oracle Technique

Another variant of the technique showing a black-box separation is two-oracle technique, which was introduced by Hsiao and Reyzin [69] to show a separation between

public-coin collision-resistant hash functions and private-coin ones. Unlike the one-oracle technique, this technique directly shows a fully black-box separation by introducing the following types of oracles; a helper oracle A that implements an underlying primitive, and a breaker oracle B that is used to construct an adversary of a target primitive Q. They proposed that it is sufficient to demonstrate the following oracles A and B to show there is no fully black-box construction of a primitive Q from a primitive P:

- There is a PPT oracle machine L s.t. L^{A} implements P.
- For any PPT oracle machine G, if G^{A} implements Q, then there exists a PPT oracle machine M s.t. $\mathsf{M}^{\mathsf{A},\mathsf{B}}$ Q-breaks the security of G^{A}.
- There is no PPT oracle machine S s.t. $S^{\mathsf{A},\mathsf{B}}$ P-breaks the security of L^{A}.

In [69], they introduced a helper oracle G that implements a private-coin collision-resistant hash function and a "collision-finder" oracle F, and showed the separation by following the above proposal.

There are many works that employ the two-oracle technique. Haintner *et al.* [70] showed tight lower bound of a statistically hiding commitment scheme based on a one-way function. The following are also proven by this technique: A fully black-box separation of a non-interactive commitment scheme from a one-way permutation [71], and a fully black-box separation of a collision-resistant hash function from a hierarchical identity-based encryption scheme [72].

Meta-Reduction

A technique that has a different flavor compared with previous techniques is the meta-reduction, which was introduced by Gennaro and Trevisan [73]. This technique says that if there exists a reduction from a primitive Q to a primitive P, then we can use this reduction to break the security of P. A meta-reduction that demonstrates the impossibility of a primitive Q from a primitive P proceeds as follows:

- Assume that there exists P (or some assumption).
- Construct a reduction (i.e., an algorithm) S from Q to P. That is, for any adversary $\mathcal{A}$ that Q-breaks the security of an implementation of Q, $S^{\mathcal{A}}$ P-breaks the security of an implementation of P.
- Demonstrate a PPT simulator that simulates the reduction S, and prove that the distributions of outputs of S and the simulator are indistinguishable.

In other words, a meta-reduction uses a reduction as an adversary that breaks the security of an underlying primitive. Note that the existence of such a simulator conflicts the assumption that the existence of the underlying primitive.

A striking result of the meta-reduction technique is by Gentry and Wichs [54]. They proved that the soundness of a succinct non-interactive argument (SNARG), which is a variant of an efficient NIZK due to its succinct proof size, cannot be based on a falsifiable assumption [74]. Note that falsifiable assumptions are modeled as a game between a challenger and an adversary, which include general assumptions such as the existence of a one-way function, a trapdoor permutations etc, and concrete assumptions such as the hardness of factoring, DDH assumption etc. Therefore, this result suggests that we should rely on stronger assumptions such as knowledge assumption to construct a SNARG. Similarly, Pass [52] showed that it is impossible to base the adaptive soundness of a statistical NIZK based on falsifiable assumptions.

Furthermore, the meta-reduction technique can be used to show lower bounds in security loss (or security preservation in [75]). Coron [76] initiated such a usage of the meta-reduction technique to show the lower bound for the security of the probabilistic signature scheme, and follow-up works are seen in the literature [77, 78, 79].

2.3 Cryptographic Primitives

In what follows, we introduce cryptographic primitives that appear in this dissertation.

One-Way Function

We first introduce the most fundamental cryptographic primitive, a one-way function. Intuitively, a one-way function is easy to compute, but hard to invert. The formal definition of a one-way function is as follows:

Definition 8 (One-Way Function) *A function f is a* one-way function (OWF) *if the following conditions hold:*

- *There exists a PPT M_f s.t. for any x, it holds that $\mathsf{M}_f(x) = f(x)$*
- *For any PPT $\mathcal{A}$, there exists a negligible function ϵ s.t.*

$$\Pr_{x \leftarrow D_f}[\mathcal{A}(f(x)) = f^{-1}(f(x))] \leq \epsilon$$

where D_f is a domain of f. If the second condition is satisfies, then we say f has ϵ-security or f is an ϵ-OWF.

As mentioned earlier, a one-way function is equivalent to many primitives such as a signature scheme [37, 80], a pseudorandom generator [38], a pseudorandom function [81] (and thus, a symmetric key encryption scheme), a commitment scheme [39], and an (interactive) zero-knowledge proof system [82]. Note that these results are shown in a black-box manner, but they do not guarantee its efficiency.

Public Key Encryption Scheme

We define a public key encryption scheme. This primitive is also referred to an asymmetric key encryption scheme, because a party who encrypts a message and a party who decrypts a ciphertext have different keys.

Definition 9 (Public Key Encryption Scheme) *A tuple* $\Pi = (\Pi.\mathsf{Key}, \Pi.\mathsf{Enc}, \Pi.\mathsf{Dec})$ *of PPTs is a* public key encryption scheme (PKE) *where each machine works as follows:*

$\Pi.\mathsf{Key}$**:** $pk \leftarrow \Pi(\mathsf{Key}, sk)$
Given a secret key sk*, output a public key* pk*.*

$\Pi.\mathsf{Enc}$**:** $c \leftarrow \Pi(\mathsf{Enc}, pk, m, r)$
Given a public key pk*, a plaintext* m *and a randomness* r*, output a ciphertext* c*.*

$\Pi.\mathsf{Dec}$**:** $\{m, \perp\} \leftarrow \Pi(\mathsf{Dec}, sk, c)$
Given a secret key sk *and a ciphertext* c*, output a plaintext* m *or* $\perp$ *where* $\perp$ *indicates* c *is invalid. With probability 1,* $\Pi(\mathsf{Dec}, sk, \Pi(\mathsf{Enc}, pk, m, r)) = m$ *where* $pk \leftarrow \Pi(\mathsf{Key}, sk)$*.*

Definition 10 (Security Property of PKEs) *Let* $\Pi = (\Pi.\mathsf{Key}, \Pi.\mathsf{Enc}, \Pi.\mathsf{Dec})$ *be a PKE. For* $atk \in \{CPA, CCA1, CCA2\}$ *and any PPT adversary* $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$*, if the advantage* $|\Pr[\mathtt{ExptPKE}^{atk}_{\mathcal{A},\Pi}(n) = 1] - 1/2|$ *of the following experiment* $\mathtt{ExptPKE}^{atk}_{\mathcal{A},\Pi}$ *is negligible in* n*, then* Π *is said to be* atk secure*:*

$\mathtt{ExptPKE}^{atk}_{\mathcal{A},\Pi}(n)$

Choose a secret key sk, $pk \leftarrow \Pi(\mathsf{Key}, sk)$, $(m_0, m_1, \mu) \leftarrow \mathcal{A}_0^{O_0}(pk)$, $b \leftarrow \{0,1\}, r \leftarrow \{0,1\}^n$, $c \leftarrow \Pi(\mathsf{Enc}, pk, m_b, r)$, $b' \leftarrow \mathcal{A}_1^{O_1}(\mu, c)$	:	Output 1 *if* $b' = b$ else output 0

where $O_0 = \phi$ *and* $O_1 = \phi$ *if* $atk = CPA$*,* $O_0 = \Pi(\mathsf{Dec}, sk, \cdot)$ *and* $O_1 = \phi$ *if* $atk = CCA1$*, and* $O_0 = \Pi(\mathsf{Dec}, sk, \cdot)$*,* $O_1 = \Pi(\mathsf{Dec}, sk, \cdot)$ *if* $atk = CCA2$*. None of the queries of* $\mathcal{A}_1$ *contains the challenge ciphertext* c *if* $atk = CCA2$*.*

One of the most interesting open question regarding PKEs is that if we can construct a CCA-PKE based on a CPA-PKE in a black-box manner. Gertner *et al.* [46] demonstrated that such a construction is impossible if a decryption algorithm of a construction does not have access to an encryption functionality of an underlying CPA-PKE. However, Cramer *et al.* [83] showed that this is possible if an adversary is allowed to make polynomially many queries where the polynomial is determined by a key generation algorithm (note that the adversary in [46] can make arbitrarily

polynomial many queries). We remark that many non-black-box constructions have been proposed that uses an NIZK [47, 48, 51, 84], a hash-proof system [85, 86, 87], and an identity based encryption scheme [88].

Non-Interactive Zero-Knowledge Proof System

Now, we introduce an NIZK formally. Before that, we define relevant primitives, a proof system and a witness indistinguishable proof system.

Definition 11 *A pair* $\Pi = (\Pi.\mathsf{Prv}, \Pi.\mathsf{Vrf})$ *of machines that works as follows is a* proof system *for a language* $\mathcal{L}$*:*

$\Pi.\mathsf{Prv}$**:** $\pi \leftarrow \Pi.\mathsf{Prv}(x, w, r)$
Given an instance x*, a witness* w *and a randomness* r*, output a proof* π

$\Pi.\mathsf{Vrf}$**:** $b \leftarrow \Pi.\mathsf{Vrf}(x, \pi)$
Given an instance x *and a proof* π*, output a bit* $b \in \{0, 1\}$*, where 1 means accepts and 0 means reject.*

Definition 12 *A proof system* $\Pi = (\Pi.\mathsf{Prv}, \Pi.\mathsf{Vrf})$ *for a language* $\mathcal{L}$ *has the following properties:*

Completeness: *For any* $n \in \mathbb{N}$*, for any* $(x, w) \in R_{\mathcal{L}}$*, and any randomness* $r \in \{0, 1\}^n$*,* $\Pr[\Pi.\mathsf{Vrf}(x, \Pi(\mathsf{Prv}, x, w, r)) = 1] \geq 1 - \mathrm{negl}(n)$.

Soundness: *For any* $n \in \mathbb{N}$*, any* $x \notin \mathcal{L}$*, and any* $\pi \in \{0, 1\}^{\mathrm{poly}(n)}$*, it holds that* $\Pr[\Pi.\mathsf{Vrf}(x, \Pi.\mathsf{Prv}(x, w, r))] \leq \mathrm{negl}(n)$.

Definition 13 *A proof system* $\Pi = (\Pi.\mathsf{Prv}, \Pi.\mathsf{Vrf})$ *for a language* $\mathcal{L}$ *is a* witness indistinguishable proof system (WI)*, if for any adversary* $\mathcal{A}$ *the advantage denoted by* $|\Pr[\mathtt{ExptWI}_{\Pi,\mathcal{A}}(n) = 1] - \frac{1}{2}|$ *of the following experiment* $\mathtt{ExptWI}_{\Pi,\mathcal{A}}(n)$ *is negligible:*

$$\begin{array}{cc} (x, w_0, w_1) \leftarrow \mathcal{A}^{\Pi}(1^n); & \\ b \leftarrow \{0,1\};\ r \leftarrow \{0,1\}^n; & \textit{if } (x, w_0), (x, w_1) \in R_{\mathcal{L}} \\ \pi \leftarrow \Pi(\mathsf{Prv}, x, w_b, r); \quad : & \textit{output 1 iff } b' = b \\ b' = \mathcal{A}^{\Pi}(1^n, \pi) & \textit{else output a random bit.} \end{array}$$

Our target primitive, an NIZK, is formally defined as follows.

Definition 14 (Non-Interactive Zero-Knowledge Proof System) *A tuple of PPTs* $\Pi = (\Pi.\mathsf{Crs}, \Pi.\mathsf{Prv}, \Pi.\mathsf{Vrf}, \Pi.\mathsf{CrsSim}, \Pi.\mathsf{PrvSim})$ *that work as follows is a* non-interactive zero-knowledge proof system (NIZK) *for a language* $\mathcal{L}$*.*

$\Pi.\mathsf{Crs}$**:** $\sigma \leftarrow \Pi(\mathsf{Crs}, \tau)$
Given a trapdoor τ*, output a common reference string (CRS)* σ*.*

$\Pi.\mathsf{Prv}$**:** $\pi \leftarrow \Pi(\mathsf{Prv}, \sigma, x, w)$
Given a CRS σ, an instance x and a witness w, output a proof π or $\perp$.

$\Pi.\mathsf{Vrf}$**:** $b \leftarrow \Pi(\mathsf{Vrf}, \sigma, x, \pi)$
Given a CRS σ, an instance x and a proof π, output a bit $b \in \{0,1\}$ where 1 means accept and 0 means reject.

$\Pi.\mathsf{CrsSim}$**:** $(\sigma, \tau) \leftarrow \Pi(\mathsf{CrsSim}, \tau)$
Given a trapdoor τ, output τ and a CRS σ.

$\Pi.\mathsf{PrvSim}$**:** $\pi \leftarrow \Pi(\mathsf{PrvSim}, \sigma, x, \tau)$
Given a CRS σ, an instance x and a trapdoor τ, output a proof π or $\perp$.

Definition 15 (Security Properties of NIZKs) *An NIZK Π for a language $\mathcal{L}$ has the following properties.*

Completeness: *For any $n \in \mathbb{N}$, any $\sigma \leftarrow \Pi(\mathsf{Crs}, \tau)$ and any $(x, w) \in R_{\mathcal{L}}$, it holds that* $\Pr[\Pi(\mathsf{Vrf}, \sigma, x, \Pi(\mathsf{Prv}, \sigma, x, w)) = 1] \geq 1 - \mathrm{negl}(n)$.

Soundness: *For any PPT adversary $\mathcal{A}$, it holds that*
$\Pr[\sigma \leftarrow \Pi(\mathsf{Crs}, \tau), (x, \pi) \leftarrow \mathcal{A}(\sigma) \,:\, \Pi(\mathsf{Vrf}, \sigma, x, \pi) = 1 \wedge x \notin \mathcal{L}] \leq \mathrm{negl}(n)$.

Adaptive Zero-Knowledge: *For any stateful PPT adversary $\mathcal{A}$, it holds that*

$$\begin{aligned}
&\mathtt{AdvZK}_{\mathcal{A},\Pi,\mathcal{L}}(n) \\
&\quad = \left| \Pr\left[\begin{array}{cc} \sigma \leftarrow \Pi(\mathsf{Crs}, \tau) & \\ (x, w) \leftarrow \mathcal{A}(\sigma) & : \begin{array}{c} \mathcal{A}(\pi) = 1 \\ \wedge (x, w) \in R_{\mathcal{L}} \end{array} \\ \pi \leftarrow \Pi(\mathsf{Prv}, \sigma, x, w) & \end{array}\right] \right. \\
&\qquad \left. - \Pr\left[\begin{array}{cc} (\sigma, \tau) \leftarrow \Pi(\mathsf{CrsSim}, \tau) & \\ (x, w) \leftarrow \mathcal{A}(\sigma) & : \begin{array}{c} \mathcal{A}(\pi) = 1 \\ \wedge (x, w) \in R_{\mathcal{L}} \end{array} \\ \pi \leftarrow \Pi(\mathsf{PrvSim}, \sigma, x, \tau) & \end{array}\right] \right| \\
&\quad \leq \mathrm{negl}(n).
\end{aligned}$$

While the above definition of the adaptive zero-knowledge property is standard, we use the following definition for ease of the discussion in Chapter 4 and 5:

$$\begin{aligned}
&\mathtt{AdvZK}_{\mathcal{A},\Pi,\mathcal{L}}(n) \\
&= \Pr\left[\begin{array}{cc} \sigma \leftarrow \Pi(\mathsf{Crs}, \tau) & \\ (x, w) \leftarrow \mathcal{A}(\sigma) & : \begin{array}{c} \mathcal{A}(\pi) = 1 \\ \wedge (x, w) \in R_{\mathcal{L}} \end{array} \\ \pi \leftarrow \Pi(\mathsf{Prv}, \sigma, x, w) & \end{array}\right] \\
&\quad - \Pr\left[\begin{array}{cc} (\sigma, \tau) \leftarrow \Pi(\mathsf{CrsSim}, \tau) & \\ (x, w) \leftarrow \mathcal{A}(\sigma) & : \begin{array}{c} \mathcal{A}(\pi) = 1 \\ \wedge (x, w) \in R_{\mathcal{L}} \end{array} \\ \pi \leftarrow \Pi(\mathsf{PrvSim}, \sigma, x, \tau) & \end{array}\right] \\
&\leq \mathrm{negl}(n).
\end{aligned}$$

Note, however, that this form of definition is essentially the same as the standard definition; if there exists an adversary $\mathcal{A}$ s.t. $\mathtt{AdvZK}_{\mathcal{A},\Pi,\mathcal{L}}(n) \leq -\mathrm{poly}(n)$, then it implies the existence of another adversary $\mathcal{A}'$ s.t. $\mathtt{AdvZK}_{\mathcal{A}',\Pi,\mathcal{L}}(n) \geq \mathrm{poly}(n)$.

Sahai [51] introduced the notion of a simulation-sound NIZK (SS-NIZK) that captures the intuition behind zero-knowledge proof systems. That is, a proof generated by an NIZK does not increase the power of an adversary apart from the capability of proving the same statement.

Definition 16 ((Unbounded) Simulation-Sound NIZK) *Let* $\Pi = (\Pi.\mathsf{Crs}, \Pi.\mathsf{Prv}, \Pi.\mathsf{Vrf}, \Pi.\mathsf{CrsSim}, \Pi.\mathsf{PrvSim})$ *be an NIZK for a language* $\mathcal{L}$*. Then* Π *is an (unbounded)* simulation-sound NIZK (SS-NIZK) *for* $\mathcal{L}$ *if for any stateful PPT adversary* $\mathcal{A}$*, the probability* $\Pr[\mathtt{ExptSS}_{\mathcal{A},\Pi,\mathcal{L}}(n) = 1]$ *is negligible in* n *where* $\mathtt{ExptSS}_{\mathcal{A},\Pi,\mathcal{L}}(n)$ *is defined as follows:*

$\mathtt{ExptSS}_{\mathcal{A},\Pi,\mathcal{L}}(n)$

$$\begin{array}{ll} (\sigma,\tau) \leftarrow \Pi(\mathsf{CrsSim}, 1^n) & \\ (x,\pi) \leftarrow \mathcal{A}^{\Pi(\mathsf{PrvSim},\sigma,\cdot,\tau)}(\sigma) & : \begin{array}{l} \textit{output } 1 \textit{ if } x \notin \mathcal{L} \wedge x \notin Q \wedge \Pi(\mathsf{Vrf},\sigma,x,\pi) = 1 \\ \textit{else output 0} \end{array} \end{array}$$

where Q *is a list of queries that* $\mathcal{A}$ *makes to* $\Pi(\mathsf{PrvSim},\sigma,\cdot,\tau)$.

As mentioned earlier, NIZKs are becoming building blocks for several applications, and used to construct many cryptographic protocols. There has been constructions that uses NIZKs as building block such as CCA-PKEs [47, 84], digital signatures [89, 90] and multi-party computations [60].

2.4 The Naor-Yung Construction

Every work in this dissertation somewhat relates to the well-known Naor-Yung construction (or the Naor-Yung/Sahai paradigm) [47, 51]. The Naor-Yung construction is a methodology which enhances a CPA-PKE into a CCA-PKE by using an NIZK as follows: Encrypt a message by two distinct public keys of a CPA-PKE together with a zero-knowledge proof that shows these two ciphertexts are generated from the same plaintext. Let $\Pi = (\Pi.\mathsf{Key}, \Pi.\mathsf{Enc}, \Pi.\mathsf{Dec})$ be a CPA-PKE and $\mathsf{L} = (\mathsf{L.Crs}, \mathsf{L.Prv}, \mathsf{L.Vrf}, \mathsf{L.CrsSim}, \mathsf{L.PrvSim})$ be an NIZK for the following *plaintext equality language* with respect to Π:

$$\mathcal{L}_{EQ}^{\Pi} = \{(c_0, c_1, pk_0, pk_1) \mid \exists m, r_0, r_1 \text{ s.t. } c_0 = \Pi(\mathsf{Enc}, pk_0, m, r_0) \\ \wedge\, c_1 = \Pi(\mathsf{Enc}, pk_1, m, r_1)\}.$$

The formal description of the Naor-Yung construction M is as follows:

M.Key: $pk^* \leftarrow \mathsf{M}(\mathsf{Key}, sk_0, sk_1)$

Given secret keys sk_0 and sk_1, compute public keys $pk_0 \leftarrow \Pi(\mathsf{Key}, sk_0), pk_1 \leftarrow \Pi(\mathsf{Key}, sk_1)$ and a CRS $\sigma \leftarrow \mathsf{L}(\mathsf{Crs}, 1^n)$, and output $pk^* := (pk_0, pk_1, \sigma)$.

M.Enc: $c^* \leftarrow \mathsf{M}(\mathsf{Enc}, pk^*, m, r_0, r_1)$

Given a public key $pk^* = (pk_0, pk_1, \sigma)$, a plaintext m and randomnesses $r_0, r_1 \in \{0,1\}^n$, compute $c_0 = \Pi(\mathsf{Enc}, pk_0, m, r_0), c_1 = \Pi(\mathsf{Enc}, pk_1, m, r_1)$ and $\pi \leftarrow \mathsf{L}(\mathsf{Prv}, \sigma, (c_0, c_1, pk_0, pk_1), m, r_0, r_1)$ and output $c^* := (c_0, c_1, \pi)$.

M.Dec: $m \leftarrow \mathsf{M}(\mathsf{Dec}, sk^*, c^*)$

Given a secret key $sk^* = sk_0$ and a ciphertext $c^* = (c_0, c_1, \pi)$, output $\perp$ if $\mathsf{L}(\mathsf{Vrf}, \sigma, (c_0, c_1, pk_0, pk_1), \pi) = 0$, otherwise output $m := \Pi(\mathsf{Dec}, sk_0, c_0)$.

The above construction is not necessarily CCA2 secure. The first CCA2-PKE based on general assumption is by Dolev *et al.* [84]. However, their construction is quite complicated, and Sahai [51] removed this drawback. He showed that CCA2 security can be achieved if the NIZK in the Naor-Yung construction is simulation-sound.

Chapter 3

Simplification of The Augmented Black-Box Framework

3.1 Introduction

In this chapter, we show the first result; the simplification of the black-box framework introduced by Brakerski *et al* [49].

After the seminal work of Impagliazzo and Rudich [36], black-box construction has become one of the main research topic in cryptography. However, non-black-box techniques, which are the methodologies that use internal structure of an underlying primitive (such as algebraic structure) to construct more high-level primitives, are also extensively studied independent of black-box techniques [47, 89, 91, 92]. Among them, the Naor-Yung construction [47] is one of the most widely-known non-black-box result.

Although black-box and non-black-box techniques have been developed independently, a new framework that combines them has proposed. Brakerski *et al.* [49] introduced the *augmented* black-box framework that captures the power of certain non-black-box techniques. More precisely, they introduced an oracle that instantiates a WI [93] for an NP-complete language, and showed a black-box construction of an NIZK based on the oracle. To demonstrate the power of the framework, they showed known construction and separation in their model; the Naor-Yung construction is possible in their framework, and the separation between a OWF and a key-agreement protocol (KA) [94]. We remark that it is not known if we can construct a CCA-PKE based only on a CPA-PKE in a black-box manner. Therefore, the augmented black-box framework is powerful enough to capture the power of non-black-box constructions.

As mentioned in Chapter 2, one of the major black-box technique is relativizing (Definition 7). In the beginning of the line of the black-box works, researchers treated

simple oracles such as the one that implements a OWF [47]. However as more sophisticated primitives appeared, they became have to deal with oracles that implement more complicated primitives such as trapdoor permutation [66, 95], which led more advanced security proof. Moreover in [49], the augmented black-box framework was accompanied by further complicated oracle that implements an NIZK. Although the augmented black-box framework is an elegant framework, security proofs in this framework might become a cumbersome task due to the high complexity of the oracle. It is generally preferable to introduce simpler oracle from the view of separation proofs.

In this work we simplify the WI oracle in [49] that aims to give the same results for both construction and separation. Technically, we simplify the prover oracle from $\mathsf{Prv}(x, w, r)$ to $\mathsf{Prv}(x, w)$ by removing the randomness r, which results in a non-WI oracle but we can build an NIZK based on it. A question is how non-trivial this seemingly small change is and how it aims in separating cryptographic objects in the presence of NIZKs. As it is observed in Section 6.3.3 of [96], a very small change of the specification of Prv spoils the separation proof**.** Thus it is highly non-trivial to see if the small change works or not.

In the particular case about the impossibility of a KA based on a OWF, the proof of separation in [49] is done in a way that the randomness r does not play an essential role in building an adversary. Thus we conclude that the randomness in Prv was not necessary *in the first place*.

3.1.1 Related Work

As is already mentioned, the augmented black-box framework captures the power of NIZKs. There are follow-up works of [49] that capture other non-black-box techniques. Asharov and Segev [67] showed that there is no black-box constructions of a collision-resistant function family from an indistinguishability obfuscation, and of a perfectly complete key-agreement protocol from a private-key functional encryption scheme. The more recent work of Garg *et al.* [68] captures the power of Yao's garbled circuit [97] and showed PKEs are impossible in their model. These frameworks give rather complicated oracles such as functional encryption schemes or garbled circuit. Couteau *et al.* [98] introduced the notion of black-box uselessness s.t. if there is no black-box construction of Q based on P and P' respectively, then combining P and P' does not help to construct Q in a black-box manner. In their security proof, they heavily rely on the technique used in [49].

3.2 The WI Oracle by Brakerski et al.

This section reviews the work of Brakerski *et al.* [49]. They introduced a distribution of oracles s.t. a pair of oracles chosen according to this distribution constitutes a WI and they built an NIZK based on the oracle. Moreover they defined the augmented black-box framework and demonstrated the power of the framework by showing the construction and separation in their model. We first overview their result, and then introduce the observation regarding the specification of the oracle made by Yerukhimovich [96], which has motivated this work.

Instantiation of a WI Oracle

Fix a security parameter $n \in \mathbb{N}$ and an oracle O. Let $\mathcal{L} = \mathsf{CIRCUIT\text{-}SAT}^O$. The distribution of oracles that a pair of oracles $\mathsf{WI}_n = (\mathsf{WI}_n.\mathsf{Prv}, \mathsf{WI}_n.\mathsf{Vrf})$ chosen according to this distribution constitutes a WI for $\mathcal{L}_n$ with overwhelming probability as follows.

$\mathsf{WI}_n.\mathsf{Prv}$**:** $\pi \leftarrow \mathsf{WI}_n(\mathsf{Prv}, x, w, r)$

The prover oracle $\mathsf{WI}_n.\mathsf{Prv}$ is a random function s.t. $\mathsf{WI}_n.\mathsf{Prv} : \{0,1\}^{3n} \to \{0,1\}^{7n}$. Given an instance $x \in \{0,1\}^n$, a witness $w \in \{0,1\}^n$ and a randomness $r \in \{0,1\}^n$, output a proof $\pi \in \{0,1\}^{7n}$. Note that $\mathsf{WI}_n.\mathsf{Prv}$ does not check if $(x, w) \in R_n$.

$\mathsf{WI}_n.\mathsf{Vrf}$**:** $b \leftarrow \mathsf{WI}_n(\mathsf{Vrf}, x, \pi)$

The verifier oracle $\mathsf{WI}_n.\mathsf{Vrf}$ is a function s.t. $\mathsf{WI}_n.\mathsf{Vrf} : \{0,1\}^{8n} \to \{0,1\}$. Given an instance $x \in \{0,1\}^n$ and a proof $\pi \in \{0,1\}^{7n}$, $\mathsf{WI}_n.\mathsf{Vrf}$ works as follows:

$$\mathsf{WI}_n(\mathsf{Vrf}, x, \pi) = \begin{cases} 1 & \text{if } \exists w, r \text{ s.t. } \pi = \mathsf{WI}_n(\mathsf{Prv}, x, w, r) \wedge (x, w) \in R_n \\ 0 & \text{otherwise.} \end{cases}$$

We say a pair of oracles chosen according to this distribution a *WI oracle*. Let $\mathsf{WI} = \{\mathsf{WI}_n\}_{n \in \mathbb{N}}$ be a family of WI oracles. Then it has been shown that WI constitutes a WI for $\mathcal{L}$.

Theorem 3 *For measure 1 of the oracles* WI *under the distribution above,* WI *constitutes a WI for* $\mathcal{L}$*.*

Recall that the augmented black-box framework is defined to capture the power of NIZKs, whereas the above oracles only constitutes a WI. To achieve the purpose, they showed there exists a black-box construction of an NIZK for $\mathcal{L}$ based on the WI oracle.

Theorem 4 *There exists a black-box construction of an NIZK based on* WI.

As a technical result, the following theorem is shown.

Theorem 5 *Let O be an oracle s.t. there exists a OWF f^O relative to O, and* WI *be a WI oracle. Then f^O is one-way relative to O and* WI.

They demonstrated that the Naor-Yung/Sahai construction is accomplished in this framework. This is not surprising, as the NIZK is capable to prove an NP-complete language.

Theorem 6 *There is an augmented black-box construction of a CCA-PKE based on a CPA-PKE.*

Although the augmented black-box framework is powerful enough to encompass the Naor-Yung/Sahai construction, still there exists a gap between a OWF and a KA [47].

Theorem 7 *There is no augmented black-box construction of a (perfectly complete) KA based on a OWF.*

In Chapter 3.5.1, we review their proof in detail.

Important Observation

We mention an important observation regarding the oracle interface made by Yerukhimovich [96], which contains the full version of [49]. He first tried to instantiate the prover interface independent on w, i.e., $\mathsf{WI}(\mathsf{Prv}, x, r) = \pi$, to guarantee that the proof does not leak any secret. However, he found such an instantiation is too powerful to construct a KA in black-box way: Let A and B be PPTs who play a KA. A chooses random strings $x, r \leftarrow \{0,1\}^n$ and B picks $x' \leftarrow \{0,1\}^n$. They make query $y = O(x)$ and $y' = O(x')$ and send the values r, y, y' each other. Then, A and B query $\mathsf{WI}.\mathsf{Prv}$ for a proof of the following language $\{y, y' \mid \exists w \, s.t. \, y = O(w) \wedge y' = O(w))\}$ and obtain proofs π_A and π_B. Note that this language has exactly two witnesses x and x' and each of A and B knows one, while an eavesdropper cannot learn either due to the one-wayness of O. If $\mathsf{WI}.\mathsf{Prv}$ is independent of the witness then $\pi_A = \pi_B$ and A and B have a secret which an eavesdropper can not learn. Since a KA cannot be constructed under the instantiation $\mathsf{WI}(\mathsf{Prv}, x, w, r) = \pi$, a slight modification on the oracle can significantly affect on its capability.

3.3 Simplified Proof System Oracle

This section introduces a more simplified proof system oracle. In [49], they constructed an NIZK by making use of witness indistinguishability of the WI oracle defined in Chapter 3.2. However, as the prover oracle is a random function, we observe that we can remove the random coin r from its interface, resulting a simpler prover oracle. We begin by describing distribution of oracles s.t. an oracle chosen according to this distribution constitutes a proof system oracle. Then we show that we can construct a WI based on the simplified oracle in a black-box manner over the choice of the simplified oracle. Finally we show that, for measure 1 of oracles under this distribution, we can construct a WI. Since a WI can be constructed based on our oracle, we obtain the same construction and separation results as [49]. Our simplified oracle actually makes the construction of WI slightly more complicated than before. However it is mostly a once-for-all task where our construction can be reused in future proofs of separation.

Coin-Free Proof System Oracle

Fix a security parameter n and an oracle O. Throughout this chapter, we let $\mathcal{L} = \mathsf{CIRCUIT\text{-}SAT}^O$. We introduce a distribution of pair of oracles as follows.

Definition 17 *A pair* $\mathsf{CF}_n = (\mathsf{CF}_n.\mathsf{Prv}, \mathsf{CF}_n.\mathsf{Vrf})$ *of oracles chosen from the following distribution is a* coin-free proof system oracle *for* $\mathcal{L}_n$*;*

$\mathsf{CF}_n.\mathsf{Prv}$**:** $\pi \leftarrow \mathsf{CF}_n(\mathsf{Prv}, x, w)$
The prover oracle $\mathsf{CF}_n.\mathsf{Prv}$ *is a random function* $\mathsf{CF}_n.\mathsf{Prv} : \{0,1\}^{2n} \to \{0,1\}^{6n}$. *Given an instance* $x \in \{0,1\}^n$ *and a witness* $w \in \{0,1\}^n$*, output a proof* $\pi \in \{0,1\}^{6n}$. *Note that* $\mathsf{CF}_n.\mathsf{Prv}$ *does not check if* $(x, w) \in R_{\mathcal{L}_n}$.

$\mathsf{CF}_n.\mathsf{Vrf}$**:** $b \leftarrow \mathsf{CF}_n(\mathsf{Vrf}, x, \pi)$
The verifier oracle $\mathsf{CF}_n.\mathsf{Vrf}$ *is a random function* $\mathsf{CF}_n.\mathsf{Vrf} : \{0,1\}^{7n} \to \{0,1\}$. *Given an instance* $x \in \{0,1\}^n$ *and a proof* $\pi \in \{0,1\}^{6n}$*,* $\mathsf{CF}_n.\mathsf{Vrf}$ *works as follows:*

$$\mathsf{CF}_n(\mathsf{Vrf}, x, \pi) = \begin{cases} 1 & \text{if } \exists w \text{ s.t. } \pi = \mathsf{CF}_n(\mathsf{Prv}, x, w) \ \wedge (x, w) \in R_{\mathcal{L}_n} \\ 0 & \text{otherwise.} \end{cases}$$

We treat the infinite oracles $\mathsf{CF} = (\mathsf{CF.Prv}, \mathsf{CF.Vrf})$ as a sequence of oracles $\{\mathsf{CF}_n\}_{n\in\mathbb{N}}$. It is clear that CF_n constitutes a proof system. We remark that CF_n is no longer witness indistinguishable, since an adversary, given a proof π in the experiment $\mathtt{ExptWI}$, can decide which witness w_0 or w_1 was used to generate π by making queries $\mathsf{CF}_n(\mathsf{Prv}, x, w_0)$ and $\mathsf{CF}_n(\mathsf{Prv}, x, w_1)$.

Construction of a WI

We build a WI for $\mathcal{L}_n$ based on a coin-free proof system oracle. Our construction is similar to the construction of the NIZK in [49]. The key observation for our construction is that we can flip a random coin in the construction, and any NP language can be cast to the instance of CF since it proves an NP-complete language. Given $(x, w) \in R_{\mathcal{L}_n}$, the prover randomizes these values and proves knowledge about w and the randomness so that the proof achieves witness indistinguishability. Technically, we introduce a OWF f for the randomization and let the prover choose $r \leftarrow \{0,1\}^n$ uniformly, compute $c = f(r)$ and prove both $(x, w) \in R_{\mathcal{L}_n}$ and $c = f(r)$ simultaneously. As proven later, because $\mathcal{L}_n$ is an NP-complete language, we can reduce the above two statements to a single statement for $\mathcal{L}_{3n}$.

Fix a security parameter n. Let O be an oracle and CF_{3n} be a coin-free proof system oracle s.t. there exists an ϵ-OWF $f^O : \{0,1\}^n \to \{0,1\}^{2n}$ relative to O and CF_{3n}. Such a OWF exists because of Theorem 5 and the fact that a WI implies a proof system generally. We define $\mathcal{L}_n^* := \{c \,|\, \exists\, r \text{ s.t. } c = f^O(r)\}$ and $\hat{\mathcal{L}}_n := \{(x, c) \mid \exists\, w, r \text{ s.t. } c = f^O(r) \wedge (x, w) \in R_{\mathcal{L}_n}\}$. The construction of WI is, given $(x, w) \in R_{\hat{\mathcal{L}_n}}$, choose $r \leftarrow \{0,1\}^n$ uniformly, compute $c = f^O(r)$ and apply Karp reduction [99] to $((x, c), (w, r)) \in R_{\hat{\mathcal{L}}_n}$ to obtain $(x', w') \in \mathcal{L}_{3n}$. While it is already known that for any oracle O, the class NP^O has a complete language [100], we demonstrate the above reduction explicitly.

Lemma 1 *There exists a Karp reduction from $\hat{\mathcal{L}}_n$ to $\mathcal{L}_{3n}$.*

Proof of Lemma 1. Given (x, c), whether $(x, c) \in \hat{\mathcal{L}}_n$ can be determined by proving $x \in \mathcal{L}_n$ and $c \in \mathcal{L}_n^*$ one by one. Furthermore, $\mathcal{L}_n$ is already $\mathsf{CIRCUIT\text{-}SAT}^O$ with input length n. Hence, if we prove that $\mathcal{L}_n^*$ can be reduced to $\mathcal{L}_{2n}$ then the lemma is done. Toward this, we show the existence of a circuit C that simulates the verifier [1] for $\mathcal{L}_n^*$.

Without loss of generality, we consider the simplest construction of f. That is, given an oracle $O : \{0,1\}^n \to \{0,1\}^{2n}$ and $r \in \{0,1\}^n$, f makes a query to O on r and outputs the answer c accordingly. An O-gate in a circuit is represented as a gate with n input wires and $2n$ output wires. The simulator circuit C is described as follows:

- Given $r \in \{0,1\}^n$ and $c \in \{0,1\}^{2n}$, it forwards r to an O-gate resulting in $c' \in \{0,1\}^{2n}$.
- Output 1 if $c = c'$ (this comparison is performed by comparing c and c' one by one bit), otherwise 0.

[1] The terminology verifier is in terms of language, not in terms of proof system.

Clearly, C can be constructed at most polynomial number of gates. Note that, no matter how f modifies the answer from O, it can be represented by polynomial number of gates. □

Now, we are ready to construct a WI based on the oracles. Formally, we construct a WI $\Pi = (\Pi.\mathsf{Prv}, \Pi.\mathsf{Vrf})$ as follows:

$\Pi^{O,\mathbf{CF}_{3n}}.\mathsf{Prv}$**:** $\pi \leftarrow \Pi^{O,\mathsf{CF}_{3n}}(\mathsf{Prv}, x, w)$
Given $x, w \in \{0,1\}^n$, choose $r \leftarrow \{0,1\}^n$ uniformly and compute $c = f^O(r)$. Let $\hat{x} := (x, c)$ and $\hat{w} := (w, r)$. Note that if $(x, w) \in R_{\mathcal{L}_n}$ then $(\hat{x}, \hat{w}) \in R_{\hat{\mathcal{L}}_n}$. Apply Karp reduction to $(\hat{x}, \hat{w}) \in R_{\hat{\mathcal{L}}_n}$ to obtain $(x', w') \in R_{\mathcal{L}_{3n}}$. Note that while $\hat{w} \in \{0,1\}^{2n}$, it can be reduced to the witness for $\mathcal{L}_{3n}$. Compute $\pi' = \mathsf{CF}_{3n}(\mathsf{Prv}, x', w')$, and output $\pi := (c, \pi')$.

$\Pi.\mathsf{Vrf}^{O,\mathbf{CF}_{3n}}$**:** $b \leftarrow \Pi^{O,\mathsf{CF}_{3n}}(\mathsf{Vrf}, x, \hat{\pi})$
Given $x \in \{0,1\}^n$ and $\hat{\pi} = (c, \pi) \in \{0,1\}^{2n} \times \{0,1\}^{18n}$, let $x' := (x, c)$. Apply Karp reduction to $x' \in \hat{\mathcal{L}}$ to obtain $\hat{x} \in \mathcal{L}_{3n}$. Output $b = \mathsf{CF}_{3n}(\mathsf{Vrf}, \hat{x}, \pi)$.

To prove the security of the construction formally, we should show the existence of a family of oracles $\mathsf{CF} = \{\mathsf{CF}_n\}_{n\in\mathbb{N}}$ s.t. for measure 1 of oracles CF under the distribution defined in Definition 17 the construction is a WI for $\mathcal{L}_n$. Toward this, firstly we show that the construction is WI for $\mathcal{L}_n$ with overwhelming probability over the choice of coin-free proof system oracles for a fixed security parameter n.

Lemma 2 *Fix a security parameter n and an oracle O. The construction Π is a WI for $\mathcal{L}_n \in \mathrm{NP}^O$ with overwhelming probability over the choice of CF_{3n}.*

Proof of Lemma 2. The completeness is immediate and the soundness is as following. Suppose that $\Pi.\mathsf{Vrf}$ is given an $x \notin \mathcal{L}_n$ and a (forged) proof $\pi = (c, \pi')$ for x. Then $\Pi.\mathsf{Vrf}$ applies Karp reduction to (x, c) to obtain x' and makes a query to $\mathsf{CF}_{3n}.\mathsf{Vrf}$ on (x', π') to determine if the proof is valid. Since Karp reduction is deterministic and CF_{3n} has the soundness, there is no chance that this proof is accepted.

We show the witness indistinguishability by following the idea of the proof of Lemma 2.4 in [49]. Let $\mathcal{A}$ be an adversary which is bounded by at most q queries. Note that an adversary in the experiment $\mathtt{ExptWI}_{\mathcal{A}}^{O,\mathsf{CF}_{3n}}(n)$ has oracle access to O and CF_{3n} and we abuse notation to write $\mathcal{A}$ to denote $\mathcal{A}^{O,\mathsf{CF}_{3n}}$. The experiment proceeds as follows. The adversary $\mathcal{A}$ outputs values (x, w_0, w_1) with $(x, w_0), (x, w_1) \in R_{\mathcal{L}_n}$. Given x, w_0 and w_1, the challenger chooses $b \leftarrow \{0,1\}$ and $r \leftarrow \{0,1\}^n$ uniformly, computes $c = f^O(r)$, apply Karp reduction to (x, c) and (w_b, r) to obtain $(x', w') \in R_{\mathcal{L}_{3n}}$, compute $\pi' = \mathsf{CF}_{3n}(\mathsf{Prv}, x', w')$, and outputs $\pi = (c, \pi')$. Then $\mathcal{A}$ is given π and tries to decide which of w_0 and w_1 was used to generate π. In the following we first define an bad event s.t. $\mathcal{A}$ breaks the witness indistinguishability by accident and prove that such an event occurs only with negligible probability. Then we show that,

assuming such event never happens, $\mathcal{A}$ breaks the witness indistinguishability only with negligible probability.

Let $\mathtt{Spoof}$ be an event that $\mathcal{A}$ makes a query $\mathsf{CF}_{3n}(\mathsf{Vrf}, x', \pi')$ returning 1, yet no query to $\mathsf{CF}_{3n}.\mathsf{Prv}$ on $(x', w') \in R_{\mathcal{L}_{3n}}$ that results in π' was made previously. We prove that $\mathtt{Spoof}$ occurs only with negligible probability. Because $\mathsf{CF}_{3n}.\mathsf{Prv}$ is a random function, Spoof is the event that uniformly chosen $\pi' \in \{0,1\}^{18n}$ is a valid proof for x'. However, the probability that such π' is valid is at most $2^{6n}/2^{18n} = 2^{-12n}$ because at most 2^{6n} points are distributed in the range of $\mathsf{CF}_{3n}.\mathsf{Prv}$ and making a query to $\mathsf{CF}_{3n}.\mathsf{Prv}$ reveals one point in the range. Taking a union bound, the probability that $\mathtt{Spoof}$ occurs during the execution of $\mathcal{A}$ is at most $q \cdot 2^{-12n}$, which is negligible.

We prove that, assuming Spoof never occurs, no polynomial time adversary $\mathcal{A}$ can violate the witness indistinguishability. As Spoof never occurs, $\mathcal{A}$ is necessary to compute w' s.t. $\mathsf{CF}_{3n}(\mathsf{Prv}, x', w') = \pi'$. Note that $\mathcal{A}$ can compute x' by herself from given information x and c. To find such w' without Spoof, $\mathcal{A}$ should compute r s.t. $c = f^O(r)$, which is possible only with negligible probability because f is an ϵ-OWF. Summarizing the above discussion, $\mathcal{A}$ breaks the witness indistinguishability with probability at most $q \cdot 2^{-12n} + \epsilon$, which is negligible. □

For the proof of the formal security of the construction, we use the following Borel-Cantelli Lemma.

Theorem 8 *Let $E_1, E_2, \cdots$ be a sequence of events on the same probability space. Then $\sum_{n=1}^{\infty} \Pr[E_n] < \infty$ implies that $\Pr[\bigwedge_{k=1}^{\infty} \bigvee_{n \geq k} E_n] = 0$.*

Lemma 3 *Fix an oracle O. For measure 1 of oracles CF under the distribution defined in Definition 17, the construction Π is a WI for $\mathcal{L}$.*

Proof of Lemma 3. Let f^O be an ϵ-OWF and $\mathcal{A}$ be an adversary whose running time is bounded by a polynomial q. The completeness and the soundness can be shown in the same way as the proof of Lemma 2. Without loss of generality, we set $\epsilon = \mathrm{poly} \cdot 2^{-n}$. Then, as shown in the proof of Lemma 2, the probability that $\mathcal{A}$ succeeds to break the witness indistinguishability of Π is at most $q \cdot 2^{-12n} + \epsilon = q \cdot 2^{-12n} + \mathrm{poly} \cdot 2^{-n} < (q + \mathrm{poly}) \cdot 2^{-n} < 2^{-n/2}$. In other words, we have

$$\left| \Pr_{\mathsf{CF}_{3n}} \left[\mathtt{ExptWI}_{\mathcal{A}}^{O,\mathsf{CF}_{3n}}(n) = 1 \right] - \frac{1}{2} \right] \right| < 2^{-n/2}.$$

For any $n \in \mathbb{N}$ and any adversary $\mathcal{A}$, let $E_{n,\mathcal{A}}$ denote the event where an oracle CF_{3n} is chosen s.t.

$$\left| \Pr_{\mathsf{CF}_{3n}} \left[\mathtt{ExptWI}_{\mathcal{A}}^{O,\mathsf{CF}_{3n}}(n) = 1 \right] - \frac{1}{2} \right] \right| > 2^{-n/3}.$$

Applying an averaging argument, for any sufficiently large $n \in \mathbb{N}$ and any adversary $\mathcal{A}$, the probability that $E_{n,\mathcal{A}}$ occurs is at most $1/n^2$. Then the Borel-Cantelli Lemma

implies that the probability that $E_{n,\mathcal{A}}$ happens for infinitely many values of n is zero over the choice of CF. Therefore, for sufficiently large n and measure 1 of oracles under the distribution for coin-free proof system oracles, we have

$$\left| \Pr_n \left[\mathtt{ExptWI}_{\mathcal{A}}^{O,\mathsf{CF}_{3n}}(n) = 1 \right] - \frac{1}{2} \right] \right| \leq 2^{-n/3}.$$

We remark that q is an arbitrary polynomial. Hence, we obtain the following statement by removing measure 0 of countable many oracles for specific adversaries: For measure 1 of oracles and any PPT adversary $\mathcal{A}$, it holds that $|\Pr_n[\mathtt{ExptWI}_{\mathcal{A}}^{O,\mathsf{CF}_{3n}}(n) = 1] - 1/2| < \text{negl}$. □

The following corollary implies that our coin-free proof system oracle introduces better results than the previous WI oracle, when we consider black-box separation in the augmented black-box framework.

Corollary 1 *Let O be an oracle that implements a primitive Q,* WI *be a WI oracle and* CF *be a coin-free proof system oracle. If there exists an augmented black-box construction of a primitive P based on O and* WI*, then there exists an augmented black-box construction of P based on O and* CF*.*

We say an augmented black-box framework that takes a coin-free proof system oracle a *simplified* augmented black-box framework.

3.4 The Naor-Yung Construction

In this section, we claim that the Naor-Yung/Sahai paradigm [47, 51] falls into the simplified augmented black-box framework as well as the original augmented black-box framework. The Naor-Yung construction requires an NIZK for $\mathcal{L}' = \{(c_0, c_1, pk_0, pk_1) \mid \exists\ m, r_0, r_1 \text{ s.t.} c_0 = O(e, pk_0, m, r_0) \ \wedge\ c_1 = O(e, pk_1, m, r_1)\}$ where $O = (g, e, d)$ constitutes a CPA-PKE, pk_0 and pk_1 are public keys generated by g, m is a plaintext, c_0 and c_1 are ciphertexts and r_0 and r_1 are randomnesses. By following the construction of an NIZK in [49], we can construct an NIZK for $\mathcal{L}$ based on the WI presented in Chapter 3.3, and $\mathcal{L}'$ can be reduced to $\mathcal{L}$. Moreover we can translate the NIZK into an SS-NIZK [51] for $\mathcal{L}$. (For more discussion, see [49], Section 3.)

Lemma 4 *Let O be an oracle that implements a CPA-PKE and* CF *be a coin-free proof system oracle. We can construct a CCA-PKE based on O and* CF*.*

3.5 Impossibility of a KA from a OWF

As stated in Theorem 7, it has been shown that there exists no augmented black-box construction of a perfectly complete 1-bit KA based on a OWF and the oracle WI [49].

In the first attempt, we tried to simplify the separation proof in the presence of a coin-free proof system oracle. However, we found that the randomness of WI did not play an essential role in the proof. Hence, we confirm that randomness is not necessary for proof system oracle in the augmented black-box framework *at first place*. We first review the separation proof in [49] (in particular, the construction of the adversary), introduce notation and analyze the separation proof.

Before the detailed discussion, we formally introduce a KA as follows:

Definition 18 (Key-Agreement Protocol) *A pair (A, B) of interactive algorithms that work as follows is a* key-agreement protocol (KA)*: On input a security parameter and randomnesses r_A and r_B respectively, A and B interacts each other, and outputs a value k_A and k_B respectively. A KA is* perfectly complete *if* $\Pr[k_A = k_B] = 1$.

The set of messages sent by A and B respectively during the execution of a KA is called a *transcript*, denoted by T. An execution of a KA (A, B) on input a security parameter n and randomnesses r_A and r_B that results in a transcript T and a key k is denoted by $(T, k) \leftarrow \langle A(r_A), B(r_B)\rangle(n)$.

We treat a 1-bit KA (i.e., a KA such that $|k| = 1$), and thus we define a security of a 1-bit KA as follows:

Definition 19 (Security of a KA) *For any adversary $\mathcal{A}$,*

$$\Pr[(T, k) \leftarrow \langle A(r_A), B(r_B)\rangle(n) : k' = \mathcal{A}(T) \wedge k' = k] \leq 1/2 + \mathrm{negl}\,.$$

3.5.1 Previous Separation Result

We introduce the idea behind the adversary in [49]. First, they overviewed the proof by Impagliazzo and Rudich [36] as follows. Let us consider the construction of a KA based only on a random oracle O and let (A, B) be a black-box construction of a 1-bit KA with perfect completeness based on O. Given security parameter 1^n, A and B interact each other, resulting a transcript T and a shared key k. Let r_A be a random tape of A and $Q(A)$ be a set of query/answer pair that A made. A pair $(r_A, Q(A))$ is said a *view* of A. The intuition behind the adversary is that, if A and B agree on a key, then they must make the same query to O obtaining the same value (otherwise it implies that the KA is possible without O).

They built an adversary E that attacks the KA, where E is computationally unbounded but makes at most polynomially many queries. Given 1^n and T, E simulates the view of A at first, then learns about O based on the simulation. Hence, E finds the common query with exploiting its inefficiency, because the KA is perfectly complete. (See [49] for more discussion.)

However, the intuition is not the case in the augmented black-box framework. Suppose that a party, say A, obtains a proof π for an $x \in \mathcal{L}_n$, and send x and π to

B. Then B obtains the value $\mathsf{WI}(\mathsf{Vrf}, x, \pi) = 1$, while A knows the value without making the query to WI.Vrf. Hence, they can obtain the same value without common query.

To overcome this problem, they modified E to simulate the query/answer pairs made by both parties. Let $Q(B)$ be a set of query/answer pair that B makes in the execution of a KA, $Q(AB) := Q(A) \cup Q(B)$ and q be the number of queries that are made during the execution of (A, B). The adversary E works as follows.

Adversary. Let $Q(E)$ be a set of query/answer pair and K be a set of "key candidates." First E sets $Q(E) := \phi$ and $K := \phi$. Then E repeats the following $2q + 1$ times:

Simulation Phase E simulates the view of A that is consistent with T and $Q(E)$ (if there is no such view, then E aborts). Let $\hat{Q}(E)$ denote the set of the simulated query/answer pairs in this phase. Note that $\hat{Q}(E)$ is not necessary consistent with the real oracles O and WI. Following the simulated view, E outputs a key $\hat{k}$ and sets $K := K \cup \{\hat{k}\}$.

Update Phase E makes all queries in $\hat{Q}(E) \setminus Q(E)$ to O and WI, and adds the resulting query/answer pairs to $Q(E)$.

After $2q + 1$ iterations, E outputs the majority of K as a simulated shared key.

They demonstrated that, in an iteration of E, either E learns a query made by B, or E outputs a correct key. Therefore, as (A, B) makes at most q queries, taking the majority of K, E outputs a correct key with high probability. The proof will be described in Chapter 3.5.2.

3.5.2 Our Result

Preliminaries

Let O be a random oracle and CF_n be a oracle that is uniformly chosen according to the distribution defined in Definition 17. A set of queries Q *fixes* $x \in \mathcal{L}$ if either (i) there is a w and an O-query in Q s.t. $(x, w) \in R_{\mathcal{L}}$, or (ii) there is a query $\mathsf{CF}_n(\mathsf{Vrf}, x, \cdot) = 1$ in Q. A set of queries Q *fixes* $x \notin \mathcal{L}$ if there is no O-query in Q that implies $x \in \mathcal{L}$, regardless of how any of the other O-queries outside of Q are answered. By following [49], we set the normal form of a KA as follows: In case a party makes a query $\mathsf{CF}_n(\mathsf{Prv}, x, w)$, the party also makes O-queries necessary to fix $x \in \mathcal{L}$ beforehand, and in case a party receives a proof $\pi = \mathsf{CF}_n(\mathsf{Prv}, x, w)$ then the party also asks $\mathsf{CF}_n(\mathsf{Vrf}, x, \pi)$. Partial oracles that only contain the query/answer pairs in $Q(E) \vee \hat{Q}(E)$ is denoted by O' and $\mathsf{CF}'_n = (\mathsf{CF}'_n.\mathsf{Prv}, \mathsf{CF}'_n.\mathsf{Vrf})$.

The Separation Proof

In [49] they showed that the adversary in Chapter 3.5.1 breaks the KA. We give the overview of the proof. Although in [49], they dealt with some subtleties, we ignore them and focus only on essential part of the proof.

They first showed that the event Spoof defined in the proof of Lemma 2 occurs with probability at most $1/8$ throughout the execution of E, and supposed this event never occurs. Then, they defined three events concerning $\hat{Q}(E)$ and showed that the first event implies E learns at least one correct query/answer pair in $Q(AB)$ and the other events imply the first event (or such an event cannot occur). The events are as follows:

E_1: $\hat{Q}(E)$ disagrees with $Q(AB)$ on the answer to some O-, CF_n.Prv- or CF_n.Vrf-query.

E_2: There is an x s.t. $Q(AB)$ fixes $x \in \mathcal{L}$ but $\hat{Q}(E)$ fixes $x \notin \mathcal{L}$, or vice versa.

E_3: A CF'_n.Vrf-query returning 0 conflicts with O- or CF_n.Prv-queries in $Q(AB)$, or vice versa.

While events E_2 and E_3 contain some sub-cases, we do not describe the details here. It is clear that E learns at least one correct query/answer pair in $Q(AB)$ if E_1 occurs.

Moreover they proved that if none of these events occurs, then E computes a correct key, by showing the existence of oracles that result in T. Since $|Q(AB)| \leq q$, E_1 occurs at most q times. Thus at least $q + 1$ keys in K are correct keys in the final step of the attack, resulting a correct key.

Analysis of The Proof

As stated at the beginning of Chapter 3.5, we found that the randomness of WI does not play an essential role in the separation proof. That is, the events E_1, E_2 and E_3 occur even in the presence of a coin-free proof system oracle. Here we explain why these events cannot be excluded.

The event E_1 is the case that the answer of the real oracle and the simulated oracle, say O and O', differs on the same query. Because O is a random oracle, there is no guarantee that E simulates this oracle perfectly. Similarly, E_2 occurs due to the unpredictability of oracles. The third event capture the case that both parties obtain the same value without common query. Even in the presence of a coin-free proof system oracle, there is no denying that this event happens. Hence, we conclude that the randomness is not necessary for the separation proof between a KA and a OWF in the augmented black-box framework and obtain the following lemma.

Lemma 5 *Let O be a random oracle and* CF *be a coin-free proof system oracle s.t. a OWF f exists relative to O and* CF. *There is no simplified augmented black-box construction of a KA with perfect completeness based on O and CF.*

3.6 Conclusion And Future Work

In this work we introduced the coin-free proof system oracle, and showed the same construction and separation results in [49]. Regarding the separation proof, we confirmed that the coin-free proof system oracle works correctly. We hope that our simpler oracle makes it easier to prove securities in the augmented black-box framework.

There are open questions still remain. One of such question is to show other construction or separation results in the simplified augmented black-box framework, especially to known black-box separation results. Focusing on specific topic, the construction of the NIZK is based on a proof system oracle for an NP-complete language, which seems too strong. It is still debatable whether we can construct an NIZK based on a proof system oracle for more restricted language.

Chapter 4

Impossibility of NIZKs for Plaintext Equality

4.1 Introduction

In this chapter, we demonstrate the second result. That is, we show the black-box impossibility of an NIZK that proves the witness equality.

Recently, practical NIZKs have been proposed such as [3, 30, 101] and thus NIZKs are becoming building blocks for practical applications such as blockchain. There already exist NIZKs for NP complete languages based on some assumptions [2, 21, 102, 103]. However, when it comes using NIZKs as building blocks for advanced protocols, it is often assumed that NIZKs support convenient languages for the constructions [104, 105].

An *extended language* [50] (or a composite statement [106][1]) is a language that combines two languages $\mathcal{L}$ and $\hat{\mathcal{L}}$ by a logical binary operator $\diamond \in \{\wedge, \vee\}$, i.e., $\mathcal{L} \diamond \hat{\mathcal{L}}$. There are several applications that employ NIZKs for extended languages. For instance, it is often necessary that an NIZK in a cryptocurrency supports both Boolean and algebraic statements. Thus, an NIZK for an extended language is employed to meet this requirement [106, 107].

While NIZKs for extended languages are seen in the literature, to the best of our knowledge, there are few works that treat such NIZKs in a black-box manner. However, Abe *et al*. [50] initiated the study of *black-box language extension*, which investigates, given NIZKs for languages $\mathcal{L}$ and $\hat{\mathcal{L}}$ respectively, if we can construct an NIZK for $\mathcal{L} \diamond \hat{\mathcal{L}}$ in a black-box manner. For instance, given NIZKs for languages $\mathcal{L}$ and $\hat{\mathcal{L}}$ respectively, it is possible to construct an NIZK for $\mathcal{L} \wedge \hat{\mathcal{L}}$ by invoking given NIZKs in parallel. They showed the (im)possibility of NIZKs that prove extended

[1] While the terminology "composite statement" in [106] includes nested functions, we only focus on statements combined by AND or OR in this paper.

languages. However, their result tells nothing about extended languages that take account of binary relations between witnesses.

It is non-trivial whether we can construct NIZKs for extended languages that take account of witness relations in a black-box manner whereas such languages are found in the literature [47, 89, 104, 105]. For instance, it is not obvious if we can construct an NIZK for $(x \in \mathcal{L}) \wedge (\hat{x} \in \hat{\mathcal{L}}) \wedge (w = \hat{w})$ where $(x, w) \in R_{\mathcal{L}}$ and $(\hat{x}, \hat{w}) \in R_{\hat{\mathcal{L}}}$ from NIZKs for $\mathcal{L}$ and $\hat{\mathcal{L}}$ in a black-box manner. We refer to such a language as an *equality language*. Often it is not hard to construct an NIZK for an equality language in a non-black-box manner, especially by exploiting algebraic properties. For instance, the well-known Chaum-Pedersen protocol [108] constitutes an NIZK in the random oracle model for the equality language over discrete logarithms of group elements. It should also be noted that relations among witnesses might be easily proven if the underlying NIZK employs the commit-and-prove methodology (CP-NIZK) [61, 62] where commitments of witnesses are provided as parts of proofs. The CP systems, including CP-NIZKs, are popular in the literature, e.g., [107, 109, 110, 111, 112], as noted in [113].

As mentioned above, CP-NIZKs for proving the witness equality have already been proposed. However, there also exist NIZKs that do not invoke commitment schemes at all and it is open if we can construct NIZKs for proving equality languages based on such NIZKs in a black-box manner.

We study a black-box language extension to NIZKs for an equality language. We in particular focus on NIZKs for proving the equality of plaintexts embedded in a pair of ciphertexts, which is a well-studied NIZK required in the Naor-Yung construction [47], i.e., the plaintext equality language defined in Chapter 2.4. Note that zero-knowledge proofs for this language are found in the literature such as [110, 112]. We focus on such a language since the Naor-Yung construction is one of the most notable applications of an NIZK that proves an equation among witnesses. Technically, we first introduce oracles that constitute a CPA-PKE and a simulation-sound NIZK (SS-NIZK) [51] that proves the validity of a ciphertext generated by the CPA-PKE respectively. Then we show that, given these oracles, there is no black-box construction of a (standard) NIZK for the plaintext equality language. Our result suggests that we should rely on specific properties or structures of the underlying NIZKs or language to prove witness equality even at the expense of the efficiency or generality of the construction.

4.1.1 Related Work

As is already mentioned above, equality languages are seen in the literature [105, 107, 108, 109, 111, 113], and some of them deal with the plaintext equality language [47, 110, 112]. Particularly, Campanelli *et al.* [107] introduced a framework for commit-

and-prove SNARKs (CP-SNARKs). They generalized CP-SNARKs for certain basic binary relations, and the CP-SNARK that proves the equality between two committed values plays an essential role for their framework. Because SNARKs are known to be efficient NIZKs, we claim that proving equality between witnesses is meaningful for both theoretical and practical aspects. We remark that CP-SNARKs are found in the literature such as [114, 115].

Abe *et al.* [50] showed the impossibility of a black-box constructions of NIZKs for $\mathcal{L} \vee \hat{\mathcal{L}}$ (resp., SS-NIZKs for $\mathcal{L} \wedge \hat{\mathcal{L}}$) based on NIZKs (resp., SS-NIZKs) for languages $\mathcal{L}$ and $\hat{\mathcal{L}}$. In spite of such negative results, there is still room for consideration of a black-box construction of NIZKs for extended languages. We argue that our result is orthogonal to [50] as it is not trivial if an SS-NIZK for a conjunctive language implies a (standard) NIZK for a conjunctive language that takes account of a witness relation (and thus, the plaintext equality language).

4.1.2 Technical Overview

We follow the "swapping technique" that is introduced in [50] in the construction of our adversary. The idea behind the technique is the following: Let O be an oracle implemented by a uniformly chosen random injection and $\mathcal{L}^O = \{x \mid \exists\, w \text{ s.t. } x = O(w)\}$ be a language. Let $x = O(w)$ for some w and $x' \notin \mathcal{L}^O$, where $|x| = |x'|$. Suppose that we are considering a game between a challenger and an adversary, and the adversary internally simulates some oracle algorithm. When the algorithm makes a query to O on w, the adversary actually relays the query to the oracle. Even if the adversary returns x' as the answer to the algorithm (i.e., the adversary sets $x' := O(w)$), the algorithm cannot detect this swap, as O is implemented by a random injection. In other words, there must be another "correct" oracle O' that maps w to x' in the oracle distribution. Thus, the simulated algorithm runs correctly and outputs its result based on the swapped value.

Our adversary works as follows: Let M be a black-box construction of a proof system for the plaintext equality language that is complete and zero-knowledge. Given a CRS in a soundness game, the adversary runs the prover algorithm of M on a false statement. The adversary cheats the prover by following the above swapping technique, and finally the prover outputs a (forged) proof. This proof should pass the verification by M since it is generated by the prover. Actually, the oracle O and the construction of the adversary are more involved. See Chapter 4.4 for more details.

4.1.3 Comparison to the Results of Abe *et al.*

We argue that the swapping technique can be applied to our problem, as both our work and [50] treat conjunctive languages, while the plaintext equality language requires the equality of witnesses behind two ciphertexts. In [50], they demonstrated that a

forged proof is indistinguishable from a real proof by showing that M.Vrf makes a query on a witness only with negligible probability. We take a step further and prove that the absence of such a query causes M.Vrf not to validate the witness equality.

More technically, our adversary differs from in [50] in terms of the soundness game it participates. Recall that we consider the standard soundness game whereas they considered the unbounded simulation-soundness game. In [50], the adversary simulates the oracle answers based on a list of query/answer pairs that the challenger gives to the adversary, and they put an assumption, named the "full verification model," that the list contains sufficient information for the simulation. On the contrary, we consider the standard soundness game so that the adversary's behavior becomes simpler and we do not put any assumptions such as the full verification model.

4.2 Basic Notation

In this chapter, we introduce notations that are necessary for this work. We formally define an extended language.

Definition 20 (Extended Language [50]) *Let $\mathcal{L}$ and $\hat{\mathcal{L}}$ be languages, and let $\diamond \in \{\vee, \wedge\}$ denotes a logical binary operator. An* extended language *is defined as the union $\bigcup_n(\mathcal{L}_n \diamond \hat{\mathcal{L}}_n)$ where $\mathcal{L}_n \diamond \hat{\mathcal{L}}_n := \{(x, \hat{x}) \,|(x \in \mathcal{L}_n) \diamond (\hat{x} \in \hat{\mathcal{L}}_n)\}$. An extension is non-trivial if $\mathcal{L}_n \diamond \hat{\mathcal{L}}_n \notin \mathcal{L}_{n'}$ for any n and n'.*

If a statement $(x, \hat{x})$ for an extended language $\mathcal{L} \diamond \hat{\mathcal{L}}$ satisfies $x \in \mathcal{L}$ and $\hat{x} \in \hat{\mathcal{L}}$, we say such a statement is an (yes, yes)-instance, and define other such statements in the obvious way.

Definition 21 (Extended Language with Witness Relation) *Let $\mathcal{L}$ and $\hat{\mathcal{L}}$ be languages,* R *be a binary relation and $\diamond \in \{\vee, \wedge\}$ denotes a logical binary operator. An* extended language with witness relation R *is defined as the union $\bigcup_n(\mathcal{L}_n \diamond \hat{\mathcal{L}}_n)$, for any n and n',*

$$\mathcal{L}_n \diamond \hat{\mathcal{L}}_n := \{(x, \hat{x}) \,|\exists\, w, \hat{w} \text{ s.t. } \{(x, w) \in R_{\mathcal{L}_n} \diamond (\hat{x}, \hat{w}) \in R_{\hat{\mathcal{L}}_n}\} \wedge \mathsf{R}(w, \hat{w}) = 1\}.$$

The extension is non-trivial if $\mathcal{L}_n \diamond \hat{\mathcal{L}}_n \notin \mathcal{L}_{n'}$ for any n and n'.

In this work we make use of the CPA-PKE oracle defined by Gertner *et al.* [46].

Definition 22 (The CPA-PKE Oracle [46]) *Let $O = (O.g, O.e, O.d, O.w, O.u)$ be an oracle that is chosen uniformly according to the following distributions.*

$O.g$**:** $pk \leftarrow O(g, sk)$
Given a secret key $sk \in \{0,1\}^n$*, output a public key* $pk \in \{0,1\}^{3n}$ *where* $O.g$ *is a random injection.*

$O.e$**:** $c \leftarrow O(e, pk, b, r)$
Given a public key $pk \in \{0,1\}^{3n}$*, a message bit* $b \in \{0,1\}$ *and a randomness* $r \in \{0,1\}^n$*, output a ciphertext* $c \in \{0,1\}^{3n}$ *where* $O(e, pk, \cdot, \cdot)$ *is a random injection for any* $pk \in \{0,1\}^{3n}$*.*

$O.d$**:** $\{0, 1, \bot\} \leftarrow O(d, sk, c)$
Given a secret key $sk \in \{0,1\}^n$ *and a ciphertext* $c \in \{0,1\}^{3n}$*, output a bit* $b \in \{0,1\}$ *if there exists a randomness* r *s.t.* $O(e, O(g, sk), b, r) = c$*; otherwise output* $\bot$*.*

$O.w$**:** $O(e, pk, sk_1, r_{pk,1,j}), \cdots, O(e, pk, sk_n, r_{pk,n,j}) \leftarrow O(w, pk, j)$
Given a public key $pk \in \{0,1\}^{3n}$ *and an index* $j \in \{0,1\}^n$*, output* $\bot$ *if* $O(g^{-1}, pk) = \bot$ [2]*; otherwise output* $O(e, pk, sk_1, r_{pk,1,j}), \cdots, O(e, pk, sk_n, r_{pk,n,j})$ *where* $(sk_1, \cdots, sk_n) = sk = O(g^{-1}, pk)$ *and* $r_{pk,k,j}$ *are uniformly chosen from* $\{0,1\}^n$ *for any* $1 \leq k \leq n$*.*

$O.u$**:** $\{\top, \bot\} \leftarrow O(u, pk, c)$
Given a public key $pk \in \{0,1\}^{3n}$ *and a ciphertext* $c \in \{0,1\}^{3n}$*, output* $\top$ *if there exist an* $sk \in \{0,1\}^n, b \in \{0,1\}$ *and* $r \in \{0,1\}^n$ *s.t.* $O(g, sk) = pk$ *and* $O(e, pk, b, r) = c$*; otherwise* $\bot$*.*

We denote by O_n the set of all oracles that satisfy the above syntax for a security parameter n.

Theorem 9 (The Construction of a CPA-PKE [46]) *Let* $O = (O.g, O.e, O.d, O.w, O.u)$ *be an oracle uniformly chosen from the distribution defined in Definition 22. Then, the following construction* Π *is a CPA-PKE.*

$\Pi^O.\mathsf{Key}$**:** $pk^* \leftarrow \Pi^O(\mathsf{Key}, sk^*)$
Given a secret key $sk^* \in \{0,1\}^n$*, compute* $pk \leftarrow O(g, sk^*)$ *and output* sk^* *and* $pk^* := pk$*.*

$\Pi^O.\mathsf{Enc}$**:** $c^* \leftarrow \Pi^O(\mathsf{Enc}, pk^*, b, r)$
Given a public key $pk^* = pk$*, a message bit* $b \in \{0,1\}$ *and a randomness* $r \in \{0,1\}^n$*, compute* $c = O(e, pk, b, r)$ *and output* $c^* := c$*.*

$\Pi^O.\mathsf{Dec}$**:** $b \leftarrow \Pi^O(\mathsf{Dec}, sk^*, c^*)$
Given a secret key $sk^* = sk$ *and a ciphertext* $c^* = c$*, output* $b = O(d, sk, c)$ *where it might be the case that* $b = \bot$*.*

[2] Since $O.g$ is injective, $O.g^{-1}$ is uniquely defined.

It is known that there is no black-box construction of a CCA-PKE based on O in a bit restricted model named the shielding model [46].

The Naor-Yung Construction
Recall that the Naor-Yung construction requires an NIZK for the following plaintext equality language:

$$\mathcal{L}_{EQ}^{\Pi} = \{(c_0, c_1, pk_0, pk_1) \mid \exists m, r_0, r_1 \text{ s.t. } c_0 = \Pi(\mathsf{Enc}, pk_0, m, r_0) \\ \wedge c_1 = \Pi(\mathsf{Enc}, pk_1, m, r_1)\}.$$

where $\Pi = (\Pi.\mathsf{Key}, \Pi.\mathsf{Enc}, \Pi.\mathsf{Dec})$ is a CPA-PKE (for the formal definition, see Chapter 2.4).

We remark that $\mathcal{L}_{EQ}^{\Pi}$ is an extended language with witness relation which is defined in Definition 21 since it requires that plaintexts behind two ciphertexts are the same. We sometimes omit the description "with respect to Π" if it is clear from the context. Note that $\mathcal{L}_{EQ}^{\Pi}$ is a non-trivial extended language.

4.3 An NIZK Oracle for a Single Ciphertext Language

This section introduces an oracle s.t. given a CPA-PKE oracle and a pair of a public key and a ciphertext, the oracle constitutes an SS-NIZK that proves the validity of the ciphertext. Recall that our main purpose is to show the impossibility of the construction of an NIZK for the plaintext equality language from a CPA-PKE and an SS-NIZK for the CPA-PKE. As a first step, we show the existence of a CPA-PKE and an SS-NIZK respectively, based on the CPA-PKE oracle introduced in Definition 22 and an NIZK oracle that we introduce in this chapter.

Before introducing our NIZK oracle, we formally define the language that the oracle is able to prove and give an intuition behind the oracle. Let $\Pi = (\Pi.\mathsf{Key}, \Pi.\mathsf{Enc}, \Pi.\mathsf{Dec})$ be a CPA-PKE. A *single ciphertext language* with respect to Π is

$$\mathcal{L}_{CPA}^{\Pi} := \{(c, pk) \mid \exists b, r \text{ s.t. } c = \Pi(\mathsf{Enc}, pk, b, r)\}.$$

Now, we define the NIZK oracle. Our oracle has several interfaces that almost constitute the functionalities of an NIZK. Namely, the CRS generator and the prover interfaces are implemented by random injections H_{crs} and H_{prf} respectively, where these random injections work only when valid inputs are given to the interfaces. We guarantee the soundness of a proof by constructing the prover interface so that it works only when it is given a correct witness or a trapdoor.

Definition 23 (An NIZK Oracle) *Let O be a CPA-PKE oracle chosen uniformly from the distribution defined in Definition 22 and $H_{\mathsf{crs}} : \{0,1\}^n \to \{0,1\}^{2n}$ and $H_{\mathsf{prf}} : \{0,1\}^{8n} \to \{0,1\}^{9n}$ be random injections. An NIZK oracle* $\mathsf{ZK} = (\mathsf{ZK.Crs}, \mathsf{ZK.Prv},$

$\mathsf{ZK.PrvSim}, \mathsf{ZK.Vrf}$) *is equipped with* H_{crs} *and* H_{prf}*, with oracle access to* O *(we omit superscript* O *for legibility), and provides four functionalities:*

$\mathsf{ZK.Crs}$**:** $\sigma \leftarrow \mathsf{ZK}(\mathsf{Crs}, \tau)$
Given a trapdoor $\tau \in \{0,1\}^n$*, output a CRS* $\sigma \leftarrow H_{\mathsf{crs}}(\tau)$*.*

$\mathsf{ZK.Prv}$**:** $\{\pi, \perp\} \leftarrow \mathsf{ZK}(\mathsf{Prv}, \sigma, (c, pk), b, r)$
Given a CRS $\sigma \in \{0,1\}^{2n}$*, a statement* (c, pk) *where* $c, pk \in \{0,1\}^{3n}$*, a bit* $b \in \{0,1\}$ *and a randomness* $r \in \{0,1\}^n$*, output* $\perp$ *if* $\perp \leftarrow H_c^{-1}(\sigma), \perp \leftarrow O(g^{-1}, pk)$ *or* $c \neq O(e, pk, b, r)$*, otherwise output* $\pi \leftarrow H_{\mathsf{prf}}(\sigma||c||pk)$*.*

$\mathsf{ZK.PrvSim}$**:** $\{\pi, \perp\} \leftarrow \mathsf{ZK}(\mathsf{PrvSim}, \sigma, (c, pk), \tau)$
Given a CRS $\sigma \in \{0,1\}^{2n}$*, a statement* (c, pk) *where* $c, pk \in \{0,1\}^{3n}$ *and a trapdoor* $\tau \in \{0,1\}^n$*, output* $\perp$ *if* $\tau \nleftarrow H_c^{-1}(\sigma)$ *or* $\perp \leftarrow O(g^{-1}, pk)$*, otherwise output* $\pi \leftarrow H_{\mathsf{prf}}(\sigma||c||pk)$*.*

$\mathsf{ZK.Vrf}$**:** $\{0,1\} \leftarrow \mathsf{ZK}(\mathsf{Vrf}, \sigma, (c, pk), \pi)$
Given a CRS $\sigma \in \{0,1\}^{2n}$*, a statement* (c, pk) *where* $c, pk \in \{0,1\}^{3n}$ *and a proof* $\pi \in \{0,1\}^{9n}$*, output 1 if* $(\sigma||c||pk) \leftarrow H_p^{-1}(\pi)$*, otherwise 0.*

We let ZK_n *denote the set of all oracles that satisfy the above syntax for a security parameter* n*.*

Given σ without τ, the validity of σ can be checked easily by making a query $\mathsf{ZK}(\mathsf{Prv}, \sigma, (c, pk), b, r)$ if the query $O(e, pk, b, r) = c$ has been made previously.

We first show the construction of a CPA-PKE under the existence of O and ZK. It has been already proven that the construction Π in Theorem 9 is a CPA-PKE under the existence of O [46]. Hence, it is sufficient to show that queries to ZK does not break the CPA security of Π.

Lemma 6 *Let* O *be a CPA-PKE oracle chosen uniformly from the distribution defined in Definition 22 and* ZK *be an NIZK oracle chosen uniformly from the distribution in Definition 23. Then the construction* Π *in Theorem 9 is CPA secure under the existence of* O *and* ZK*.*

Proof of Lemma 6. What we are to prove is that there is no PPT adversary $\mathcal{A}^{O,\mathsf{ZK}}$ that breaks the CPA security of Π. It is sufficient for the lemma to show the following claim.

Claim 1 *If there exists an adversary* $\mathcal{A}^{O,\mathsf{ZK}}$ *that breaks the CPA security of* Π*, then an adversary* $\hat{\mathcal{A}}^O$ *that breaks CPA security of* Π *can be constructed from* $\mathcal{A}$*.*

Proof of Claim 1. As the existence of an adversary that breaks the CPA security of Π only with accessing O contradicts the result in [46], the above claim implies that there is no adversary $\mathcal{A}^{O,\mathsf{ZK}}$ that breaks CPA security of Π. In what follows we show that a PPT with oracle access to O can simulate ZK within negligible difference. Let q be a bound for running time of a PPT. We assume that, for ease of discussion, $\mathcal{A}$ is stateful.

We first define two bad events and prove that these events occur only with negligible probability. Then, under the assumption that these events never happen, we show that ZK can be simulated by an oracle PPT $\hat{\mathcal{A}}^{O}$ within negligible difference.

Let `BadCrs` be an event that a PPT adversary $\mathcal{A}^{O,\mathsf{ZK}}$ makes a query on a legitimate CRS σ without prior generation by $\mathsf{ZK.Crs}$. Similarly, let `BadProof` be an event that a PPT adversary $\mathcal{A}^{O,\mathsf{ZK}}$ makes a query on a legitimate proof π without prior generation by $\mathsf{ZK.Prv}$ or $\mathsf{ZK.PrvSim}$.

We first evaluate `BadCrs`. Recall that $\mathsf{ZK.Crs}$ is implemented by a random injections $H_{\mathsf{crs}} : \{0,1\}^n \to \{0,1\}^{2n}$. Considering the domain of H_{crs}, the probability that a CRS which is not generated by $\mathsf{ZK.Crs}$ is legitimate is at most $2^n/2^{2n} = 1/2^n$. As $\mathcal{A}$ makes at most q queries, the probability that this event occurs is at most $q/2^n$, which is negligible.

`BadProof` can be evaluated in the same way as `BadCrs`. Since both of $\mathsf{ZK.Prv}$ and $\mathsf{ZK.PrvSim}$ are implemented by a random injection $H_{\mathsf{prf}} : \{0,1\}^{8n} \to \{0,1\}^{9n}$, this event happens with probability at most $q/2^n$. Summarizing the above, these events occur with probability at most $2q/2^n$, which is negligible.

Now we show how a PPT with oracle access to O simulates ZK, assuming those bad events never happen.

$\mathsf{ZK}(\mathsf{Crs}, \tau)$**:** If τ is already asked, return the assigned answer. Otherwise, pick a CRS $\sigma \in \{0,1\}^{2n}$ uniformly and output σ as an answer.

$\mathsf{ZK}(\mathsf{Prv}, \sigma, (c, pk), b, r)$**:** If either $\pi = \mathsf{ZK}(\mathsf{Prv}, \sigma, (c, pk), b, r)$ or $\pi = \mathsf{ZK}(\mathsf{PrvSim}, \sigma, (c, pk), \tau)$ is already answered, return π. Note that $\mathsf{ZK}(\mathsf{Prv}, \sigma, (c, pk), [b], [r]) = \mathsf{ZK}(\mathsf{PrvSim}, \sigma, (c, pk), [\tau])$ since both $\mathsf{ZK.Prv}$ and $\mathsf{ZK.PrvSim}$ return the same proof $\pi = H_{\mathsf{prf}}(\sigma||c||pk)$. Further, we do not care if a query to $O.e$ that results in c has been made previously as we are only focusing on how to simulate ZK. Output $\perp$ if the query that results in σ is not made previously. Otherwise, pick a proof $\pi \in \{0,1\}^{9n}$ uniformly and output π as an answer.

$\mathsf{ZK}(\mathsf{PrvSim}, \sigma, (c, pk), \tau)$**:** If either $\pi = \mathsf{ZK}(\mathsf{Prv}, \sigma, (c, pk), b, r)$ or $\pi = \mathsf{ZK}(\mathsf{PrvSim}, \sigma, (c, pk), \tau)$ is already answered, return π. (The fact that π is already answered indicates $\sigma = \mathsf{ZK}(\mathsf{Crs}, \tau)$ is already asked.) Output $\perp$ if the query that results in σ is not asked previously. Otherwise, pick a proof $\pi \in \{0,1\}^{9n}$ uniformly and output π as an answer.

$\mathsf{ZK}(\mathsf{Vrf}, \sigma, (c, pk), \pi)$**:** If $b = \mathsf{ZK}(\mathsf{Vrf}, \sigma, (c, pk), \pi)$ is already asked, output the answer b. Output $\perp$ if the queries that result in σ or π are not asked previously. Output 1 if a query to ZK.Prv or ZK.PrvSim that results in π has been made previously, otherwise 0.

The above simulation is correct unless `BadCrs` and `BadProof` happen. Hence, if there exists an adversary $\mathcal{A}^{O,\mathsf{ZK}}$ that breaks the CPA security of Π with non-negligible advantage, then we can construct an adversary $\hat{\mathcal{A}}^{O}$ that breaks the CPA security of Π with non-negligible advantage, which concludes the claim.

Summarizing the above, we conclude that Π is a CPA-PKE under the existence of O and ZK, which justifies Claim 1, and thus Lemma 6. □

The Construction of an SS-NIZK

Now, we present the construction of an SS-NIZK based on the NIZK oracle.

Lemma 7 *Let O be a CPA-PKE oracle chosen uniformly from the distribution defined in Definition 22, ZK be an NIZK oracle chosen uniformly from the distribution in Definition 23 and Π be the CPA-PKE in Theorem 9. The following construction L is an SS-NIZK for $\mathcal{L}^{\Pi}_{CPA}$.*

$\mathsf{L}^{O,\mathsf{ZK}}.\mathsf{Crs}$**:** $\sigma \leftarrow \mathsf{L}^{O,\mathsf{ZK}}(\mathsf{Crs}, 1^n)$
Given a security parameter, choose a trapdoor $\tau \leftarrow \{0,1\}^n$ and output a CRS $\sigma \leftarrow \mathsf{ZK}(\mathsf{Crs}, \tau)$.

$\mathsf{L}^{O,\mathsf{ZK}}.\mathsf{Prv}$**:** $\pi \leftarrow \mathsf{L}^{O,\mathsf{ZK}}(\mathsf{Prv}, \sigma, (c, pk), b, r)$
Given a CRS σ, a statement (c, pk), a bit b and a randomness r, output $\pi \leftarrow \mathsf{ZK}(\mathsf{Prv}, \sigma, (c, pk), b, r)$.

$\mathsf{L}^{O,\mathsf{ZK}}.\mathsf{Vrf}$**:** $b \leftarrow \mathsf{L}^{O,\mathsf{ZK}}(\mathsf{Vrf}, \sigma, (c, pk), \pi)$
Given a CRS σ, a statement (c, pk) and a proof π, output a bit $b \leftarrow \mathsf{ZK}(\mathsf{Vrf}, \sigma, (c, pk), \pi)$.

$\mathsf{L}^{O,\mathsf{ZK}}.\mathsf{CrsSim}$**:** $(\tau, \sigma) \leftarrow \mathsf{L}^{O,\mathsf{ZK}}(\mathsf{CrsSim}, 1^n)$
Given a security parameter, choose a trapdoor $\tau \leftarrow \{0,1\}^n$, and output τ and a CRS $\sigma \leftarrow \mathsf{ZK}(\mathsf{Crs}, \tau)$.

$\mathsf{L}^{O,\mathsf{ZK}}.\mathsf{PrvSim}$**:** $\pi \leftarrow \mathsf{L}^{O,\mathsf{ZK}}(\mathsf{PrvSim}, \sigma, (c, pk), \tau)$
Given a CRS σ, a statement (c, pk) and a trapdoor τ, output $\pi \leftarrow \mathsf{ZK}(\mathsf{PrvSim}, \sigma, (c, pk), \tau)$.

Proof of Lemma 7. In the following, we omit the superscript O, ZK for simplicity. The completeness is immediate. We show L is zero-knowledge. The difference between L.Crs and L.CrsSim is only their interface. Further, for any σ and (c, pk) s.t. $pk \leftarrow O(g, [sk])$, $c \leftarrow O(e, pk, [b], [r])$ and $\sigma \leftarrow \mathsf{ZK}(\mathsf{Crs}, [\tau])$, it holds

that $\mathsf{ZK}(\mathsf{Prv}, \sigma, (c, pk), b, r) = \mathsf{ZK}(\mathsf{PrvSim}, \sigma, (c, pk), \tau)$. Therefore, for any PPT $\mathcal{A}$, $\mathtt{AdvZK}_{\mathcal{A},\mathsf{L},\mathcal{L}^{\Pi}_{CPA}}(n) = 0$.

To show the simulation soundness, we show that any PPT adversary $\mathcal{A}^{O,\mathsf{ZK}}$, whose running time is bounded by a polynomial q, has negligible advantage in the experiment $\mathtt{ExptSS}_{\mathcal{A},\mathsf{L},\mathcal{L}^{\Pi}_{CPA}}$. In order to break the simulation-soundness, $\mathcal{A}$ should create a forgery proof π' for a pair $(c', pk') \notin \mathcal{L}^{\Pi}_{CPA}$. Since the proof is computed by the random injection H_{prf} and $\mathcal{L}^{\Pi}_{CPA}$ is with respect to O, $\mathcal{A}$ has to make a successful query to forge the proof. That is, (i) making a query to $\mathsf{ZK.Prv}$ on (c', pk') that returns π', (ii) making a query to $\mathsf{ZK.PrvSim}$ on (c', pk') returning π', or (iii) making a query to $\mathsf{ZK.Vrf}$ on (c', pk') and π' that returns 1.

The first case is useless as $\mathsf{ZK.Prv}$ outputs $\perp$ for any $(c', pk') \notin \mathcal{L}^{\Pi}_{CPA}$. In the second case, to make a successful query to $\mathsf{ZK.PrvSim}$ on σ, $\mathcal{A}$ is required to find the trapdoor τ of σ. As $\mathsf{ZK.CrsSim}$ is implemented by the random injection H_{crs}, all $\mathcal{A}$ can do is to make a query to $\mathsf{ZK.PrvSim}$ to find the uniformly chosen trapdoor $\tau \in \{0,1\}^n$. However, the probability that such a τ is the trapdoor of σ is at most $1/2^n$. Hence, taking union bound on at most q queries, the probability that $\mathcal{A}$ succeeds to output a forgery proof is at most $q/2^n$, which is negligible.

In the third case, $\mathcal{A}$ seeks for π' s.t. $1 \leftarrow \mathsf{ZK}(\mathsf{Vrf}, \sigma, (c', pk'), \pi')$. Similar to the previous case, the probability that such a π' is in the domain of H_{prf} is at most $2^{8n}/2^{9n} = 1/2^n$. Thus $\mathcal{A}$ succeeds to output a forgery proof in the third strategy with probability at most $q/2^n$. To summarize the above, the probability that $\mathcal{A}$ succeeds to output a forgery proof is at most $2q/2^n$, which is negligible. □

4.4 Separation

This chapter presents a negative result on constructions of (standard) NIZKs for the plaintext equality language. That is, there is no fully black-box construction of an NIZK for the plaintext equality language based on a CPA-PKE Π and an SS-NIZK for $\mathcal{L}^{\Pi}_{CPA}$. We note that, as mentioned earlier, our result is not obtained trivially from the result of Abe *et al.* [50]. This chapter is devoted to prove the following theorem.

Theorem 10 *There does not exist a fully black-box construction* M *that converts any SS-NIZK for* $\mathcal{L}^{\Pi}_{CPA}$ *with respect to a CPA-PKE* Π *into an NIZK for* $\mathcal{L}^{\Pi}_{EQ}$ *that is complete, adaptive zero-knowledge and sound.*

For proving the theorem, it suffices to show the absence of a black-box construction of an NIZK for $\mathcal{L}^{\Pi}_{EQ}$ from a specific CPA-PKE Π and an NIZK for $\mathcal{L}^{\Pi}_{CPA}$, as we are concerned about fully black-box separation. Thus, we prove the theorem with respect to the CPA-PKE Π introduced in Theorem 9. We assume the existence of a construction M that is complete and zero-knowledge for $\mathcal{L}^{\Pi}_{EQ}$ based on O and ZK,

which are uniformly chosen from distributions introduced in Definition 22 and Definition 23 respectively. Then, we construct an adversary that breaks the soundness of M. Hence, we show the following lemma to prove our main theorem.

Lemma 8 *Let O and ZK be oracles that are uniformly chosen from the distributions introduced in Definition 22 and Definition 23 respectively, where ZK is chosen with respect to O. Furthermore, let Π be a CPA-PKE introduced in Theorem 9 and M be a black-box construction of a proof system for $\mathcal{L}_{EQ}^{\Pi}$ based on O and ZK that is complete and adaptive black-box zero-knowledge, which is based on O and ZK. Then there exists a polynomial time adversary $\mathcal{A}$ that breaks the soundness of M.*

Proof of Lemma 8. Before exhibiting our adversary, we set terminologies. In the following, we omit the superscript O, ZK if it is unnecessary. Let $q = \mathrm{poly}(n)$ be the maximum number of queries that M makes to O and ZK in its execution. We construct an adversary $\mathcal{A}^{O,\mathsf{ZK}}$ that attacks the soundness of M in the soundness game. We note that, while O and ZK constitute an SS-NIZK for $\mathcal{L}_{CPA}^{\Pi}$ based on O and ZK, we do not require M to have the simulation soundness. Thus, the soundness game we will consider is not $\mathtt{ExptSS}$, but a simpler one: A challenger outputs a CRS $\hat{\sigma}$. Given $\hat{\sigma}$, an adversary generates a forgery proof $\hat{\pi}$ for a statement $\hat{x} \notin \mathcal{L}_{EQ}^{\Pi}$, and sends $\hat{x}$ and $\hat{\pi}$ to the challenger. If the challenger accepts $\hat{\pi}$, then the adversary wins the game.

Overview and idea for an adversary. We construct an adversary that simulates the prover algorithm $\mathsf{M}^{O,\mathsf{ZK}}.\mathsf{Prv}$ on a false statement, and fools the prover as if it were a correct statement. Given $\hat{\sigma}$, the adversary computes two public keys pk_1 and pk_2 and computes ciphertexts $c_1 = O(e, pk_1, b, r_1)$, $c_2 = O(e, pk_2, b, r_2)$ and $c_1' = O(e, pk_1, 1-b, r_1)$ for a message bit b and randomnesses r_1 and r_2. Then, $\mathcal{A}$ runs $\mathsf{M}.\mathsf{Prv}$ on $\hat{\sigma}$, $\hat{x} = (c_1', c_2, pk_1, pk_2)$ and $\hat{w} = (b, r_1, r_2)$. We remark that $\mathsf{M}.\mathsf{Prv}$ is supposed to return $\perp$ on this input since $\hat{x} \notin \mathcal{L}_{EQ}^{\Pi}$. A crucial observation is that, O and ZK are equipped by random injections, and there is another pair of oracles O^* and ZK^* in the distributions O_n and ZK_n that contain entries that swap the computation results for c_1 and c_1'. For instance, there exists an oracle O^* that includes entries $O^*(e, pk, b, r; c_1')$ and $O^*(e, pk, 1-b, r; c_1)$. (Note that M is supposed to work properly with such swapped oracles as O^* is a valid oracle.) We exploit this property to construct the adversary.

Another important observation is that $\mathsf{M}.\mathsf{Prv}$ simulated by $\mathcal{A}$ does not directly access the given oracles. In other words, when $\mathsf{M}.\mathsf{Prv}$ calls an oracle, $\mathcal{A}$ forwards the query to the corresponding oracle, and returns its output to $\mathsf{M}.\mathsf{Prv}$. When $\mathsf{M}.\mathsf{Prv}$ makes inexpedient queries for the forgery, $\mathcal{A}$ swaps the answer to the other consistent one. Because $O.e$ and $\mathsf{ZK}.\mathsf{Prv}$ are implemented by random injections, $\mathsf{M}.\mathsf{Prv}$ cannot detect such swapping. To do so, the adversary defines partial oracles O' and ZK' based on the query/answer pairs that she learned when she computed c_1 and c_1'. Then, $\mathcal{A}$ runs $\mathsf{M}.\mathsf{Prv}$ with algorithms O'' and ZK'' that, roughly speaking, work as follows: If

$\mathsf{M}.\mathsf{Prv}$ makes a query that is registered in O' or ZK', then $\mathcal{A}$ returns the registered answer, otherwise $\mathcal{A}$ forwards the query to the real oracles and returns the answer. In other words, $\mathcal{A}$ runs $\mathsf{M}^{O'',\mathsf{ZK}''}.\mathsf{Prv}$ on a false statement to obtain a forged proof. Note that M with oracle access to O'' and ZK'' does not abort unless the completeness error happens since they are correct oracles in O_n and ZK_n, respectively.

Soundness Game and the Adversary

In what follows, we describe our adversary in a soundness game.

Step 1: Setup Phase
The challenger generates a CRS $\hat{\sigma} \leftarrow \mathsf{M}(\mathsf{Crs}, 1^n)$ and sends $\hat{\sigma}$ to the adversary $\mathcal{A}$.

Step 2: Forgery Phase
Given $\hat{\sigma}$, $\mathcal{A}$ samples two distinct secret keys $sk_1, sk_2 \leftarrow \{0,1\}^n$ and computes $pk_1 = O(g, sk_1)$ and $pk_2 = O(g, sk_2)$. Choose $b \leftarrow \{0,1\}$ and $r_1, r_2 \leftarrow \{0,1\}^n$, set $b' := 1 - b$ and compute $c_1 = O(e, pk_1, b, r_1)$, $c_2 = O(e, pk_2, b, r_2)$ and $c_1' = O(e, pk_1, b', r_1)$. Then, $\mathcal{A}$ defines partial oracles O' and ZK' based on the query/answer pairs she has learned where O' consists of entries $(e, pk_1, b', r_1; c_1)$, $(e, pk_1, b, r_1; c_1')$, $(d, sk_1, c_1'; b)$, and $(d, sk_1, c_1; b')$ and ZK' consists of entries $(\mathsf{Prv}, \cdot, (c_1, pk_1), b, r_1; \perp)$ and $(\mathsf{Prv}, \cdot, (c_1', pk_1), b', r_1; \perp)$. Let $\hat{x} := (c_1', c_2, pk_1, pk_2)$ and $\hat{w} := (b, r_1, r_2)$. Run $\mathsf{M}^{O'',\mathsf{ZK}''}(\mathsf{Prv}, \hat{\sigma}, \hat{x}, \hat{w})$ where O'' and ZK'' simulate O and ZK as follows:

[Algorithm O'']

- If a given query is in O', return the output that is registered in O'.
- For any other queries, forward it to O and return the output.

[Algorithm ZK'']

- If a given query is in ZK', return the answer corresponding to the query that is registered in ZK'.
- Given a query $(\mathsf{Prv}, [\sigma], (c_1', pk_1), b, r_1)$ with a legitimate σ, return $\pi \leftarrow \mathsf{ZK}(\mathsf{Prv}, \sigma, (c_1', pk_1), b', r_1)$ and record $(\mathsf{Prv}, \sigma, (c_1', pk_1), b, r_1; \pi)$ to ZK'. Recall that for a function $f : \{0,1\}^{n_1} \to \{0,1\}^{n_2}$ where $n_1 < n_2$, we say $y \in \{0,1\}^{n_2}$ is legitimate if there exists $x \in \{0,1\}^{n_1}$ s.t. $f(x) = y$.
- Given a query $(\mathsf{Prv}, [\sigma], (c_1, pk_1), b', r_1)$ with a legitimate σ, return $\pi \leftarrow \mathsf{ZK}(\mathsf{Prv}, \sigma, (c_1, pk_1), b, r_1)$ and record $(\mathsf{Prv}, \sigma, (c_1, pk_1), b', r_1; \pi)$ to ZK'.
- For any other queries, forward it to ZK and return the output.

When M outputs a proof $\hat{\pi}$, $\mathcal{A}$ sends $\hat{x}$ and $\hat{\pi}$ to the challenger.

Step 3: Verification Phase
Given $\hat{x} = (c_1', c_2, pk_1, pk_2)$ and $\hat{\pi}$, the challenger outputs 1 if $1 \leftarrow \mathsf{M}(\mathsf{Vrf}, \hat{\sigma}, \hat{x}, \hat{\pi})$. Otherwise it outputs 0.

Now, we show the adversary demonstrated in the above soundness game breaks the soundness of M.

Evaluation of $\mathcal{A}$

Let P be the probability that the challenger outputs 1 in Step 3. The probability is taken over the choice of O, ZK and all random coins by the challenger and the adversary. Let $\mathtt{AdvZK}_{\mathcal{A}',\mathsf{M},\mathcal{L}^{\Pi}_{EQ}} \leq \rho_{\mathsf{zk}}$ for any stateful PPT adversary $\mathcal{A}'$, and $\Pr[\mathsf{Vrf}(\sigma, x, \mathsf{Prv}(\sigma, x, w)) = 1] \geq 1 - \rho_{\mathsf{co}}$ for any $n \in \mathbb{N}$, any $\sigma \leftarrow \mathsf{Crs}(1^n)$ and any $(x, w) \in R_{\mathcal{L}^{\Pi}_{EQ}}$. As M is complete and zero-knowledge, ρ_{co} and ρ_{zk} are negligible in n. We let $|\Pr[\mathtt{ExptPKE}^{cpa}_{\mathcal{A}'',\Pi} = 1] - 1/2| \leq \rho_{\mathsf{cpa}} = \mathrm{negl}(n)$ for any PPT adversary $\mathcal{A}''$, which is taken over the choice of oracles and randomness in the experiment. We assume that M makes at most $q = \mathrm{poly}(n)$ queries.

In order to prove the lemma, we show that P is non-negligible by considering a sequence of games s.t. the final game introduces the situation that the challenger outputs 1 trivially. Let P_i denotes the probability that the challenger outputs 1 in $\mathsf{Game}\ i$. In games from $\mathsf{Game}\ 1$ to $\mathsf{Game}\ 4$, we exclude some bad events that happen only by accident and simplify the game. In the remaining games, we replace oracle O and ZK with O'' and ZK'' respectively step by step, finally reaching the situation that $\mathsf{M.Vrf}$ always accepts the proof unless the completeness error occurs. Again, the intuition behind the analysis is that, swapping oracle answers only results in the other correct oracle, as we are considering oracles that are implemented by random injections.

$\mathsf{Game}\ 0$: The above soundness game. Thus $P_0 = P$.

$\mathsf{Game}\ 1$: The game halts if one of the following events occurs:

- A successful query that includes a legitimate σ is made without a prior query that results in σ.
- A successful query that includes a legitimate π is made to $\mathsf{ZK.Vrf}$ without a prior query that generates π.

In other words, the above cases exclude the events that the challenger and the adversary find a legitimate CRS or a legitimate proof by chance. A query on σ without prior generation is successful only if σ is in the domain of H_{crs}. Thus, considering the domain of H_{crs}, it happens with probability at most $2^n/2^{2n} = 1/2^n$. Similarly, any query to $\mathsf{ZK.Vrf}$ on π without prior generation by $\mathsf{ZK.Prv}$ or $\mathsf{ZK.PrvSim}$ outputs 1 with probability at most $2^{8n}/2^{9n} = 1/2^n$. Since at most $3q$ queries can be made throughout the soundness game by M, there is at most $3q(1/2^n + 1/2^n) = 6q/2^n$ chance of halting the game by observing the above events. Thus we have $|P_1 - P_0| \leq 6q/2^n$. In what

follows, we consider probabilities which are conditioned on the events do not occur. We will exclude certain events in Game 2-4 as well, and treat the probabilities in the same manner.

Game 2: The game halts if one of the following events occurs:

- A query that includes a legitimate public key pk is made without a prior query to $O.g$ that results in pk.
- A query that includes a legitimate ciphertext c is made without a prior query to $O.e$ or $O.w$ that results in c.

The first event describes the situation where a uniformly chosen public key is in the range of g. Considering the domain of g, this event happens with probability at most $2^n/2^{3n} = 1/2^{2n}$. We evaluate the second case as follows. Recall that for a public key pk, the encryption oracle $O(e, pk, \cdot, \cdot)$ constitutes a random injection $\{0,1\}^{n+1} \to \{0,1\}^{3n}$. Hence, even if one knows a legitimate public key pk, the probability that he generates a legitimate ciphertext without making a query to $O.e$ or is at most $2^{n+1}/2^{3n} = 2/2^{2n}$. Similarly, as $O.w$ outputs n ciphertexts, the probability that a legitimate ciphertext without making a query to $O.w$ or is at most $n \cdot 2^{n+1}/2^{3n} = 2n/2^{2n}$. As at most $3q$ queries are made by M in the soundness game, we have $|P_2 - P_1| \leq 3q(1/2^{2n} + 2/2^{2n} + 2n/2^{2n}) = (6n+9)q/2^{2n}$.

Game 3: The game halts if a query that contains one of the following values is made in Step 1; sk_1, sk_2, pk_1, pk_2, c_1, c'_1, c_2, r_1 and r_2. Note that this event can be defined after the adversary in Step 2 obtains these values.

This avoids the challenger to learn the secrets beforehand. Since the secret keys and the randomnesses are chosen uniformly and $O.g$ and $O.e$ are random injections, the challenger finds these values only by chance. We evaluate the probability to make a query on each value as follows.

The secret keys: Considering the domain of $O.g$, the probability that making queries on a specific secret key is at most $1/2^n$. Hence, given at most q queries in an execution of M, the probability that making a query on sk_1 or sk_2 is bounded by $q(1/2^n + 1/2^n) = 2q/2^n$. The remaining cases are evaluated in the same way.

The public keys: The probability that a query on a specific public key is made without prior generation is at most $1/2^{3n}$. We remark that the situation making a query on sk_1 or sk_2 to $O.g$ to find pk_1 or pk_2 is already excluded by the above case. Thus, this event occurs with probability at most $q(1/2^{3n} + 1/2^{3n}) = 2q/2^{3n}$.

The ciphertexts: Similar to the above case, the probability that the challenger finds a specific ciphertext is at most $1/2^{3n}$. Taking union bound for at most q queries and three ciphertexts, a query on the ciphertexts is made with probability at most $3q/2^{3n}$.

The randomness: Since r_1 and r_2 are chosen uniformly, this event happens only by chance among at most q queries. Therefore, this event occurs with probability at most

$2q/2^n$. Summarizing the above, we have $|P_3 - P_2| \leq 4q/2^n + 5q/2^{3n}$.

Game 4: The game halts if the output of $O.w$ contains c_1, c_1' or c_2 in Step 1. We exclude this case because we cannot avoid the possibility that the challenger makes a query to $O.w$ to obtain something about the proof. Recall that the output of $O.w$ is uniformly chosen n ciphertexts. Since $O.e$ is a random injection, the probability that an output of $O.w$ contains a specific ciphertext is at most $n/2^{3n}$. Since at most q queries are made in Step 1, we have $|P_4 - P_3| \leq nq(1/2^{3n} + 1/2^{3n} + 1/2^{3n}) = 3nq/2^{3n}$.

Game 5: Replace O and ZK in Step 1 with O'' and ZK'' with partial oracles O' and ZK' defined at the end of Step 2. In other words, the randomness in the adversary is chosen at the beginning of the game. Observe that the adversary in the previous game chooses b, r_1 and r_2 in Step 2 independent of $\hat{\sigma}$. Hence, it does not affect the distribution of the output of this game if the randomness in the adversary is chosen beforehand. The view of Step 1 changes only if M makes a query defined in O' or ZK'. Observe that each query registered in O' and ZK' contains r_1 or sk_1. Hence this event does not happen as we have excluded such cases in Game 3. Thus we have $P_5 = P_4$.

Game 6: Replace O and ZK in Step 3 with O'' and ZK'' that contains the partial oracles O' and ZK' defined at the end of Step 2, respectively. Note that the view in Step 3 differs only if the challenger (i.e., M.Vrf) makes a query registered in O' or ZK'. Observe that such a query contains the secret randomness r_1, i.e., $O'(e, pk_1, b, r_1; c_1')$, $O'(e, pk_1, b', r_1; c_1)$, $\mathsf{ZK}'(\mathsf{Prv}, \cdot, (c_1, pk_1), b, r_1; \perp)$, $\mathsf{ZK}'(\mathsf{Prv}, \cdot, (c_1', pk_1), b', r_1; \perp)$, $\mathsf{ZK}'(\mathsf{Prv}, [\sigma], (c_1', pk_1), b, r_1; [\pi])$ and $\mathsf{ZK}'(\mathsf{Prv}, [\sigma], (c_1, pk_1), b', r_1; [\pi])$. Hence, it is sufficient to show the probability that the challenger makes queries on r_1 in Step 3 to evaluate the difference between Game 5 and Game 6.

Let $\mathtt{AskRnd}^i$ be an event that r_1 is asked by M.Vrf in Step 3 in Game i. Obviously, $|P_6 - P_5| \leq \Pr[\mathtt{AskRnd}^6]$. In what follows, we consider a sequence of subgames that finally reaches the situation where $\hat{\pi}$ is independent of r_1 and we can evaluate the probability that $\mathtt{AskRnd}$ occurs.

Game 6.0: The same as Game 6.

Game 6.1: We use the following oracles $\tilde{O}$ and $\tilde{\mathsf{ZK}}$ instead. Let O' and ZK' be the same partial oracles as previous game. Let $\mathsf{R}_{\mathsf{cpa}}$ and R_{zk} be uniformly chosen partial oracles so that $\tilde{O} := O'||\mathsf{R}_{\mathsf{cpa}} \in O_n$ and $\tilde{\mathsf{ZK}} := \mathsf{ZK}||\mathsf{R}_{\mathsf{zk}} \in \mathsf{ZK}_n$ where $\tilde{\mathsf{ZK}}$ constitutes an NIZK for $\mathcal{L}_{CPA}^{\tilde{O}}$ and bad events in Game 1-4 never happen in Step 1. Note that such $\tilde{O}$ and $\tilde{\mathsf{ZK}}$ must exist as O_n and ZK_n are implemented by random injections. We remark that in both Game 6.0 and 6.1, only queries outside of O' and ZK' are made as long as $\mathtt{AskRnd}$ never happens in Step 3. As these oracles in both Game 6.0 and Game 6.1 yield the same view in the soundness game, we have $\Pr[\mathtt{AskRnd}^{6.0}] = \Pr[\mathtt{AskRnd}^{6.1}]$.

Game 6.2: Choose oracles uniformly, i.e., $\tilde{O} \leftarrow O_n$ and $\tilde{\mathsf{ZK}} \leftarrow \mathsf{ZK}_n$ so that the

events defined in Game 1-4 never happen. Furthermore, we modify the adversary so that it runs M.Prv on $\hat{x} = (c_1, c_2, pk_1, pk_2)$ and $\hat{w} = (b, r_1, r_2)$. In Game 6.1, the partial oracles O' and ZK' are determined by the (random) choice of oracles and the randomness of the adversary and the remaining parts of oracles are chosen uniformly so that they are consistent to O' and ZK' respectively. Thus, the view in Game 6.2 does not differ from that in Game 6.1 while we have modified the way of choosing oracles. Therefore, we have $\Pr[\mathtt{AskRnd}^{6.2}] = \Pr[\mathtt{AskRnd}^{6.1}]$.

Game 6.3: Replace M.Crs and M.Prv in the game with M.CrsSim and M.PrvSim and let the challenger pass the trapdoor $\hat{\tau}$ generated in Step 1 to the adversary. Note that this modification does not affect the upper-bound of the probability that the events defined in Game 1-4 occur. We allow the challenger to pass $\hat{\tau}$ to the adversary because we are considering a game transition. That is, in a game transition, it is sufficient if we can compare the distribution of outputs of games. Hence, it does not matter how intermediate games of the transition are composed. We first claim the following:

Claim 2 $\Pr[\mathtt{AskRnd}^{6.2}] - \Pr[\mathtt{AskRnd}^{6.3}] \leq \rho_{\mathsf{zk}} + 2\epsilon$ *where* $\epsilon = 10q/2^n + (6n + 9)q/2^{2n} + (3n + 5)q/2^{3n}$.

Note that ϵ is the summation of the upper-bounds of the probabilities that events defined in Game 1-4 occur. The proof of this claim appears at the end of this chapter.

Now, a proof generated in Game 6.3 is independent of r_1 since it is generated by M.PrvSim. We show that if $\Pr[\mathtt{AskRnd}^{6.3}] \geq \rho_{\mathsf{cpa}} + \epsilon$, then we can construct an adversary that breaks the CPA security of $\Pi^{\tilde{O},\tilde{\mathsf{ZK}}}$. We construct a PPT adversary $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1)$ in the experiment $\mathtt{ExptPKE}^{cpa}_{\mathcal{B},\Pi}(n)$ so that $\mathcal{B}_1$ simulates Game 6.3 and outputs a correct message bit if AskRnd happens.

$\mathcal{B}_0$: Given a public key pk, output μ where μ contains pk (note that as we are considering 1-bit messages, it is not necessary to choose massages explicitly).

$\mathcal{B}_1$: Given (μ, c) where c is a ciphertext, simulate Game 6.3 as described below. Throughout the simulation, if $\mathcal{B}_1$ observes events that are defined in Game 1 or Game 2, then abort the simulation and output a random bit.

(i) Choose a trapdoor $\hat{\tau} \leftarrow \{0,1\}^n$ and run $\mathsf{M}^{\tilde{O},\tilde{\mathsf{ZK}}}(\mathsf{CrsSim}, \hat{\tau})$ to obtain $\hat{\sigma}$.

(ii) Choose $sk' \leftarrow \{0,1\}^n$ to obtain $pk' = O(g, sk')$, choose $b^* \leftarrow \{0,1\}$ uniformly and $r' \leftarrow \{0,1\}^n$ and compute $c' = O(e, pk', b^*, r')$. Halt the simulation and output a random bit b' if $\mathcal{B}_1$ observes that events defined in Game 3 or Game 4 occur.

(iii) Run $\mathsf{M}^{\tilde{O},\tilde{\mathsf{ZK}}}(\mathsf{PrvSim}, \hat{\sigma}, (c, c', pk, pk'), \hat{\tau})$ to obtain a proof $\hat{\pi}$.

(iv) Run $\mathsf{M}^{\tilde{O},\tilde{\mathsf{ZK}}}(\mathsf{Vrf},(c,c',pk,pk'),\hat{\pi})$. If $\mathsf{M}^{\tilde{O},\tilde{\mathsf{ZK}}}.\mathsf{Vrf}$ makes a query that includes some randomness r, then the adversary sees if $c = O(e,pk,b^*,r)$. If such an r is found, then abort the simulation and output $b' := b^*$, otherwise output a random bit b'.

Note that the above simulation can be done in polynomial time. Now, we have the following claim.

Claim 3 *If* $\Pr[\mathtt{AskRnd}^{6.3}] \geq \rho_{\mathsf{cpa}} + \epsilon$*, then* $|\Pr[\mathtt{ExptPKE}^{cpa}_{\mathcal{B},\Pi}(n) = 1] - 1/2| \geq \rho_{\mathsf{cpa}}$.

Proof of Claim 3. Let $b \in \{0,1\}$ be a plaintext behind c. Let $\mathtt{AskRnd}^{cpa}$ be an event that a query that contains r s.t. $c = O(e,pk,b^*,r)$ is made in (iv) and $\mathtt{Bad}$ be an event that events defined in Game 1-4 occur during the execution of $\mathcal{B}_1$. Observe that $\Pr[\mathtt{AskRnd}^{6.3}] = \Pr[\mathtt{AskRnd}^{cpa} \mid \overline{\mathtt{Bad}} \wedge b^* = b] = \Pr[\mathtt{AskRnd}^{cpa} \mid \overline{\mathtt{Bad}}]/2$. Furthermore $\Pr[b' = b \mid \mathtt{AskRnd}^{cpa}] = 1$ and $\Pr[b' = b \mid \overline{\mathtt{AskRnd}^{cpa}}] = 1/2$. Then, we have the following formula, which justifies Claim 3.

$$
\begin{aligned}
&\Pr[\mathtt{ExptPKE}^{cpa}_{\mathcal{B},\Pi}(n) = 1] = \Pr[b' = b] \\
=&\Pr[b' = b \mid \mathtt{AskRnd}^{cpa}] \cdot \Pr[\mathtt{AskRnd}^{cpa}] + \Pr[b' = b \mid \overline{\mathtt{AskRnd}^{cpa}}] \cdot \Pr[\overline{\mathtt{AskRnd}^{cpa}}] \\
=&\Pr[b' = b \mid \mathtt{AskRnd}^{cpa}] \cdot \Pr[\mathtt{AskRnd}^{cpa}] \\
&+ \Pr[b' = b \mid \overline{\mathtt{AskRnd}^{cpa}}] \cdot (1 - \Pr[\mathtt{AskRnd}^{cpa}]) \\
=&\Pr[\mathtt{AskRnd}^{cpa}] + \frac{1}{2}(1 - \Pr[\mathtt{AskRnd}^{cpa}]) = \frac{1}{2} + \frac{1}{2}\Pr[\mathtt{AskRnd}^{cpa}] \\
=&\frac{1}{2} + \frac{1}{2}(\Pr[\mathtt{AskRnd}^{cpa} \mid \mathtt{Bad}] \cdot \Pr[\mathtt{Bad}] + \Pr[\mathtt{AskRnd}^{cpa} \mid \overline{\mathtt{Bad}}] \cdot \Pr[\overline{\mathtt{Bad}}]) \\
\geq&\frac{1}{2} + \frac{1}{2}\Pr[\mathtt{AskRnd}^{cpa} \mid \overline{\mathtt{Bad}}] \cdot \Pr[\overline{\mathtt{Bad}}] = \frac{1}{2} + \Pr[\mathtt{AskRnd}^{6.3}] \cdot \Pr[\overline{\mathtt{Bad}}] \\
=&\frac{1}{2} + \Pr[\mathtt{AskRnd}^{6.3}] \cdot (1 - \Pr[\mathtt{Bad}]) \geq \frac{1}{2} + \Pr[\mathtt{AskRnd}^{6.3}] \cdot (1 - \epsilon) \\
\geq&\frac{1}{2} + \Pr[\mathtt{AskRnd}^{6.3}] - \epsilon \geq \frac{1}{2} + \rho_{\mathsf{cpa}}.
\end{aligned}
$$

□

Due to Claim 3, we obtain $\Pr[\mathtt{AskRnd}^{6.3}] \leq \rho_{\mathsf{cpa}} + \epsilon$. Summarizing the above, we have $|P_6 - P_5| \leq \Pr[\mathtt{AskRnd}^{6.0}] = \Pr[\mathtt{AskRnd}^{6.1}] = \Pr[\mathtt{AskRnd}^{6.2}] \leq \rho_{\mathsf{zk}} + 2\epsilon + \Pr[\mathtt{AskRnd}^{6.3}] \leq \rho_{\mathsf{zk}} + 2\epsilon + \rho_{\mathsf{cpa}} + \epsilon = \rho_{\mathsf{zk}} + \rho_{\mathsf{cpa}} + 3\epsilon$, which is negligible.

Game 7: Similar to the modification in Game 6.1, modify O'' and ZK'' to be $O'' := O'||\mathsf{R}_{\mathsf{cpa}}$ and $\mathsf{ZK}'' := \mathsf{ZK}'||\mathsf{R}_{\mathsf{zk}}$ with random partial oracles $\mathsf{R}_{\mathsf{cpa}}$ and R_{zk} that make O'' and ZK'' one of oracles in O_n and ZK_n respectively, so that it no longer uses O and ZK, respectively. As the same discussion in Game 6.1 can be applied, we have $P_7 = P_6$.

Now O'' and ZK'' are oracles in O_n and ZK_n respectively, that generate a correct proof on (c_1', c_2, pk_1, pk_2). Then, $\mathsf{M.Vrf}$ accepts the generated proof unless the completeness error occurs. Thus we have $P_7 > 1 - \rho_{\mathsf{co}}$.

Summarizing the above, we have $P > 1 - \rho_{\mathsf{co}} - \rho_{\mathsf{zk}} - \rho_{\mathsf{cpa}} - 4\epsilon$ where $\epsilon = 10q/2^n + (6n+9)q/2^{2n} + (3n+5)q/2^{3n}$, hence P is non-negligible. This concludes Lemma 8, thus Theorem 10. What remains is to apply a standard technique from Borel-Cantelli Lemma that allows us to choose oracles from particular distributions. □

Proof of Claim 2. To justify the claim, we construct a stateful PPT adversary $\mathcal{B}'^{\mathcal{O},\mathsf{ZK}}$ that attacks the zero-knowledgeness of M. The adversary works in a zero-knowledge game (between a challenger) as follows:

Step I : Given a CRS $\hat{\sigma}$, sample two distinct secret keys $sk_1, sk_2 \leftarrow \{0,1\}^n$ and compute $pk_1 = O(g, sk_1)$ and $pk_2 = O(g, sk_2)$. Choose $b \leftarrow \{0,1\}$ and $r_1, r_2 \leftarrow \{0,1\}^n$ and compute $c_1 = O(e, pk_1, b, r_1)$ and $c_2 = O(e, pk_2, b, r_2)$. Set $\hat{x} = (c_1, c_2, pk_1, pk_2)$ and $\hat{w} = (b, r_1, r_2)$ and output $(\hat{x}, \hat{w})$. (Note that $(\hat{x}, \hat{w})$ is chosen in the same way as Game 6.2 and Game 6.3).

Step II : Given a proof $\hat{\pi}$, run $\mathsf{M}(\mathsf{Vrf}, \hat{\sigma}, \hat{x}, \hat{\pi})$. Output $b' = 1$ if $\mathcal{B}'$ observes a query that includes r_1 during the execution of $\mathsf{M.Vrf}$, otherwise $b' = 0$. Note that since $\mathcal{B}'$ simulates $\mathsf{M.Vrf}$, $\mathcal{B}'$ fetches all queries made by $\mathsf{M.Vrf}$.

We denote the situation by $\tilde{b} = 1$ (resp., $\tilde{b} = 0$) where the challenger runs $\mathsf{M.Crs}$ and $\mathsf{M.Prv}$ (resp., $\mathsf{M.CrsSim}$ and $\mathsf{M.PrvSim}$). Let $\mathtt{BadRl}$ (resp., $\mathtt{BadSm}$) be an event that one of the events that are described in Game 1-4 occurs in the CRS generation phase of the challenger conditioned on $\tilde{b} = 1$ (resp., $\tilde{b} = 0$). We remark that, considering the upper-bounds by Game1-4, $\Pr[\mathtt{BadRl}]$ and $\Pr[\mathtt{BadSm}]$ are upper-bounded by $10q/2^n + (6n+9)q/2^{2n} + (3n+5)q/2^{3n}$ $(= \epsilon)$.

Observe that the distributions of values that are given to $\mathsf{M.Vrf}$ (i.e., $\hat{\sigma}, \hat{x}, \hat{\pi}$) in Step II are the same as those of Game 6.2 (resp., Game 6.3), if $\tilde{b} = 1$ (resp., $\tilde{b} = 0$) and $\mathtt{BadRl}$ (resp., $\mathtt{BadSm}$) does not occur. Thus, it holds that $\Pr[b' = 1 \,|\tilde{b} = 1 \wedge \overline{\mathtt{BadRl}}] = \Pr[\mathtt{AskRnd}^{6.2}]$ and $\Pr[b' = 1 \,|\tilde{b} = 0 \wedge \overline{\mathtt{BadSm}}] = \Pr[\mathtt{AskRnd}^{6.3}]$. Therefore, considering the advantage $\mathtt{AdvZK}_{\mathcal{B}',\mathsf{M},\mathcal{L}^{\Pi}_{EQ}}$, we obtain the following formula:

$$
\begin{aligned}
\rho_{\mathsf{zk}} \geq\ & \mathtt{AdvZK}_{\mathcal{B}',\mathsf{M},\mathcal{L}^{\Pi}_{EQ}} = \Pr[b'=1\,|\tilde{b}=1] - \Pr[b'=1\,|\tilde{b}=0] \\
=& \Pr[b'=1\,|\tilde{b}=1 \wedge \mathtt{BadRl}] \cdot \Pr[\mathtt{BadRl}] + \Pr[b'=1\,|\tilde{b}=1 \wedge \overline{\mathtt{BadRl}}] \cdot \Pr[\overline{\mathtt{BadRl}}] \\
& - \Pr[b'=1\,|\tilde{b}=0 \wedge \mathtt{BadSm}] \cdot \Pr[\mathtt{BadSm}] - \Pr[b'=1\,|\tilde{b}=0 \wedge \overline{\mathtt{BadSm}}] \cdot \Pr[\overline{\mathtt{BadSm}}] \\
\geq & - \Pr[b'=1\,|\tilde{b}=0 \wedge \mathtt{BadSm}] \cdot \Pr[\mathtt{BadSm}] \\
& + \Pr[b'=1\,|\tilde{b}=1 \wedge \overline{\mathtt{BadRl}}] \cdot \Pr[\overline{\mathtt{BadRl}}] - \Pr[b'=1\,|\tilde{b}=0 \wedge \overline{\mathtt{BadSm}}] \cdot \Pr[\overline{\mathtt{BadSm}}] \\
\geq & - \Pr[\mathtt{BadSm}] \\
& + \Pr[b'=1\,|\tilde{b}=1 \wedge \overline{\mathtt{BadRl}}] \cdot \Pr[\overline{\mathtt{BadRl}}] - \Pr[b'=1\,|\tilde{b}=0 \wedge \overline{\mathtt{BadSm}}] \cdot \Pr[\overline{\mathtt{BadSm}}] \\
\geq & - \epsilon \\
& + \Pr[b'=1\,|\tilde{b}=1 \wedge \overline{\mathtt{BadRl}}] \cdot \Pr[\overline{\mathtt{BadRl}}] - \Pr[b'=1\,|\tilde{b}=0 \wedge \overline{\mathtt{BadSm}}] \cdot \Pr[\overline{\mathtt{BadSm}}] \\
=& - \epsilon + \Pr[\mathtt{AskRnd}^{6.2}] \cdot \Pr[\overline{\mathtt{BadRl}}] - \Pr[\mathtt{AskRnd}^{6.3}] \cdot \Pr[\overline{\mathtt{BadSm}}] \\
=& - \epsilon + \Pr[\mathtt{AskRnd}^{6.2}] \cdot (1 - \Pr[\mathtt{BadRl}]) - \Pr[\mathtt{AskRnd}^{6.3}] \cdot (1 - \Pr[\mathtt{BadSm}]) \\
=& - \epsilon + (\Pr[\mathtt{AskRnd}^{6.2}] - \Pr[\mathtt{AskRnd}^{6.3}]) \\
& - \Pr[\mathtt{AskRnd}^{6.2}] \cdot \Pr[\mathtt{BadRl}] + \Pr[\mathtt{AskRnd}^{6.3}] \cdot \Pr[\mathtt{BadSm}] \\
\geq & - \epsilon + (\Pr[\mathtt{AskRnd}^{6.2}] - \Pr[\mathtt{AskRnd}^{6.3}]) - \Pr[\mathtt{AskRnd}^{6.2}] \cdot \Pr[\mathtt{BadRl}] \\
\geq & - 2\epsilon + (\Pr[\mathtt{AskRnd}^{6.2}] - \Pr[\mathtt{AskRnd}^{6.3}]).
\end{aligned}
$$

Therefore, we obtain $\Pr[\mathtt{AskRnd}^{6.2}] - \Pr[\mathtt{AskRnd}^{6.3}] \leq \rho_{\mathsf{zk}} + 2\epsilon$, as we claimed. □

Limitation of the Swapping Technique

Finally, we discuss the limitation of the power of the swapping technique. At the first attempt in this work, we tried to prove that it is impossible to obtain an NIZK for the witness equality from NIZKs for arbitrary non-trivial languages in a black-box manner. However, we found that there is a language class that the swapping technique does not work. That is, if multiple queries are necessary to verify the language, the verifier in the soundness game could detect the swapping.

Let O be a random oracle, $\mathcal{L} = \{x \,|\, \exists\, w \text{ s.t. } x = O(O(w))\}$, $(x, w) \in R_{\mathcal{L}}$ and $O(w) = a$. The adversary forges a proof for an invalid statement $x' \notin \mathcal{L}$ by setting $O(a) := x'$. However, if the proof generated by $\mathsf{M.Prv}$ contains a (note that such a construction does not harm the zero-knowledge property as a is output by the random oracle), the verifier could detect the swapping by making a query $x = O(a) \neq x'$. We might be able to correct the drawback with respect to the language $\mathcal{L}$, but we do not know how to deal with this type of problem generally.

Regardless of this negative observation, we argue that the swapping technique is still a powerful methodology. We remark that proving the knowledge about the language $\mathcal{L} = \{x \,|\, \exists\, w \text{ s.t. } x = O(w)\}$, where O is some oracle, captures a natural

scenario that, given an output of a cryptographic protocol, prove its validity. As the swapping technique could deal with this type of practical languages, it is still a useful tool for proving a separation in a black-box manner.

4.5 Conclusion and Open Question

In this work we uncovered the impossibility of a black-box construction of an NIZK for the plaintext equality language based on a CPA-PKE and an SS-NIZK for a single ciphertext language. However, it still remains open whether we can extend a specific NIZK to an NIZK for an equality language in a black-box way. While there are known results that employ the CP technique for an equality language [107, 109, 110, 111, 112], it is still open if such a language can be generally proven by, for instance, Quasi-Adaptive NIZKs [30].

Another important direction is showing the (im)possibility of NIZKs for witness relations other than equality. There are a lot of constructions of NIZKs for these extended languages that employ non-black-box techniques such as [104, 116, 117]. It is still debatable whether these NIZKs are possible in a black-box manner.

Chapter 5

Limits on The Power of Commit-and-Prove NIZKs

5.1 Introduction

In this chapter, we investigate the limitation on the power of NIZKs that employ the *commit-and-prove* methodology [60, 61, 62]. Roughly speaking, this technique guarantees that, given a proof and a commitment, the proof is carried out with respect to the opening of the commitment. NIZKs that employ the commit-and-prove methodology (CP-NIZKs) are seen in the literature [3, 107, 114, 115, 118]. The commit-and-prove methodology itself is of interest. For instance, as noted in [113], the commit-and-prove technique is standard when one wants to prove that the witnesses to two distinct statements are the same [107, 109, 110, 111, 112].

One of the most notable applications of CP-NIZKs is Zcash [13], which uses zk-SNARK by [5] to guarantee the anonymity of users. In fact, the zk-SNARK does not explicitly employ a commit-and-prove methodology. However, as mentioned in [119], a prover in Zcash proves knowledge about a committed value, and thus we can regard Zcash as an application of a CP-NIZK.

In Chapter 4, we showed that, given an NIZK that proves the validity of the ciphertext of a CPA-PKE, it is impossible to construct an NIZK that proves the equality of the plaintexts behind two distinct ciphertexts in a black-box manner. However, while somewhat folklore, we can construct an NIZK that proves the witness equality if the underlying NIZKs are CP-NIZKs. Suppose that we are given CP-NIZKs for distinct languages $\mathcal{L}$ and $\hat{\mathcal{L}}$ that share the same commitment scheme. Then, we can construct an NIZK for the language $\mathcal{L} = \{x, \hat{x} \mid \exists w, \hat{w} \text{ s.t. } (x, w) \in R_{\mathcal{L}} \wedge (\hat{x}, \hat{w}) \in R_{\hat{\mathcal{L}}} \wedge w = \hat{w}\}$ by executing the NIZKs on the same commitment. In other words, the commit-and-prove methodology trivially breaks the barrier presented in Chapter 4. However, it is not clear if a commit-and-prove technique overcomes the negative result for disjunc-

tive languages by [50]. Hence, the following question is still open:

Is it possible to construct an NIZK for a disjunctive language based on CP-NIZKs in a black-box manner?

We investigate the above problem and answer negatively. That is, there is no fully black-box construction of an NIZK for a disjunctive language based on CP-NIZKs. We first formalize CP-NIZKs and introduce an oracle that implements a CP-NIZK for a certain language. Then, we demonstrate a polynomial-time adversary that attacks the soundness of an NIZK for a disjunctive language.

Specifically, let O be a certain oracle and ZK be an oracle that implements a CP-NIZK for an oracle-relativized language, denoted by $\mathcal{L}^O$. Assume that there exists a black-box construction $\mathsf{M}^{O,\mathsf{ZK}}$ of a proof system for $\mathcal{L}^O \vee \mathcal{L}'$ that is complete and zero-knowledge, where $\mathcal{L}'$ is some language. Then, we can construct an adversary that breaks the soundness of M in the standard soundness game. We follow the swapping technique which was introduced in Chapter 4. This result suggests that if we want to augment the capability of NIZKs in terms of the languages they prove, we should rely on certain algebraic structures.

5.1.1 Related Work

As mentioned above, many CP-NIZKs have been proposed [3, 107, 114, 115, 118]. Particularly, in [107], a commit-and-prove methodology plays an essential role in obtaining a modular composition of zk-SNARKs.

A Σ-protocol for a disjunctive language has already been proposed [7]. Therefore, we can obtain an NIZK for a disjunctive language in the random oracle model if we apply the Fiat-Shamir transformation [23] to the Σ-protocol. However, this does not affect the meaning of this work as we consider NIZKs in the CRS model.

5.2 Basic Notation

We formally define a commit-and-prove NIZK. We partially follow the definition in [118], but there are some differences. We will explain the differences after the definition.

Definition 24 (CP-NIZK) *A tuple of Turing machines* $\Pi = (\Pi.\mathsf{Crs}, \Pi.\mathsf{Com}, \Pi.\mathsf{Prv}, \Pi.\mathsf{Vrf}, \Pi.\mathsf{CrsSim}, \Pi.\mathsf{ComSim}, \Pi.\mathsf{PrvSim})$ *that work as follows is a* commit-and-prove non-interactive zero-knowledge proof system (CP-NIZK) *for a language* $\mathcal{L}$*.*

$\Pi.\mathsf{Crs}$**:** $ck \leftarrow \Pi(\mathsf{Crs}, \tau)$

Given a trapdoor τ*, output a CRS (or a commitment key)* ck*.*

$\Pi.\mathsf{Com}$**:** $c \leftarrow \Pi(\mathsf{Com}, ck, w, r)$
Given a CRS ck, a witness w and a randomness (or an opening) r, output a commitment c or $\perp$.

$\Pi.\mathsf{Prv}$**:** $\pi \leftarrow \Pi(\mathsf{Prv}, ck, x, w, r)$
Given a CRS ck, an instance x, a witness w and a randomness r, output a proof π or $\perp$.

$\Pi.\mathsf{Vrf}$**:** $b \leftarrow \Pi(\mathsf{Vrf}, ck, x, c, \pi)$
Given a CRS ck, an instance x, a commitment c and a proof π, output a bit $b \in \{0,1\}$ where 1 means accept and 0 means reject.

$\Pi.\mathsf{CrsSim}$**:** $(ck, \tau) \leftarrow \Pi(\mathsf{CrsSim}, \tau)$
Given a trapdoor τ, output a CRS ck and τ.

$\Pi.\mathsf{ComSim}$**:** $c \leftarrow \Pi(\mathsf{ComSim}, ck, \tau, r)$
Given a CRS ck, a trapdoor τ and a randomness r, output a commitment c if $ck = \Pi(\mathsf{Crs}, \tau)$, otherwise $\perp$.

$\Pi.\mathsf{PrvSim}$**:** $\pi \leftarrow \Pi(\mathsf{PrvSim}, ck, x, c, \tau)$
Given a CRS ck, an instance x, a commitment c and a trapdoor τ, output π.

Definition 25 (Security Properties of CP-NIZKs) *A CP-NIZK Π for a language $\mathcal{L}$ satisfies the following conditions.*

Completeness: *For any $n \in \mathbb{N}$, any $\sigma \leftarrow \Pi(\mathsf{Crs}, \tau)$, any $(x, w) \in R_{\mathcal{L}}$ and any $c \leftarrow \Pi(\mathsf{Crs}, ck, w, [r])$,* $\Pr[\Pi(\mathsf{Vrf}, \sigma, x, c, \Pi(\mathsf{Prv}, \sigma, x, w, r)) = 1] \geq 1 - \mathrm{negl}(n)$.

Soundness: *For any PPT adversary $\mathcal{A}$, the following holds;*

$$\Pr\left[\begin{array}{c} \sigma \leftarrow \Pi(\mathsf{Crs}, \tau) \\ (x, c, \pi) \leftarrow \mathcal{A}(\sigma) \end{array} \;:\; \Pi(\mathsf{Vrf}, \sigma, x, c, \pi) = 1 \wedge x \notin \mathcal{L}\right] \leq \mathrm{negl}(n).$$

Composable Zero-Knowledge: Π *is* composable zero-knowledge *if the following two conditions hold:*

- *For any PPT $\mathcal{A}$, the following advantage* $\mathtt{AdvKeyIND}_{\Pi,\mathcal{A},\mathcal{L}}$ *is negligible in n:*
  $|\Pr[ck \leftarrow \Pi(\mathsf{Crs}, \tau) : 1 \leftarrow \mathcal{A}(ck)] - \Pr[(ck, \tau) \leftarrow \Pi(\mathsf{CrsSim}, \tau) : 1 \leftarrow \mathcal{A}(ck)]|$.
- *For any stateful PPT $\mathcal{A}$, the following advantage* $\mathtt{AdvPrfIND}_{\Pi,\mathcal{A},\mathcal{L}}$ *is negligible in n:*

$$\left| \Pr\left[\begin{array}{c} (ck,\tau) \leftarrow \Pi(\mathsf{CrsSim},\tau) \\ (x,c,w) \leftarrow \mathcal{A}^{O_0(\cdot)}(ck,\tau) \\ \textit{if } (w,[r],c) \in Q \textit{ then } \pi \leftarrow \Pi(\mathsf{Prv},ck,x,w,r) \\ \textit{otherwise output } \perp \end{array} : \begin{array}{c} \mathcal{A}(\pi)=1 \\ \wedge (x,w) \in R_{\mathcal{L}} \end{array} \right] \right.$$

$$\left. -\Pr\left[\begin{array}{c} (ck,\tau) \leftarrow \Pi(\mathsf{CrsSim},1^n) \\ (x,c,w) \leftarrow \mathcal{A}^{O_1(\cdot)}(ck,\tau) \\ \textit{if } (w,[r],c) \in Q \textit{ then } \pi \leftarrow \Pi(\mathsf{PrvSim},ck,x,c,\tau) \\ \textit{otherwise output } \perp \end{array} : \begin{array}{c} \mathcal{A}(\pi)=1 \\ \wedge (x,w) \in R_{\mathcal{L}} \end{array} \right] \right|$$

where $O_0 = \Pi(\mathsf{Com}, ck, \cdot, \cdot)$, $O_1 = \Pi(\mathsf{ComSim}, ck, \tau, \cdot)$ and Q is a list made by the challenger as follows (note that $\mathcal{A}$ actually calls an oracle through the challenger): When $\mathcal{A}$ calls O_0 or O_1, the adversary sends a witness w to the challenger, then the challenger chooses a randomness r uniformly, obtains $c = \Pi(\mathsf{Com}, ck, w, r)$ or $c = \Pi(\mathsf{ComSim}, ck, \tau, r)$, returns c to $\mathcal{A}$ and records (w, r, c) to Q.

There are three differences between our definition and the definition in [118]. In [118], every algorithm of a CP-NIZK takes a tag as an input to identify the type of value that is given, such as a group element or a field element, while we do not require such a tag. Second, they divide a witness into pieces while we treat only a single witness, because it is sufficient for our purpose. Finally, in the composable zero-knowledge game of [118], the adversary outputs a statement and indices that correspond to the witnesses and commitments it chooses, while ours outputs a statement, a single witness and a single commitment.

In this work we treat a CP-NIZK for a hard language that is defined as following, since it is convenient for our purpose.

Definition 26 (Hard Language) *Let R be an efficiently verifiable binary relation. Let $\mathcal{L}_n = \{x \in \{0,1\}^{\mathrm{poly}(n)} \mid \exists w \in \{0,1\}^{\mathrm{poly}'(n)} \text{ s.t. } R(x,w)=1\}$ and $\mathcal{L} = \bigcup_n \mathcal{L}_n$. Let $\mathcal{C}_n \subseteq \{0,1\}^{\mathrm{poly}(n)}$ be a set and $\mathcal{C} = \bigcup_n \mathcal{C}_n$. $\mathcal{L}$ is ϵ_{ind}-hard if the followings hold:*

- *For any security parameter n, $\mathcal{L}_n \cap \mathcal{C}_n = \emptyset$.*
- *$\mathcal{L}$ and $\mathcal{C}$ are efficiently samplable. That is, for any security parameter n, there exist distributions $\mathcal{D}_{\mathcal{L}_n}$ and $\mathcal{D}_{\mathcal{C}_n}$ from which $\mathcal{L}_n$ and $\mathcal{C}_n$ are efficiently samplable respectively.*
- *For any PPT $\mathcal{A}$ and any security parameter n, it hols that*

$$\begin{aligned} \mathtt{LangIND}_{\mathcal{L}_n,\mathcal{C}_n,\mathcal{A}}(n) &= |\Pr[x \leftarrow \mathcal{D}_{\mathcal{L}_n} : 1 \leftarrow A(x)] - \Pr[x \leftarrow \mathcal{D}_{\mathcal{C}_n} : 1 \leftarrow A(x)]| \\ &\leq \epsilon_{\mathsf{ind}} \end{aligned}$$

where ϵ_{ind} is negligible.

5.3 A CP-NIZK Oracle

In this section, we introduce an oracle that implements a CP-NIZK for a hard language and demonstrate that such an oracle indeed constitutes a CP-NIZK. Before introducing the CP-NIZK oracle, we define another oracle that implements a hard language as follows:

Definition 27 *Let* $H_{\mathsf{smpl}} : \{0,1\}^{n+1} \to \{0,1\}^{2n}$ *be a random injection. An oracle* O *provides the three functionalities* SmplYes*,* SmplNo *and* Promise *as follows:*

O.SmplYes**:** $x \leftarrow O(\mathsf{SmplYes}, w)$
Given $w \in \{0,1\}^n$*, compute* $x \leftarrow H_{\mathsf{smpl}}(1||w)$ *and output* x*.*

O.SmplNo**:** $x \leftarrow O(\mathsf{SmplNo}, w)$
Given $w \in \{0,1\}^n$*, compute* $x \leftarrow H_{\mathsf{smpl}}(0||w)$ *and output* x*.*

O.Promise**:** $b \leftarrow O(\mathsf{Promise}, x)$
Given $x \in \{0,1\}^{2n}$*, output 0 if* $\perp \leftarrow H_{\mathsf{smpl}}(x)$*, otherwise 1.*

Let $\mathcal{O}_n$ be the set of all oracles that satisfy the above syntax with security parameter n, and let $\mathcal{O}$ be the collection of $\mathcal{O}_n$ for all $n > 0$.

For $O \in \mathcal{O}_n$, let $\mathsf{L}^O = (\mathsf{L}^O.\mathsf{SmplYes}, \mathsf{L}^O.\mathsf{SmplNo}, \mathsf{L}^O.\mathsf{Promise})$ be an oracle machine that works as follows:

L^O.SmplYes**:** Given $(\mathsf{SmplYes}, w)$, output $x \leftarrow O(\mathsf{SmplYes}, w)$.

L^O.SmplNo**:** Given (SmplNo, w), output $x \leftarrow O(\mathsf{SmplNo}, w)$.

L^O.Promise**:** Given $(\mathsf{Promise}, x)$, output $b \leftarrow O(\mathsf{Promise}, x)$.

It is known that L^O constitutes a hard language as shown in the following lemma:

Lemma 9 ([50]) *The algorithm* L^O *constitutes a hard language* $(\mathcal{L}_n^O, \mathcal{C}_n^O)$ *where*

$$\mathcal{L}_n^O = \{x \mid \exists w \, s.t. \, x = H_{\mathsf{smpl}}(1||w)\},$$
$$\mathcal{C}_n^O = \{x \mid \exists w \, s.t. \, x = H_{\mathsf{smpl}}(0||w)\}.$$

Now, we introduce an oracle that almost directly implements a CP-NIZK for $\mathcal{L}_n^O$. The oracle has several functionalities, and some of them are implemented by random injections, which are accessible only within the interfaces. Namely, the CRS generator and prover interfaces (i.e., Crs, Prv and PrvSim) are implemented by random injections H_{crs} and H_{prf}, respectively, where these random injections work only when valid inputs are given to the interfaces. We guarantee the soundness of a proof generated by the oracle by making the prover interfaces so that they work only when they are given a correct witness or a trapdoor of a CRS.

Definition 28 *Let $O \in \mathcal{O}_n$ be an oracle of the kind that is defined in Definition 27, and let $\mathcal{L}_n^O$ be the language defined in Lemma 9. An NIZK oracle* $\mathsf{ZK} = (\mathsf{ZK.Crs}, \mathsf{ZK.Com}, \mathsf{ZK.Prv}, \mathsf{ZK.Vrf}, \mathsf{ZK.PrvSim})$ *for $\mathcal{L}_n^O$ is equipped with random injections $H_{\mathsf{crs}} : \{0,1\}^n \to \{0,1\}^{2n}$, $H_{\mathsf{com}} : \{0,1\}^{4n} \to \{0,1\}^{5n}$ and $H_{\mathsf{prf}} : \{0,1\}^{9n} \to \{0,1\}^{10n}$ that implement the functionalities below*[1].

$\mathsf{ZK.Crs}$**:** $ck \leftarrow \mathsf{ZK}(\mathsf{Crs}, \tau)$
Given a trapdoor $\tau \in \{0,1\}^n$, output a CRS $ck \leftarrow H_{\mathsf{crs}}(\tau)$.

$\mathsf{ZK.Com}$**:** $c \leftarrow \mathsf{ZK}(\mathsf{Com}, ck, w, r)$
Given a CRS $ck \in \{0,1\}^{2n}$, a witness $w \in \{0,1\}^n$ and a randomness $r \in \{0,1\}^n$, output a commitment $c \leftarrow H_{\mathsf{com}}(ck||w||r)$ if there exists a trapdoor τ s.t. $H_{\mathsf{crs}}^{-1}(ck) = \tau$, otherwise output $\perp$.

$\mathsf{ZK.Prv}$**:** $\pi \leftarrow \mathsf{ZK}(\mathsf{Prv}, ck, x, w, r)$
Given a CRS $ck \in \{0,1\}^{2n}$, a statement $x \in \{0,1\}^{2n}$, a witness $w \in \{0,1\}^n$ and a randomness $r \in \{0,1\}^n$, if $(x,w) \in R_{\mathcal{L}_n^O}$ and there exists a trapdoor τ s.t. $H_{\mathsf{crs}}^{-1}(ck) = \tau$, then compute $c = H_{\mathsf{com}}(ck, w, r)$ and output a proof $\pi \leftarrow H_{\mathsf{prf}}(ck||x||c)$, otherwise output $\perp$.

$\mathsf{ZK.Vrf}$**:** $b \leftarrow \mathsf{ZK}(\mathsf{Vrf}, ck, x, c, \pi)$
Given a CRS $ck \in \{0,1\}^{2n}$, a statement $x \in \{0,1\}^{2n}$, a commitment $c \in \{0,1\}^{5n}$ and a proof $\pi \in \{0,1\}^{10n}$, output 1 if $\pi = H_{\mathsf{prf}}(ck||x||c)$. Otherwise output 0.

$\mathsf{ZK.PrvSim}$**:** $\pi \leftarrow \mathsf{ZK}(\mathsf{PrvSim}, ck, x, c, \tau)$
Given a CRS $ck \in \{0,1\}^{2n}$, a statement $x \in \{0,1\}^{2n}$, a commitment $c \in \{0,1\}^{5n}$ and a trapdoor $\tau \in \{0,1\}^n$, if $ck = H_{\mathsf{crs}}(\tau)$ then output a proof $\pi \leftarrow H_{\mathsf{prf}}(ck||x||c)$, otherwise output $\perp$.

Let $\mathcal{Z}_n$ be the set of all oracles that satisfy the above syntax with security parameter n, and let $\mathcal{Z}$ be the collection of $\mathcal{Z}_n$ for all $n > 0$. The reader may wonder that the above oracle lacks interfaces that implement CrsSim and ComSim. However, we can construct such functionalities from $\mathsf{ZK.Crs}$ and $\mathsf{ZK.Com}$, respectively.

The Construction of a CP-NIZK

Let $O \in \mathcal{O}_n$ be an oracle of the type defined in Definition 27, $\mathcal{L}_n^O$ be the language defined in Lemma 9 and $\mathsf{ZK} \in \mathcal{Z}_n$ be an oracle of the type defined in Definition 28. We construct M for a CP-NIZK for $\mathcal{L}_n^O$ based on O and ZK as follows:

[1] Since these functions are injections, their inverse functions are defined uniquely.

$\mathsf{M.Crs}$**:** $ck \leftarrow \mathsf{M}(\mathsf{Crs}, \tau)$
Given a trapdoor τ, output $ck \leftarrow \mathsf{ZK}(\mathsf{Crs}, \tau)$.

$\mathsf{M.Com}$**:** $c \leftarrow \mathsf{M}(\mathsf{Com}, ck, w, r)$
Given a CRS ck, a witness w and a randomness r, output a commitment $c \leftarrow \mathsf{ZK}(\mathsf{Com}, ck, w, r)$.

$\mathsf{M.Prv}$**:** $\pi \leftarrow \mathsf{M}(\mathsf{Prv}, ck, x, w, r)$
Given a CRS ck, a statement x, a witness w and a randomness r, output a proof $\pi \leftarrow \mathsf{ZK}(\mathsf{Prv}, ck, x, w, r)$.

$\mathsf{M.Vrf}$: $b \leftarrow \mathsf{M}(\mathsf{Vrf}, ck, x, c, \pi)$
Given a CRS ck, a statement x, a commitment c and a proof π, output a verification result $b \leftarrow \mathsf{ZK}(\mathsf{Vrf}, ck, x, c, \pi)$.

$\mathsf{M.CrsSim}$**:** $(ck, \tau) \leftarrow \mathsf{M}(\mathsf{CrsSim}, \tau)$
Given a trapdoor τ, output τ and a CRS $ck \leftarrow \mathsf{ZK}(\mathsf{Crs}, \tau)$.

$\mathsf{M.ComSim}$**:** $c \leftarrow \mathsf{M}(\mathsf{ComSim}, ck, \tau, r)$
Given a CRS ck, a trapdoor τ and a randomness r, if $ck \leftarrow \mathsf{ZK}(\mathsf{Crs}, \tau)$, then output a commitment $c = \mathsf{ZK}(\mathsf{Com}, ck, \tau, r)$.

$\mathsf{M.PrvSim}$**:** $\pi \leftarrow \mathsf{M}(\mathsf{PrvSim}, ck, x, c, \tau)$
Given a CRS ck, a statement x, a commitment c and a trapdoor τ, output a proof $\pi \leftarrow \mathsf{ZK}(\mathsf{PrvSim}, ck, x, c, \tau)$.

Lemma 10 M *is a CP-NIZK for* $\mathcal{L}_n^O$.

Proof of Lemma 10. Let $\mathcal{A}$ be a PPT adversary. Without loss of generality, we assume that $\mathcal{A}$ makes at most $q = \mathrm{poly}(n)$ queries. Completeness is immediate. We show that M is sound. Suppose that $\mathcal{A}$ is given a legitimate CRS ck and outputs $x \notin \mathcal{L}_n^O$, c and π. Since $x \notin \mathcal{L}_n^O$, $\mathcal{A}$ should evoke $\mathsf{M.Prv}$ to generate π for x that passes the verification by $\mathsf{M.Vrf}$. There are two possibilities in which $\mathcal{A}$ can create such a proof: Call $\mathsf{ZK.PrvSim}$ on a trapdoor τ of ck, c and x to obtain a simulated proof, or compute a proof π so that it passes the verification without a query that results in π.

Regarding the first case, as $\mathsf{ZK.Crs}$ is implemented by a random injection, $\mathcal{A}$ should make a query to $\mathsf{ZK.Crs}$ on a trapdoor and see if the given CRS is returned to find τ. Since there are 2^n candidates for τ, the probability that $\mathcal{A}$ makes a query on τ is $1/2^n$. Taking the union bound for at most q queries, $\mathcal{A}$ makes such a query with probability at most $q/2^n$. We evaluate the second case. We remark that, in this case, $\mathcal{A}$ computes a legitimate π without making a query to $\mathsf{ZK.PrvSim}$, since otherwise it implies the first case. Considering the domain of $H_{\mathsf{prf}} : \{0,1\}^{9n} \to \{0,1\}^{10n}$, the

probability that such a proof is legitimate is at most $q \cdot 2^{9n}/2^{10n} = q/2^n$, which is negligible.

Now, we show that M is composable zero-knowledge. As a first step, we prove that $\mathtt{AdvKeyIND}_{\mathsf{M},\mathcal{A},\mathcal{L}_n^O}$ is negligible in n. Since $\mathsf{M.Crs}$ and $\mathsf{M.CrsSim}$ are implemented by the same random injection, $\mathcal{A}$ cannot distinguish the algorithm from which the CRS comes. Thus, we have $\mathtt{AdvKeyIND}_{\mathsf{M},\mathcal{A},\mathcal{L}_n^O}(n) = 0$.

We demonstrate that, by a hybrid argument, for any PPT $\mathcal{A}$, $\mathtt{AdvPrfIND}_{\mathsf{M},\mathcal{A},\mathcal{L}_n^O}$ is negligible. Let O_0 and O_1 be the oracles that are explicitly given in Definition 24 (note that in the composable zero-knowledge game, O and ZK are given in addition to O_0 or O_1). We say $\mathcal{A}$ is in the "real" (resp., "simulated") world if $\mathcal{A}$ is given O_0 (resp., O_1). We introduce three games where the first game corresponds to the real world and the third game corresponds to the simulated world. Let P_i be the probability that $\mathcal{A}$ chooses a pair $(x, w) \in R_{\mathcal{L}_n^O}$ and finally outputs 1 in $\mathsf{Game}\ i$.

$\mathsf{Game}\ 0$: A composable zero-knowledge game where the adversary is given O_0 and the challenger runs $\mathsf{M.Prv}$ to obtain a proof π. We describe the composable zero-knowledge game in the real world as follows:

Step 1 The challenger uniformly chooses $\tau \leftarrow \{0,1\}^n$, runs $(ck, \tau) \leftarrow \mathsf{M}(\mathsf{CrsSim}, \tau)$ and sends (ck, τ) to the adversary.

Step 2 Given (ck, τ), the adversary outputs (x, w, c), where $(x, w) \in R_{\mathcal{L}^O}$ and c is obtained as follows:

- The adversary sends w to the challenger along with calling O_0.
- Given w, the challenger chooses a randomness $r \in \{0,1\}^n$ uniformly, obtains $c = \mathsf{M}(\mathsf{Com}, ck, x, w, r)$, sends c to the challenger and adds (w, r, c) to Q where Q is an initially empty list.

Note that the adversary obtains at most q commitments.

Step 3 Given (x, w, c), the challenger determines whether there exists an entry (w, r, c) in Q. If such an entry exists, then the challenger computes $\pi \leftarrow \mathsf{M}(\mathsf{Prv}, ck, x, w, r)$ and sends π to the adversary, otherwise outputs $\perp$.

Step 4 Given π, the adversary outputs 0 if $\mathcal{A}$ decides that π is generated by the real prover, otherwise 1.

$\mathsf{Game}\ 1$: Modify $\mathsf{Game}\ 0$ so that the challenger runs $\mathsf{M}(\mathsf{PrvSim}, ck, x, c, \tau)$ to obtain π in Step 3.

We remark that $\mathsf{M.PrvSim}$ gives an output other than $\perp$ on the input (ck, x, c, τ). Observe that $\mathsf{M}(\mathsf{PrvSim}, ck, x, c, \tau) = \mathsf{M}(\mathsf{Prv}, ck, x, w, r) = \pi = H_{\mathsf{prf}}(ck||x||c)$. Hence, this modification does not change the distribution of the composable zero-knowledge game and we have $P_1 = P_0$.

Game 2: Replace O_0 in Game 1 with O_1. Note that this game corresponds to the simulated world.

Recall that O_0 and O_1 are actually $\mathsf{M.Com}$ and $\mathsf{M.ComSim}$, respectively, and they are implemented by the same random injection H_{com}. Therefore, the distribution of the output of this game differs from that of Game 1 only if $\mathcal{A}$ obtains a commitment that is generated by the challenger herself, i.e., only if $\mathcal{A}$ makes a query that includes the randomness r, which is chosen uniformly by the challenger. That is, as $\mathcal{A}$ knows the trapdoor τ and the witness w, $\mathcal{A}$ can obtain c by making queries $\mathsf{ZK}(\mathsf{Com}, ck, w, r)$ and $\mathsf{ZK}(\mathsf{ComSim}, ck, \tau, r)$ if $\mathcal{A}$ obtains r.

To analyze the probability that $\mathcal{A}$ makes a query that includes r, we should consider two cases. First, $\mathcal{A}$ makes such a query before obtaining c. As $r \in \{0,1\}^n$ is chosen uniformly, the probability that a query made by $\mathcal{A}$ contains r is $1/2^n$. Considering the assumption that $\mathcal{A}$ makes at most q queries during the composable zero-knowledge game, the probability that this event occurs is at most $q/2^n$. The second case is that $\mathcal{A}$ makes such a query after obtaining c. That is, $\mathcal{A}$ might gain some information about r from c. However, c is generated by the random injection H_{com}. Hence, we can apply the same discussion as the first case and conclude that this event happens with probability at most $q/2^n$. Summarizing the above, we have $|P_2 - P_1| \leq 2q/2^n$. Now, the difference between Game 0 and Game 2 corresponds to the advantage in the composable zero-knowledge game. Summarizing the above, we have $\mathtt{AdvPrfIND}_{\mathsf{M},\mathcal{A},\mathcal{L}_n^O}(n) \leq 2q/2^n$, which is negligible. □

Remark 1 *Let* $\mathsf{ZK}, \mathsf{ZK}' \in \mathcal{ZK}_n$ *be oracles that constitute CP-NIZKs for some languages* $\mathcal{L}$ *and* $\mathcal{L}'$ *respectively. If* $\mathsf{ZK.Com}$ *and* $\mathsf{ZK}'\mathsf{.Com}$ *are implemented by the same random injection, then we can construct an NIZK that proves witness equality (i.e., an NIZK for* $(x,w) \in R_{\mathcal{L}} \wedge (x',w') \in R_{\mathcal{L}'} \wedge w = w'$*).*

5.4 Separation

This section presents our main result. That is, we show the following theorem:

Theorem 11 *Given a hard language* $\mathcal{L}$ *and a CP-NIZK for* $\mathcal{L}$ *respectively, there is no fully black-box construction of a (standard) NIZK for* $\mathcal{L} \vee \hat{\mathcal{L}}$ *where* $\hat{\mathcal{L}}$ *is a hard language.*

As we would like to show fully black-box separation, it suffices to show the absence of a black-box construction of an NIZK for a specific $\mathcal{L} \vee \hat{\mathcal{L}}$. Let O be an oracle of the type defined in Definition 27, $\mathcal{L}_n^O$ be the language defined in Lemma 9 and ZK be an oracle of the type defined in Definition 28. Thus, we assume that there exists a black-box construction M of a proof system for $\mathcal{L}_n^O \vee \hat{\mathcal{L}}$, where $\hat{\mathcal{L}}$ is a hard language along with $\hat{\mathcal{C}}$, which is complete and zero-knowledge, and we present an adversary

that attacks the soundness of M. As M is complete, for any $n \in \mathbb{N}$, any $\tilde{\sigma} \leftarrow \mathsf{M}(\mathsf{Crs}, \tilde{\tau})$ and any $(\tilde{x}, \tilde{w}) \in R_{\mathcal{L}_n^O \vee \hat{\mathcal{L}}}$, we have $\Pr[\mathsf{M}(\mathsf{Vrf}, \tilde{\sigma}, \tilde{x}, \mathsf{M}(\mathsf{Prv}, \tilde{\sigma}, \tilde{x}, \tilde{w})) = 1] \geq 1 - \rho_{\mathsf{co}}$ where ρ_{co} is negligible. Similarly, it holds that $\mathtt{AdvZK}_{\mathcal{A},\mathsf{M},\mathcal{L}^O \vee \hat{\mathcal{L}}} \leq \rho_{\mathsf{zk}}$ for any PPT $\mathcal{A}$ where ρ_{zk} is negligible.

We implicitly assume that a CRS (resp., a proof) generated by $\mathsf{M.Crs}$ (resp., $\mathsf{M.Prv}$) contains CRSs (resp., proofs) generated by $\mathsf{ZK.Crs}$ (resp., $\mathsf{ZK.Prv}$), since otherwise, it implies that we can construct M without the oracle ZK. Furthermore, we assume that a proof $\tilde{\pi}$ generated by $\mathsf{M.Prv}$ contains a commitment that is necessary to verify a proof embedded in $\tilde{\pi}$. Without loss of generality, we assume that every algorithm in this section makes at most $q = \mathrm{poly}(n)$ queries. Thus, at most q values are embedded in every value output by M. As $\mathcal{L}_n^O$ and $\hat{\mathcal{L}}$ are hard languages, for any PPT $\mathcal{A}$, it holds that $\mathtt{LangIND}_{\mathcal{L}_n^O,\mathcal{C}_n^O,\mathcal{A}}(n) \leq \rho_{\mathsf{ind}}$ and $\mathtt{LangIND}_{\hat{\mathcal{L}}_n,\hat{\mathcal{C}}_n,\mathcal{A}}(n) \leq \hat{\rho}_{\mathsf{ind}}$ where ρ_{ind} and $\hat{\rho}_{\mathsf{ind}}$ are negligible in n respectively.

Observation on a CRS. As observed in [50], even if a CRS $\tilde{\sigma}$ generated by $\mathsf{M.Crs}$ contains some legitimate CRSs with respect to $\mathsf{ZK.Crs}$, these CRSs cannot be used to generate a proof by $\mathsf{M.Prv}$ unless its trapdoor is known to $\mathsf{M.Prv}$. Since $\mathsf{M.Prv}$ proves a disjunctive language, there are cases in which only one side of a statement is true (i.e., an (yes, no)-instance or a (no, yes)-instance). However, a proof generated by $\mathsf{M.Prv}$ should not leak which side of the statement is the yes-instance, as M is zero-knowledge. Assume that $\tilde{\sigma}$ contains a legitimate CRS with respect to $\mathsf{ZK.Crs}$ without its trapdoor (we say such a CRS is *non-trivial*), and that $\mathsf{M.Prv}$ is given a (no, yes)-instance. Then, the prover algorithm cannot prove the no-instance on the non-trivial CRS, as its trapdoor is required to prove the no-instance. If $\mathsf{M.Prv}$ is supposed to use only this type of CRS, then a proof generated by $\mathsf{M.Prv}$ may leak the fact that the instance is a (no, yes)-instance. Hence, to generate a zero-knowledge proof, $\mathsf{M.Prv}$ should use a CRS with respect to $\mathsf{ZK.Crs}$ whose trapdoor is embedded in $\tilde{\sigma}$ (we say such a CRS is *trivial*) or a CRS that is generated by the prover algorithm itself.

Overview of The Adversary. Before formally demonstrating an adversary $\mathcal{A}$, we sketch the adversary in the standard soundness game. First, given a CRS $\tilde{\sigma}$, the adversary learns trivial CRSs and their trapdoors with respect to $\mathsf{ZK.Crs}$ by following the above observation. Since we do not know how these pairs are encoded in $\tilde{\sigma}$, we let $\mathcal{A}$ run $\mathsf{M.Prv}$ and $\mathsf{M.Vrf}$ sufficiently many times on $\tilde{\sigma}$, an (yes, no)-instance and its witness where they are chosen uniformly. After this step, $\mathcal{A}$ samples a (no, no)-instance $(x, \hat{x})$ uniformly and runs $\mathsf{M.Prv}$ on the instance to forge a proof. Similar to Chapter 4, during the execution of $\mathsf{M.Prv}$, we apply the swapping technique [50]. Recall that a query made by $\mathsf{M.Prv}$ is actually relayed by the adversary. Thus, when $\mathsf{M.Prv}$ makes a query to $\mathsf{ZK.Prv}$ on x and a CRS ck whose trapdoor τ is known to $\mathcal{A}$ (i.e., a trivial CRS or a CRS generated during the execution of $\mathsf{M.Prv}$), the adversary obtains a proof by making a query to $\mathsf{ZK.PrvSim}$ on x, ck and τ and returns the answer to $\mathsf{M.Prv}$. Note that a query to $\mathsf{ZK.Prv}$ on x and ck should result in $\perp$ since x

is a no-instance. However, there might be a case in which $\mathsf{M.Prv}$ makes a query on a CRS whose trapdoor is not known (i.e., a non-trivial CRS or a CRS that is found accidentally without prior generation by $\mathsf{ZK.Crs}$). In such a case, the adversary assigns a randomly chosen proof as the answer to the query. (Clearly, if such a proof is verified by the challenger, it results in 0 with high probability, and the attack might fail. However, we will show later that such a random proof is verified by $\mathsf{M.Vrf}$ with only low probability.) After $\mathsf{M.Prv}$ outputs a (forged) proof, the adversary passes it to the challenger. Then, the challenger verifies the proof by $\mathsf{M.Vrf}$, and it should pass the verification, as it was generated by $\mathsf{M.Prv}$.

Soundness Game and The Adversary

We describe the adversary in a standard soundness game as follows:

Step 1: The challenger chooses $\tilde{\tau}$ uniformly, computes a CRS $\tilde{\sigma} \leftarrow \mathsf{M}(\mathsf{Crs}, \tilde{\tau})$ and sends it to the adversary. Let Q_{leg} be a list of CRSs and their trapdoors s.t. $ck \leftarrow O(\mathsf{Crs}, \tau)$ appears during this step.

Step 2: Given the CRS $\tilde{\sigma}$, the adversary $\mathcal{A}$ repeats the following q^c times where c is some constant: Sample $(x_i, w_i) \in R_{\mathcal{L}_n^O}$ and $(\hat{x}_i, \hat{w}_i) \in R_{\hat{\mathcal{C}}}$ uniformly, and obtain $\pi_i \leftarrow \mathsf{M}(\mathsf{Prv}, \tilde{\sigma}, x_i, \hat{x}_i, w_i, \hat{w}_i)$ and $b = \mathsf{M}(\mathsf{Vrf}, \tilde{\sigma}, x_i, \hat{x}_i, \pi)$.

Let Q_{triv} be a set of (ck, τ) pairs s.t. a query $ck = \mathsf{ZK}(\mathsf{Crs}, \tau)$ or $\mathsf{ZK}(\mathsf{PrvSim}, ck, \cdot, \cdot, \tau, \cdot) = \pi \neq \bot$ appears during this step. Roughly speaking, Q_{triv} is a set of pairs of a trapdoor and a CRS s.t. the adversary generates them in this phase or the pair is encoded in $\tilde{\sigma}$

Step 3: $\mathcal{A}$ chooses $w \in \{0,1\}^n$ and $\hat{x} \in \hat{\mathcal{C}}$ uniformly and obtains $x = O(\mathsf{SmplNo}, w)$ and $x^* = O(\mathsf{SmplYes}, w)$. The adversary defines new partial oracles O' and ZK' based on the query answer pairs that she has learned. That is, the adversary applies the swapping technique to x and x^* in O and ZK, respectively.

Partial Oracle O'

A new oracle O' is obtained by swapping x and x^* in O. That is, O' consists of entries $O'(\mathsf{SmplYes}, w; x)$ and $O'(\mathsf{SmplNo}, w; x^*)$.

Partial Oracle ZK'

The new oracle ZK' contains the entries $\mathsf{ZK}'(\mathsf{Prv}, \cdot, x, w, \cdot; \bot)$.

Let $\tilde{x} = (x^*, \hat{x})$ and $\tilde{w} = (w, \bot)$. The adversary evokes $\mathsf{M}^{O'', \mathsf{ZK}''}(\mathsf{Prv}, \tilde{\sigma}, \tilde{x}, \tilde{w})$ where O'' and ZK'' are algorithms defined as follows:

[Algorithm O'']

Algorithm O'' works as follows:

- If O'' is given a query registered in O', then it returns the registered answer.
- Otherwise, it forwards the query to O and returns the answer.

[Algorithm ZK'']

Let Q_{intl} be an initially empty set. Algorithm ZK'' works as follows (recall that the

bracket notation $[x]$ means a value that matches any value, and we denote the value by x thereafter):

- For any query registered in ZK', return the registered answer.
- $\mathsf{ZK}''.\mathsf{Crs}$: For any query of the form $(\mathsf{Crs}, [\tau])$, return $ck \leftarrow \mathsf{ZK}(\mathsf{Crs}, \tau)$ and record (ck, τ) in Q_{intl}.
- $\mathsf{ZK}''.\mathsf{Prv}$: For any query of the form $(\mathsf{Prv}, [ck], x^*, w, [r])$ with a legitimate CRS ck, obtain $c = \mathsf{ZK}(\mathsf{Com}, ck, w, r)$ and do the following:
  - If there exists an entry $(ck, [\tau]) \in Q_{\mathsf{triv}} \cup Q_{\mathsf{intl}}$, then return $\pi = \mathsf{ZK}(\mathsf{PrvSim}, ck, x^*, c, \tau)$ and record $(\mathsf{Prv}, ck, x^*, w, r; \pi)$ to ZK'.
  - If there is no entry s.t. $(ck, [\tau]) \in Q_{\mathsf{triv}} \cup Q_{\mathsf{intl}}$, then choose $\pi \in \{0,1\}^{10n}$ uniformly, return π and record $(\mathsf{Prv}, ck, x^*, w, r; \pi)$ and $(\mathsf{Vrf}, ck, x^*, c, \pi; 1)$ to ZK'.
- For every other query, forward it to ZK and return the answer.

If $\mathsf{M}.\mathsf{Prv}$ outputs a proof $\tilde{\pi}$, send $\tilde{x}$ and $\tilde{\pi}$ to the challenger.

Step 4: Given $\tilde{x}$ and $\tilde{\pi}$, the challenger outputs $\mathsf{M}(\mathsf{Vrf}, \tilde{\sigma}, \tilde{x}, \tilde{\pi})$. If $\mathsf{M}(\mathsf{Vrf}, \tilde{\sigma}, \tilde{x}, \tilde{\pi}) = 1$, then the adversary wins.

Evaluation

We first introduce the following lemmas, which are useful for the analysis of the adversary.

Lemma 11 ([120]) *Let $X_1, \cdots, X_{n+1}$ be independent Bernoulli random variables. Let $\Pr[X_i = 1] = p$ and $\Pr[X_i = 0] = 1 - p$ for some $p \in [0,1]$. Let E be an event in which the first n variables are sampled at 1 and X_{n+1} is sampled at 0. Then, $\Pr[E] \leq 1/(e \cdot n)$ where e is the base of the natural logarithm.*

Lemma 12 ([121]) *Let A, B, and F be events defined in some probability space, and suppose that $A \wedge \neg F \Leftrightarrow B \wedge \neg F$. Then, $|\Pr[A] - \Pr[B]| \leq \Pr[F]$.*

Our adversary follows the swapping technique similar to Chapter 4, but the analysis is slightly different. That is, in Chapter 4, we first ruled out several bad events, and then evaluated the probability that the adversary wins the game, under the assumption that these bad event never happens. In other words, we directly reduced the probability space so that the bad events do not occur. In this chapter, we follow Lemma 12, aiming to simpler analysis as we are not necessary to shrink probability space.

We show that the challenger accepts the proof with noticeable probability by a hybrid argument. We start with the soundness game, and ultimately reach a situation

where the challenger accepts the proof trivially. The first game, Game 0, is the soundness game itself. In the next four games, from Game 1 to Game 4, we exclude certain bad events that happen only by accident. Note that in these games, we sometimes let the soundness game abort (or halt) if these bad events happen. This means that the challenger outputs 0 and the adversary loses the soundness game. Then, we modify the game step by step, finally reaching a situation where the challenger trivially accepts the proof unless a completeness error occurs. Let P be the probability that the challenger accepts the proof in the soundness game and P_i be the same probability in Game i.

Game 0: This is the soundness game; thus, $P_0 = P$.

Game 1: This game excludes the case where a legitimate value suddenly appears without its prior generation by O or ZK. That is, we halt the game if one of the following events occurs:

- A successful query that includes a legitimate CRS ck is made but there is no entry $(ck, [\tau])$ in $Q_{\mathsf{leg}} \cup Q_{\mathsf{triv}} \cup Q_{\mathsf{intl}}$.
- A successful query that includes
  - $x \in \mathcal{L}_n^O$ or $x \in \mathcal{C}_n^O$
  - a legitimate commitment c
  - a legitimate proof π

  is made without their prior generation by the given oracles.

Regarding the first case, the probability that a CRS without prior generation by $\mathsf{ZK.Crs}$ is legitimate is bounded by $1/2^n$, as H_{crs} is a random injection s.t. $H_{\mathsf{crs}} : \{0,1\}^n \to \{0,1\}^{2n}$. Therefore, the probability that this query occurs can be evaluated if we take the union bound for the number of queries that are made during the soundness game. Recall that M makes at most q queries in its execution. Thus, the number of queries made in Steps 1, 3 and 4 are at most q respectively, since in these steps, only a single subroutine of M is executed once (we ignore the instance sampling queries by $\mathcal{A}$ in Step 3 because they are obviously irrelevant to the query we are concerned with). In Step 2, $\mathsf{M.Prv}$ and $\mathsf{M.Vrf}$ are executed q^c times. Thus, at most $2q \cdot q^c$ queries are made in this step (the instance sampling queries are ignored here as well). Since at most $3q + 2q^{c+1}$ queries are made in the game, the probability that the first case deviates is at most $(3q + 2q^{c+1})/2^n$. The other cases can be evaluated in the same manner by considering the domains of $H_{\mathsf{smpl}} : \{0,1\}^{n+1} \to \{0,1\}^{2n}$, $H_{\mathsf{com}} : \{0,1\}^{4n} \to \{0,1\}^{5n}$ and $H_{\mathsf{prf}} : \{0,1\}^{9n} \to \{0,1\}^{10n}$. Thus, we have

$$|P_1 - P_0| \leq 5(3q + 2q^{c+1})/2^n.$$

Game 2: We halt the game if a query including w is made in Step 1 or 2. As w is uniformly chosen by $\mathcal{A}$ and $\mathsf{M.Crs}$ makes at most q queries in Step 1, the probability that a query on w is made in Step 1 is at most $q/2^n$. Recall that the adversary chooses q^c witnesses and runs $\mathsf{M.Prv}$ and $\mathsf{M.Vrf}$ q^c times in Step 2. As M makes at most q queries, there are at most $q^c + 2q \cdot q^c$ chances to make a query on w in Step 2. Thus, we have

$$|P_2 - P_1| \leq (q + q^c + 2q^{c+1})/2^n,$$

which is negligible.

Game 3: We halt the game if the challenger observes $b = 0$ in Step 2. This excludes the case in which the challenger learns nothing in this step. Note that the challenger observes $b = 0$ only when a completeness error occurs, as the challenger chooses values honestly in this step. As this step is executed q^c times, we have

$$|P_3 - P_2| \leq q^c \cdot \rho_{\mathsf{co}}.$$

Game 4: We abort the game if a randomly assigned proof π by ZK'' in Step 3 appears as a result of a query to $\mathsf{ZK.Prv}$ or $\mathsf{ZK.PrvSim}$ by the end of Step 3. Similar to Game 1, there are at most $2q + 2q^{c+1}$ queries that may occur in this event (note that here we do not consider the case where this event occurs in Step 4). As $\mathsf{ZK.Prv}$ and $\mathsf{ZK.PrvSim}$ are implemented by the random injection H_{prf}, the probability that such a π is returned by $\mathsf{ZK.Prv}$ or $\mathsf{ZK.PrvSim}$ is at most $(2q+2q^{c+1})/2^n$. Furthermore, as there are at most q randomly assigned proofs, we have

$$|P_4 - P_3| \leq q(2q + 2q^{c+1})/2^n$$

Game 5: This game excludes the case where the adversary fails to learn all the trivial CRSs embedded in $\tilde{\sigma}$ in Step 2 and its trapdoor appears suddenly in Step 3. That is, the game halts if $\mathsf{M.Prv}$ in Step 3 makes a query $\mathsf{ZK}(\mathsf{PrvSim}, [ck], x, [c], [\tau])$ that results in a proof $\pi \neq \perp$ while $(ck, \tau) \notin Q_{\mathsf{triv}} \cup Q_{\mathsf{intl}}$.

As the query results in a value other than $\perp$, it implies that τ is the trapdoor of ck. Furthermore, we exclude the case where a legitimate ck appears without its generation by ZK in Game 1. Therefore, this is a case where a pair $(ck, \tau) \in Q_{\mathsf{leg}}$ does not appear in Step 2 but appears in Step 3. For each such pair, the probability that it does not appear in Step 2 but appears in Step 3 for the first time is bounded by $1/(eq^c)$ due to Lemma 11. As $\tilde{\sigma}$ contains at most q such pairs, we have

$$|P_5 - P_4| \leq 1/(eq^{c-1}).$$

Game 6: Replace O and ZK in Steps 1 and 2 with O'' and ZK'', which contain the partial oracles O' and ZK' defined at the end of Step 3, respectively. Observe that

the randomness chosen by the adversary is independent of the oracles and the randomness chosen by the challenger. Hence, modifying the game so that $\mathcal{A}$ chooses its randomness at the beginning of the soundness game does not affect the distribution of the soundness game. The view changes only if a query that includes x, x^* or w is made in Step 1 or Step 2, and we have already excluded such cases. Thus, we have $P_6 = P_5$.

Game 7: Replace O and ZK in Step 4 with O'' and ZK$''$, respectively. The view of the game changes if M.Vrf makes one of the following queries in Step 4:

- A query $(\mathsf{PrvSim}, [ck], x, [c], [\tau])$ that results in $\pi \neq \perp$ while $(ck, \tau) \notin Q_{\mathsf{triv}} \cup Q_{\mathsf{intl}}$ and there already exists an entry $(\mathsf{Prv}, ck, x, w, r; \pi' \neq \perp)$ in ZK$'$ s.t. $c = O(\mathsf{Com}, ck, w, r)$ and $\pi \neq \pi'$.
- A query that is registered in O' or ZK$'$.

Let us elaborate the first query. The proof π' is randomly assigned by the adversary in Step 3 to $(ck, \tau) \notin Q_{\mathsf{triv}} \cup Q_{\mathsf{intl}}$, but the entry $(\mathsf{Prv}, ck, x, w, r; \pi')$ is in ZK$'$. Recall that we have already excluded the case where a legitimate CRS suddenly appears without prior generation by O.Crs in Game 1. Hence, if M.Vrf makes such a query, it means that the adversary failed to learn a pair of a CRS and its trapdoor in Step 2 and such a pair appears in Step 4. Applying the same discussion as in Game 5, we obtain that such a query is made with probability at most $1/(eq^{c-1})$.

Observe that the second query is classified into two cases:

- A query that contains w, i.e., $(\mathsf{SmplYes}, w)$, (SmplNo, w) and $(\mathsf{Prv}, [ck], x, w, [r])$
- A query to ZK.Vrf that verifies a randomly assigned proof by ZK$''$ in Step 3.

Intuitively, the zero-knowledgeness of M is compromised if the first query is made. Regarding the second query, we follow the observation that a non-trivial CRS cannot be used to generate a proof $\tilde{\pi}$.

We define two events regarding these queries and show that they are made with small probability. Let AskW be an event in which M.Vrf makes a query on w in Step 4, and let VerRand be an event in which M.Vrf in Step 4 makes a query to ZK.Vrf that includes a randomly assigned proof generated by ZK$''$. By AskW^i (resp., VerRand^i), we denote an event in which AskW (resp., VerRand) occurs in game i. We claim the following two statements:

Claim 4 $\Pr[\mathsf{AskW}^7] \leq \rho_{\mathsf{zk}} + q/2^n$.

Claim 5 $\Pr[\mathsf{VerPi}^7] \leq 1/(eq^{c-1}) + q/2^n + 2\rho_{zk} + \rho_{ind} + \hat{\rho}_{ind}$.

The proofs of these claims appear after the proof of Theorem 11. Therefore, we obtain

$$\begin{aligned}|P_7 - P_6| &\leq 1/(eq^{c-1}) + \Pr[\mathsf{AskW}] + \Pr[\mathsf{VerPi}] \\ &\leq 1/(eq^{c-1}) + 2q/2^n + 3\rho_{\mathsf{zk}} + \rho_{\mathsf{ind}} + \hat{\rho}_{\mathsf{ind}}.\end{aligned}$$

Game 8: Let R_{o} and R_{zk} be uniformly chosen partial oracles such that $O'||\mathsf{R}_{\mathsf{o}} \in \mathcal{O}_n$ and $\mathsf{ZK}'||\mathsf{R}_{\mathsf{zk}} \in \mathcal{Z}_n$. Replace O'' and ZK'' with $O'||\mathsf{R}_{\mathsf{o}}$ and $\mathsf{ZK}'||\mathsf{R}_{\mathsf{zk}}$ respectively. Such oracles must exist since both O and ZK are implemented by random injections.

Recall that in Game 7, oracles $O'' = O'||O$ and $\mathsf{ZK}'' = \mathsf{ZK}'||\mathsf{ZK}$ are given. Furthermore, we have already excluded the case where M.Vrf in Step 4 makes queries that are inconsistent with O' and ZK' in Game 7. Therefore, replacing O and ZK with R_{o} and R_{zk} does not change the view in Step 4. Thus, we have $P_8 = P_7$.

Observe that now a proof generated by M.Prv is a correct proof on $(x, \hat{x})$. Therefore, M.Vrf should accept such a proof unless a completeness error occurs. Hence,

$$P_8 \geq 1 - \rho_{\mathsf{co}}.$$

Summarizing the above evaluations, we have

$$\begin{aligned}P \geq & 1 - (17q + 2q^2 + q^c + 12q^{c+1} + 2q^{c+2})/2^n \\ & - 2/(eq^{c-1}) - (1+q^c)\rho_{\mathsf{co}} - 3\rho_{\mathsf{zk}} - \rho_{\mathsf{ind}} - \hat{\rho}_{\mathsf{ind}},\end{aligned}$$

which concludes Theorem 11. □

Proof of Claim 4. We evaluate AskW by introducing subgames that ultimately reach a situation where a proof generated by the prover algorithm becomes independent of the witness. Let Game 7.0 be Game 7 and Game 7.1 be the following:

Game 7.1: Let R_{o} and R_{zk} be uniformly chosen partial oracles such that $O'||\mathsf{R}_{\mathsf{o}} \in \mathcal{O}_n$ and $\mathsf{ZK}'||\mathsf{R}_{\mathsf{zk}} \in \mathcal{Z}_n$. Replace O'' and ZK'' with $O'||\mathsf{R}_{\mathsf{o}}$ and $\mathsf{ZK}'||\mathsf{R}_{\mathsf{zk}}$, respectively. Note that such oracles must exist since both O and ZK are implemented by random injections.

Observe that the distribution in Game 7.1 differs from that of Game 7.0 only when a query in O' or ZK' is made. Thus, we obtain that $\Pr[\mathsf{AskW}^{7.1}] = \Pr[\mathsf{AskW}^{7.0}]$.

Game 7.2: Replace $O'||\mathsf{R}_{\mathsf{o}}$ and $\mathsf{ZK}'||\mathsf{R}_{\mathsf{zk}}$ with $O \leftarrow \mathcal{O}_n$ and $\mathsf{ZK} \leftarrow \mathcal{Z}_n$. Furthermore, let the adversary run M.Prv on a correct instance, i.e., on $(x^*, \hat{x})$.

Such modifications do not yield a difference between Game 7.1 and Game 7.2. Note that in Game 7.1, the partial oracles O' and ZK' are determined by the choice of oracles and randomnesses in the challenger and the adversary, and R_{o} and R_{zk} are chosen uniformly. Thus, modifying the choice of oracles so that they are chosen uniformly does not change the view of the game. Hence, we have $\Pr[\mathsf{AskW}^{7.2}] = \Pr[\mathsf{AskW}^{7.1}]$.

Game 7.3: Replace M.Crs in Step 1 and M.Prv in Step 3 with M.CrsSim and M.PrvSim, respectively. Furthermore, let the challenger pass the trapdoor generated by M.CrsSim to the adversary. We claim the following.

Claim 6 $\Pr[\mathsf{AskW}^{7.2}] - \Pr[\mathsf{AskW}^{7.3}] \leq \rho_{\mathsf{zk}}$.

Proof of Claim 6. We construct a stateful PPT adversary $\mathcal{B} = (\mathcal{B}_0, \mathcal{B}_1)$ that attacks the zero-knowledgeness of M, assuming that $\Pr[\mathsf{AskW}^{7.2}] - \Pr[\mathsf{AskW}^{7.3}] > \rho_{\mathsf{zk}}$. Such an adversary contradicts the assumption that M is zero-knowledge, thus justifying the claim. The adversary works in the (standard) zero-knowledge game with a challenger as follows:

$\mathcal{B}_0$: Given a CRS $\tilde{\sigma}$, sample $w \leftarrow \{0,1\}^n$ and $\hat{x} \in \hat{\mathcal{C}}$ and obtain $x^* = O(\mathsf{SmplYes}, w)$. Set $\tilde{x} = (x^*, \hat{x})$ and $\tilde{w} = (w, \perp)$ and output $(\tilde{x}, \tilde{w})$. Note that $(\tilde{x}, \tilde{w})$ is chosen in the same way as for the challenger of Game 7.2 and Game 7.3.

$\mathcal{B}_1$: Given a proof $\tilde{\pi}$, run $\mathsf{M}(\mathsf{Vrf}, \tilde{\sigma}, \tilde{x}, \tilde{\pi})$. If $\mathcal{B}_1$ observes a query that includes w during the execution of M.Vrf (i.e., AskW occurs), output $b' = 1$, otherwise $b' = 0$.

We denote by $b = 1$ (resp., $b = 0$) the situation where the challenger works with M.Crs and M.Prv (resp., M.CrsSim and M.PrvSim). Observe that the distribution of $(\tilde{\sigma}, \tilde{x}, \tilde{\pi})$ in $\mathcal{B}_1$ is the same as that given to M.Vrf in Step 4 in Game 7.2 (resp., Game 7.3) if $b = 1$ (resp., $b = 0$). Thus, we obtain that

$$\Pr[b' = 1 | b = 1] = \Pr[\mathsf{AskW}^{7.2}],$$
$$\Pr[b' = 1 | b = 0] = \Pr[\mathsf{AskW}^{7.3}].$$

Considering the definition of $\mathtt{AdvZK}_{\mathcal{B},\mathsf{M},\mathcal{L}^O \vee \hat{\mathcal{L}}}$, we obtain the following formula:

$$\begin{aligned}\mathtt{AdvZK}_{\mathcal{B},\mathsf{M},\mathcal{L}^O \vee \hat{\mathcal{L}}} =& \Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0] \\ =& \Pr[\mathsf{AskW}^{7.2}] - \Pr[\mathsf{AskW}^{7.3}] > \rho_{\mathsf{zk}}\end{aligned}$$

which contradicts the assumption that M is zero-knowledge. Therefore, we obtain $\Pr[\mathsf{AskW}^{7.2}] - \Pr[\mathsf{AskW}^{7.3}] \leq \rho_{\mathsf{zk}}$. □

Now we show that $\Pr[\mathsf{AskW}^{7.3}] \leq q/2^n$. As a proof generated by M.PrvSim is independent of w, the verifier makes a query on w only by chance. As M.Vrf makes at most q queries, $\Pr[\mathsf{AskW}^{7.3}] \leq q/2^n$. Summarizing the above, we have

$$\Pr[\mathsf{AskW}^{7.0}] = \Pr[\mathsf{AskW}^{7.1}] = \Pr[\mathsf{AskW}^{7.2}] \leq \rho_{\mathsf{zk}} + \Pr[\mathsf{AskW}^{7.3}] \leq \rho_{\mathsf{zk}} + q/2^n.$$

We have thus proven Claim 4. □

Proof of Claim 5. Recall that VerPi is the event that a randomly assigned proof in Step 3 is verified by M.Vrf in Step 4. It should be analyzed carefully as we do not know whether a proof is a randomly assigned proof when M.Vrf makes a query to ZK.Vrf, i.e., VerPi is not observable. However, we observe that there exists an alternative event that almost implies VerPi.

We introduce an alternative event VerCrs and show that $\Pr[\mathsf{AskW}] \leq \Pr[\mathsf{VerCrs}] + 1/(eq^{c-1})$. Observe that a proof is randomly assigned only when a query that contains a CRS ck s.t. $(ck, [\tau]) \notin Q_{\mathsf{triv}} \cup Q_{\mathsf{intl}}$ is made in Step 3. Hence, whenever VerPi occurs, such a CRS is queried by M.Vrf. As we have already excluded the case where a legitimate CRS suddenly appears without its prior generation by given oracle in Game 1, the appearance of such a CRS in Step 4 is due to one of the following:

- ck is trivial but the adversary failed to learn (ck, τ) in Step 2
- ck is non-trivial.

The first case can be evaluated as in Game 5; thus such a query is made with probability at most $1/(eq^{c-1})$. Regarding the second case, we define VerCrs to be an event in which M.Vrf makes a query $(\mathsf{Vrf}, [ck], [x], [c], [\pi])$ to ZK s.t. $ck \in Q_{\mathsf{nt}}$ where Q_{nt} is the list of CRSs queried in Step 2 but $(ck, [\tau]) \notin Q_{\mathsf{triv}}$ ("nt" stands for non-trivial). From the above observation, we have $\Pr[\mathsf{VerPi}] \leq \Pr[\mathsf{VerCrs}] + 1/(eq^{c-1})$. Now, we evaluate Pr[VerCrs] by introducing several subgames. We first introduce the same game transition as in Game 7.1 to Game 7.3. Then, we modify the game so that it ultimately reaches the situation where a proof is generated on a (no, yes)-instance; thus VerCrs does not occur. Let Game 7.0′ be Game 7.

Game 7.1′ (the same as Game 7.1): Replace O'' and ZK″ with $O'||\mathsf{R_o}$ and $\mathsf{ZK}'||\mathsf{R_{zk}}$ respectively. Similar to Game 7.1, we have $\Pr[\mathsf{VerCrs}^{7.1}] = \Pr[\mathsf{VerCrs}^{7.0}]$.

Game 7.2′ (the same as Game 7.2): Replace $O'||\mathsf{R_o}$ and $\mathsf{ZK}'||\mathsf{R_{zk}}$ with $O \leftarrow \mathcal{O}_n$ and $\mathsf{ZK} \leftarrow \mathcal{Z}_n$. Furthermore, let the adversary run $\mathsf{M}^{O,\mathsf{ZK}}.\mathsf{Prv}$ on a correct instance, i.e., on $(x^*, \hat{x})$. Since the same discussion as in Game 7.2 can be applied, we have $\Pr[\mathsf{VerCrs}^{7.2}] = \Pr[\mathsf{VerCrs}^{7.1}]$.

Game 7.3′ (the same as Game 7.3): Replace M.Crs in Step 1 and M.Prv in Step 3 with M.CrsSim and M.PrvSim respectively. Furthermore, let the challenger pass the trapdoor generated by M.CrsSim to the adversary. We claim the following:

Claim 7 $\Pr[\mathsf{VerCrs}^{7.2'}] - \Pr[\mathsf{VerCrs}^{7.3'}] \leq \rho_{zk}$.

Proof of Claim 7. We first construct a zero-knowledge adversary $\mathcal{B}' = (\mathcal{B}'_0, \mathcal{B}'_1)$ as follows:

$\mathcal{B}'_0$: Given a CRS $\tilde{\sigma}$, execute Step 2 in the soundness game. Then, choose $w \in \{0,1\}^n$ and $\hat{x} \in \hat{\mathcal{C}}$ uniformly and obtain $x^* = O(\mathsf{SmplYes}, w)$. Set $\tilde{x} = (x^*, \hat{x})$ and $\tilde{w} = (w, \perp)$ and output $(\tilde{x}, \tilde{w})$.

$\mathcal{B}'_1$: Given a proof $\tilde{\pi}$, run $\mathsf{M}(\mathsf{Vrf}, \tilde{\sigma}, \tilde{x}, \tilde{\pi})$. Output $b' = 1$ if $\mathcal{B}'_1$ observes VerCrs, otherwise output $b' = 0$.

Similar to Game 7.3, we denote by $b = 1$ (resp., $b = 0$) the situation where the challenger runs M.Crs and M.Prv (resp., M.CrsSim and M.PrvSim). Then, we obtain

$$\Pr[b' = 1 | b = 1] = \Pr[\mathsf{VerCrs}^{7.2'}],$$
$$\Pr[b' = 1 | b = 0] = \Pr[\mathsf{VerCrs}^{7.3'}].$$

Considering the definition of $\mathtt{AdvZK}_{\mathcal{B}',\mathsf{M},\mathcal{L}^O \vee \hat{\mathcal{L}}}$, we obtain the following formula:

$$\begin{aligned}\mathtt{AdvZK}_{\mathcal{B}',\mathsf{M},\mathcal{L}^O \vee \hat{\mathcal{L}}} =& \Pr[b' = 1 | b = 1] - \Pr[b' = 1 | b = 0] \\ =& \Pr[\mathsf{VerCrs}^{7.2'}] - \Pr[\mathsf{VerCrs}^{7.3'}].\end{aligned}$$

Therefore, if $\Pr[\mathsf{VerCrs}^{7.2'}] - \Pr[\mathsf{VerCrs}^{7.3'}] > \rho_{\mathsf{zk}}$, it contradicts the zero-knowledge property of M, which justifies Claim 7. □

Game 7.4′: Modify the adversary so that it chooses $(\hat{x}, \hat{w}) \in R_{\hat{\mathcal{L}}}$ instead of $\hat{x} \in \hat{\mathcal{C}}$ in Step 2 (i.e., $\mathcal{A}$ samples an (yes, yes)-instance). Note that since $\mathcal{A}$ runs M.PrvSim, $\hat{w}$ is not given to M.PrvSim.

Recall that $\hat{\mathcal{L}}$ is a hard language, along with $\hat{\mathcal{C}}$. Thus, it contradicts the instance indistinguishability of $\hat{\mathcal{L}}$ if $|\Pr[\mathsf{VerCrs}^{7.4'}] - \Pr[\mathsf{VerCrs}^{7.3'}]| > \hat{\rho}_{\mathsf{ind}}$. Therefore, we have $|\Pr[\mathsf{VerCrs}^{7.4'}] - \Pr[\mathsf{VerCrs}^{7.3'}]| \leq \hat{\rho}_{\mathsf{ind}}$.

Game 7.5′: Modify the adversary so that it chooses $(x^*, w) \in R_{\mathcal{C}}$ instead of $(x^*, w) \in R_{\mathcal{L}}$ (i.e., the adversary samples a (no, yes)-instance). Similar to Game 7.4′, we obtain $|\Pr[\mathsf{VerCrs}^{7.5'}] - \Pr[\mathsf{VerCrs}^{7.4'}]| \leq \rho_{\mathsf{ind}}$.

Game 7.6′: Replace M.CrsSim and M.PrvSim with M.Crs and M.Prv, respectively. Note that as $\mathcal{A}$ samples a (no, yes)-instance, the adversary runs M.Prv on $\tilde{x} = (x^*, \hat{x})$ and $\tilde{w} = (\perp, \hat{w})$. Since the same discussion as in Game 7.3′ can be applied, we have $\Pr[\mathsf{VerCrs}^{7.5'}] - \Pr[\mathsf{VerCrs}^{7.6'}] \leq \rho_{\mathsf{zk}}$.

As M.Prv runs on a (no, yes)-instance, there is little chance that VerCrs will occur. That is, the probability of a query $(\mathsf{Prv}, [ck], x^*, w, [r])$ having a legitimate CRS $ck \in Q_{\mathsf{nt}}$ is bounded by $q/2^n$, as $O.\mathsf{SmplNo}$ is implemented by a random injection and M.Prv is not given w. Therefore, $\Pr[\mathsf{VerCrs}^{7.6'}] \leq q/2^n$.

To sum up the above, we have

$$\begin{aligned}&\Pr[\mathsf{VerCrs}^{7.0'}] = \Pr[\mathsf{VerCrs}^{7.1'}] = \Pr[\mathsf{VerCrs}^{7.2'}] \\ &\leq \Pr[\mathsf{VerCrs}^{7.3'}] + \rho_{\mathsf{zk}} \leq \Pr[\mathsf{VerCrs}^{7.4'}] + \rho_{\mathsf{zk}} + \hat{\rho}_{\mathsf{ind}} \\ &\leq \Pr[\mathsf{VerCrs}^{7.5'}] + \rho_{\mathsf{zk}} + \rho_{\mathsf{ind}} + \hat{\rho}_{\mathsf{ind}} \\ &\leq \Pr[\mathsf{VerCrs}^{7.6'}] + 2\rho_{\mathsf{zk}} + \rho_{\mathsf{ind}} + \hat{\rho}_{\mathsf{ind}} \\ &\leq q/2^n + 2\rho_{\mathsf{zk}} + \rho_{\mathsf{ind}} + \hat{\rho}_{\mathsf{ind}}.\end{aligned}$$

Therefore, we obtain

$$\Pr[\mathsf{VerPi}^7] \leq 1/(eq^{c-1}) + q/2^n + 2\rho_{\mathsf{zk}} + \rho_{\mathsf{ind}} + \hat{\rho}_{\mathsf{ind}}.$$

We have thus proven Claim 5. □

5.5 Conclusion and Future Work

We revealed that there is no fully black-box construction of an NIZK for a disjunctive language based on CP-NIZKs. This result suggests that we should rely on a certain mathematical structure if we want to augment the capability of NIZKs in terms of the language they prove, while a commit-and-prove methodology is itself powerful enough to break the barrier shown in Chapter 4.

There is room for considering a black-box language extension. That is, we might be able to characterize languages (or binary relations) such that we cannot obtain NIZKs for them in a black-box manner.

Chapter 6

Conclusion

It is often required that an authentication is done without revealing any private information. This seemingly contradictory requirement can be satisfied by using NIZKs. While several efficient and practical NIZKs have been proposed and are used in many applications, they depend on underlying assumptions. Hence, much is not known about the general treatment of NIZKs as building blocks.

In this dissertation, we have studied black-box constructions that use NIZKs as oracles. As mentioned earlier, extending languages that NIZKs prove is important in the following sense: (i) NIZKs for extended languages have a lot of applications in both theory and practice. Recall that the Naor-Yung construction, which is an important paradigm in theoretical cryptography, employs an NIZK for a language that includes conjunction and equality. Regarding disjunctive relation, a majority voting can be implemented by an NIZK that proves a vote is indeed 0 or 1, as mentioned in Chapter 1. Thus, if we can expand a language that an NIZK proves in a black-box manner, it could help to construct these applications efficiently. (ii) Extending languages is an important direction to construct an NIZK for an NP-complete language. An ultimate goals regarding NIZKs is to construct an NIZK for an NP-complete language, and it is often the case that such an NIZK is achieved by language extensions. Therefore, a black-box language extension could provide a generic way toward this goal. However, this dissertation have revealed that there is no universal way to construct an NIZK for certain extended languages based on NIZKs for smaller languages.

In Chapter 3, we have simplified the existing framework that takes account of an NIZK, and obtained a new insight regarding an implementation of an NIZK in a black-box framework. In the following chapters, we have demonstrated the impossibilities of language extensions of NIZKs in a black-box manner based on the knowledge that we obtained in Chapter 3. Readers may wonder that the NIZK oracles (namely, the prover interfaces) defined in Chapter 4 and 5 accepts a randomness, while we have excluded such a value from the prover interface in Chapter 3. This is because the coin-free oracle in Chapter 3 proves an NP-complete language, while the other

oracles are for specific languages. Therefore, we found out that the power of NIZKs deeply depend on languages they prove.

In Chapter 4, we have shown that it is impossible to construct an NIZK for the witness equality based on (standard) NIZKs in a black-box manner. We remark that NIZKs for the witness equality has a vast number of applications in both theory and practice. This result suggests that we should rely on algebraic structures or the feature of languages if we want to construct an NIZK for the witness equality from NIZKs for smaller languages.

In Chapter 5, we have proven that we cannot construct an NIZK for an OR-composition language even from CP-NIZKs in a black-box manner. Note that we can construct an NIZK for the witness equality if underlying NIZKs are CP-NIZKs, and thus the commit-and-prove methodology trivially breaks the barrier demonstrated in Chapter 4. Therefore, even if we use such powerful oracles, it is impossible to extend languages that underlying NIZKs prove in a black-box manner.

We stress that the existence of CP-NIZKs does not damage the meaning of the analysis in Chapter 4. That is, it was non-trivial if we can construct an NIZK for the witness equality from standard NIZKs in a black-box manner. A black-box construction provides a general transformation from a primitive to another one. Therefore, we considered the possibility of the construction of NIZKs for the witness equality from the NIZK of the most abstract form.

We remark that these results do not exclude the possibility non-black-box constructions of NIZKs for extended languages (in fact, a lot of such constructions are seen in the literature). There are many constructions of NIZKs based on specific assumptions, such as pairing or lattice. Our results rule out the possibility that there exists a universal methodology to enhance the power of NIZKs in terms of languages, which works under any assumptions.

In summary, while an NIZK is a useful cryptographic primitive and there are practical constructions of NIZKs based on concrete assumptions, we have demonstrated the limitation of enhancing the capability of NIZKs in a black-box manner. This indicates that, if we want to extend a language that an NIZK proves, we should rely on the characteristic of the underlying assumption. Therefore, we conclude that NIZKs will develop for each underlying assumption, and in particular, cryptographers should construct an NIZK from scratch if a new assumption is proposed in the future.

Appendix A

Publications List

Journal Articles

- Kyosuke Yamashita, Mehdi Tibouchi, and Masayuki Abe, "A Coin-Free Oracle-Based Augmented Black Box Framework (Full Paper)", IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E103-A, No.10, pp.1167-1173, Oct. 2020.
- Kyosuke Yamashita, Mehdi Tibouchi and Masayuki Abe, "On the Impossibility of NIZKs for Disjunctive Languages from Commit-and-Prove NIZKs", IEEE Access, 2021.

Conference Proceedings Articles

- Kyosuke Yamashita, Mehdi Tibouchi, and Masayuki Abe "A Coin-Free Oracle-Based Augmented Black Box Framework", ProvSec2019
- Kyosuke Yamashita, Mehdi Tibouchi and Masayuki Abe "On Black-Box Extension of a Non-Interactive Zero-Knowledge Proof System for Secret Equality", INDOCRYPT 2020: 882-904

Talks

- Kyosuke Yamashita, Mehdi Tibouchi, and Masayuki Abe. "On Augmented Black-Box Constructions Based on an Oracle Without Witness Indistinguishability", SCIS 2019.
- Kyosuke Yamashita, Mehdi Tibouchi, and Masayuki Abe. "On Augmented Black-Box Construction Based on an Oracle Without Witness Indistinguishability", IWSEC 2019 (invited talk).

- Kyosuke Yamashita, Mehdi Tibouchi, and Masayuki Abe. "The Augmented Black Box Framework and Zero-Knowledge Proofs of Plaintext Equality", SCIS 2020.
- Kyosuke Yamashita, Mehdi Tibouchi, and Masayuki Abe. "Limits on The Power of Commit-and-Prove NIZKs", SCIS 2021.

Chapter 3 is based on "A Coin-Free Oracle- Based Augmented Black Box Framework (Full Paper)", IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E103-A, No.10, pp.1167-1173, Oct. 2020, DOI: https://doi.org/10.1587/transfun.2019DMP0018, copyright ©2020 IEICE.

Chapter 5 is based on "On the Impossibility of NIZKs for Disjunctive Languages from Commit-and-Prove NIZKs", IEEE Access, DOI: 10.1109/ACCESS.2021.3056078, 2021 copyright ©2021 IEEE.

Bibliography

[1] G. Oded, S Micali, and A Wigderson. How to prove all np-statements in zero-knowledge, and a methodology of cryptographic protocol design. In *Proceedings on Advances in cryptology—CRYPTO '86*, pages 171–185, London, UK, UK, 1987. Springer-Verlag.

[2] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 103–112, New York, NY, USA, 1988. ACM.

[3] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, pages 415–432, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

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

[5] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 305–326, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[6] Benedikt Bunz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 315–334, 2018.

[7] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo G. Desmedt, editor, *Advances in Cryptology — CRYPTO '94*, pages 174–187, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[8] Jens Groth. Non-interactive zero-knowledge arguments for voting. ACNS'05, pages 467–482, Berlin, Heidelberg, 2005. Springer-Verlag.

[9] Helios Voting.

[10] Silvio Micali and Michael O. Rabin. Cryptography miracles, secure auctions, matching problem verification. *Commun. ACM*, 57(2):85–93, February 2014.

[11] D. C. Parkes, M. O. Rabin, S. M. Shieber, and C. A. Thorpe. *Practical Secrecy-Preserving, Verifiably Correct and Trustworthy Auctions*, pages 70–81. Association for Computing Machinery, New York, NY, USA, 2006.

[12] Helger Lipmaa, N. Asokan, and Valtteri Niemi. Secure vickrey auctions without threshold trust. In Matt Blaze, editor, *Financial Cryptography 2003, Bermuda, 11-14 March 2002*, 2003.

[13] Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. *Zcash protocol specification*, 2020.1.15 edition, 2020.

[14] Vitalik Buterin. *Ethereum White Paper*.

[15] Georg Fuchsbauer. Wi is not enough: Zero-knowledge contingent (service) payments revisited. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS ' 19, pages 49–62, New York, NY, USA, 2019. Association for Computing Machinery.

[16] Mathias Hall-Andersen. Fastswap: Concretely efficient contingent payments for complex predicates. *IACR Cryptol. ePrint Arch.*, 2019:1296, 2019.

[17] M. Campanelli, R. Gennaro, Steven Goldfeder, and Luca Nizzardo. Zero-knowledge contingent payments revisited: Attacks and payments for services. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[18] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V. Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R. Savagaonkar. Innovative instructions and software model for isolated execution. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, New York, NY, USA, 2013. Association for Computing Machinery.

[19] Arm. Arm trustzone technology. `https://developer.arm.com/ip-products/security-ip/trustzone`.

[20] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7, 06 2002.

[21] U Feige, D Lapidot, and A Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29(1):1–28, September 1999.

[22] U. Feige, D. Lapidot, and A. Shamir. Multiple non-interactive zero knowledge proofs based on a single random string. In *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*, pages 308–317 vol.1, 1990.

[23] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

[24] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *J. Cryptol.*, 13(3):361–396, January 2000.

[25] Mihir Bellare and Moti Yung. Certifying permutations: Noninteractive zero-knowledge based on any trapdoor permutation. *J. Cryptology*, 9:149–166, 06 1996.

[26] Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: Deniable encryption, and more. STOC '14, pages 475–484, New York, NY, USA, 2014. Association for Computing Machinery.

[27] Nir Bitansky, Omer Paneth, and Daniel Wichs. Perfect structure on the edge of chaos. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography*, pages 474–502, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[28] Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Non-interactive zero-knowledge with preprocessing. In Shafi Goldwasser, editor, *Advances in Cryptology — CRYPTO' 88*, pages 269–282, New York, NY, 1990. Springer New York.

[29] Rafael Pass, abhi shelat, and Vinod Vaikuntanathan. Construction of a non-malleable encryption scheme from any semantically secure one. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, pages 271–289, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[30] Charanjit S. Jutla and Arnab Roy. Shorter quasi-adaptive nizk proofs for linear subspaces. *Journal of Cryptology*, 30(4):1116–1156, Oct 2017.

[31] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, STOC '00, pages 235–244, New York, NY, USA, 2000. Association for Computing Machinery.

[32] B. Barak, R. Canetti, J. B. Nielsen, and R. Pass. Universally composable protocols with relaxed set-up assumptions. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 186–195, 2004.

[33] Michael Ben-Or and Dan Gutfreund. Trading help for interaction in statistical zero-knowledge proofs. *Journal of Cryptology*, 16:95–116, 03 2008.

[34] Jens Groth and Rafail Ostrovsky. Cryptography in the multi-string model. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, pages 323–341, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[35] Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. Nizks with an untrusted crs: Security in the face of parameter subversion. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, pages 777–804, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[36] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC '89, pages 44–61, New York, NY, USA, 1989. ACM.

[37] J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 387–394, New York, NY, USA, 1990. Association for Computing Machinery.

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

[39] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptol.*, 4(2):151–158, January 1991.

[40] Manuel Blum and Shafi Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In George Robert Blakley and David Chaum, editors, *Advances in Cryptology*, pages 289–299, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[41] Eyal Kushilevitz and Rafail Ostrovsky. One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000*, pages 104–121, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[42] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *Advances in Cryptology — EUROCRYPT'98*, pages 334–345, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[43] Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 325–335, 2000.

[44] Mohammad Mahmoody, Ameer Mohammed, and Soheil Nematihaji. On the impossibility of virtual black-box obfuscation in idealized models. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography*, pages 18–48, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[45] R. Impagliazzo. A personal view of average-case complexity. *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 134–147, 1995.

[46] Yael Gertner, Tal Malkin, and Steven Myers. Towards a separation of semantic and cca security for public key encryption. In *Proceedings of the 4th Conference on Theory of Cryptography*, TCC'07, pages 434–455, Berlin, Heidelberg, 2007. Springer-Verlag.

[47] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 427–437, New York, NY, USA, 1990. ACM.

[48] Yehuda Lindell. A simpler construction of cca2-secure public-key encryption under general assumptions. In *Proceedings of the 22Nd International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'03, pages 241–254, Berlin, Heidelberg, 2003. Springer-Verlag.

[49] Zvika Brakerski, Jonathan Katz, Gil Segev, and Arkady Yerukhimovich. Limits on the power of zero-knowledge proofs in cryptographic constructions. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011*, volume 6597 of *Lecture Notes in Computer Science*, page 559. Springer, 2011.

[50] Masayuki Abe, Miguel Ambrona, and Miyako Ohkubo. On black-box extensions of non-interactive zero-knowledge arguments, and signatures directly from simulation soundness. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *Public-Key Cryptography – PKC 2020*, pages 558–589, Cham, 2020. Springer International Publishing.

[51] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS '99, pages 543–, Washington, DC, USA, 1999. IEEE Computer Society.

[52] Rafael Pass. Unprovable security of perfect nizk and non-interactive non-malleable commitments. In Amit Sahai, editor, *Theory of Cryptography*, pages 334–354, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[53] Nir Bitansky, Dana Dachman-Soled, Sanjam Garg, Abhishek Jain, Yael Tauman Kalai, Adriana López-Alt, and Daniel Wichs. Why "fiat-shamir for proofs" lacks a proof. In Amit Sahai, editor, *Theory of Cryptography*, pages 182–201, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[54] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. STOC '11, pages 99–108, New York, NY, USA, 2011. Association for Computing Machinery.

[55] Geoffroy Couteau and Dennis Hofheinz. Designated-verifier pseudorandom generators, and their applications. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 562–592, Cham, 2019. Springer International Publishing.

[56] Willy Quach, Ron D. Rothblum, and Daniel Wichs. Reusable designated-verifier nizks for all np from cdh. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 593–621, Cham, 2019. Springer International Publishing.

[57] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Designated verifier/prover and preprocessing nizks from diffie-hellman assumptions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 622–651, Cham, 2019. Springer International Publishing.

[58] Sam Kim and David J. Wu. Multi-theorem preprocessing nizks from lattices. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 733–765, Cham, 2018. Springer International Publishing.

[59] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for np from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 89–114, Cham, 2019. Springer International Publishing.

[60] Oded Goldreich, S. Micali, and Avi Wigderson. How to play any mental game. pages 218–229, 01 1987.

[61] Joe Kilian. *Uses of Randomness in Algorithms and Protocols*. PhD thesis, Cambridge, MA, USA, 1990.

[62] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, 08 2003.

[63] Omer Reingold, Luca Trevisan, and Salil Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *Theory of Cryptography*, pages 1–20, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[64] Paul Baecher, Christina Brzuska, and Marc Fischlin. Notions of black-box reductions, revisited. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, pages 296–315, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[65] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *Advances in Cryptology — EUROCRYPT'98*, pages 334–345, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[66] Dan Boneh, Periklis A. Papakonstantinou, Charles Rackoff, Yevgeniy Vahlis, and Brent Waters. On the impossibility of basing identity based encryption on trapdoor permutations. *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 283–292, 2008.

[67] Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, FOCS '15, pages 191–209, Washington, DC, USA, 2015. IEEE Computer Society.

[68] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ameer Mohammed. Limits on the power of garbling techniques for public-key encryption. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 335–364, Cham, 2018. Springer International Publishing.

[69] Chun-Yuan Hsiao and Leonid Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In Matt Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, pages 92–105, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[70] I. Haitner, J. J. Hoch, O. Reingold, and G. Segev. Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 669–679, 2007.

[71] Mohammad Mahmoody and Rafael Pass. The curious case of non-interactive commitments – on the power of black-box vs. non-black-box use of primitives. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, pages 701–718, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[72] Mohammad Mahmoody and Ameer Mohammed. On the power of hierarchical identity-based encryption. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 243–272, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[73] R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 305–313, 2000.

[74] Moni Naor. On cryptographic assumptions and challenges. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, pages 96–109, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[75] Michael Luby. *Pseudorandomness and Cryptographic Applications*. 01 1996.

[76] Jean-Sébastien Coron. Optimal security proofs for pss and other signature schemes. In Lars R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, pages 272–287, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[77] Christoph Bader, Tibor Jager, Yong Li, and Sven Schäge. On the impossibility of tight cryptographic reductions. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 273–304, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[78] Saqib A. Kakvi and Eike Kiltz. Optimal security proofs for full domain hash, revisited. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 537–553, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[79] Andrew Morgan, Rafael Pass, and Elaine Shi. On the adaptive security of macs and prfs. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 724–753, Cham, 2020. Springer International Publishing.

[80] Iftach Haitner, Thomas Holenstein, Omer Reingold, Salil Vadhan, and Hoeteck Wee. Universal one-way hash functions via inaccessible entropy. EUROCRYPT'10, pages 616–637, Berlin, Heidelberg, 2010. Springer-Verlag.

[81] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. On the cryptographic applications of random functions (extended abstract). In George Robert Blakley and David Chaum, editors, *Advances in Cryptology*, pages 276–288, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[82] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3):690–728, July 1991.

[83] R. Cramer, D. Hofheinz, and Eike Kiltz. A note on bounded chosen ciphertext security from black-box semantical security. *IACR Cryptol. ePrint Arch.*, 2006:391, 2006.

[84] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. In *Proceedings of the Twenty-third Annual ACM Symposium on Theory of Computing*, STOC '91, pages 542–552, New York, NY, USA, 1991. ACM.

[85] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, pages 13–25, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[86] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, pages 45–64, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[87] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.*, 33(1):167–226, January 2004.

[88] Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan L. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, pages 207–222, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[89] Mihir Bellare and Shafi Goldwasser. New paradigms for digital signatures and message authentication based on non-interactive zero knowledge proofs. In Gilles Brassard, editor, *Advances in Cryptology — CRYPTO' 89 Proceedings*, pages 194–211, New York, NY, 1990. Springer New York.

[90] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, pages 552–565, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[91] B. Barak. How to go beyond the black-box simulation barrier. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 106–115, 2001.

[92] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, pages 60–77, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[93] U. Feige and A. Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, pages 416–426, New York, NY, USA, 1990. ACM.

[94] Whitfield Diffie and Martin Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22:644 – 654, 1976.

[95] Andrew C. Yao. Theory and application of trapdoor functions. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 80–91, Washington, DC, USA, 1982. IEEE Computer Society.

[96] Arkady Yerukhimovich. *A STUDY OF SEPARATIONS IN CRYPTOGRAPHY: NEW RESULTS AND NEW MODELS*. PhD thesis, Graduate School of the University of Maryland, 2011.

[97] A. C. Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167, 1986.

[98] Couteau Geoffro, Farshim Pooya, and Mahmoody Mohammad. Black-box uselessness: Composing separations in cryptography. In *ITCS*, 2021.

[99] Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972.

[100] Judy Goldsmith and Deborah Joseph. Three results on the polynomial isomorphism of complete sets. In *Annual Symposium on Foundations of Computer Science (Proceedings)*, pages 390 – 397, 11 1986.

[101] Jens Groth. Simulation-sound nizk proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, pages 444–459, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[102] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for np. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 339–358, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[103] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for np from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 89–114, Cham, 2019. Springer International Publishing.

[104] Sanjam Garg and Divya Gupta. Efficient round optimal blind signatures. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, pages 477–495, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[105] Dennis Hofheinz and Tibor Jager. Tightly secure signatures and public-key encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, pages 590–607, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[106] Shashank Agrawal, Chaya Ganesh, and Payman Mohassel. Non-interactive zero-knowledge proofs for composite statements. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 643–673, Cham, 2018. Springer International Publishing.

[107] Matteo Campanelli, Dario Fiore, and Anaïs Querol. Legosnark: Modular design and composition of succinct zero-knowledge proofs. In *CCS '19*, 2019.

[108] David Chaum and Torben Pryds Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *Advances in Cryptology — CRYPTO' 92*, pages 89–105, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[109] Olivier Blazy, David Derler, Daniel Slamanig, and Raphael Spreitzer. Non-interactive plaintext (in-)equality proofs and group signatures with verifiable controllable linkability. In Kazue Sako, editor, *Topics in Cryptology - CT-RSA 2016*, pages 127–143, Cham, 2016. Springer International Publishing.

[110] Seung Geol Choi, Ariel Elbaz, Ari Juels, Tal Malkin, and Moti Yung. Two-party computing with encrypted data. In Kaoru Kurosawa, editor, *Advances in Cryptology – ASIACRYPT 2007*, pages 298–314, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[111] Rafail Ostrovsky, Silas Richelson, and Alessandra Scafuro. Round-optimal black-box two-party computation. In *CRYPTO*, 2015.

[112] David C. Parkes, Michael O. Rabin, Stuart M. Shieber, and Christopher Thorpe. Practical secrecy-preserving, verifiably correct and trustworthy auctions. *Electronic Commerce Research and Applications*, 7(3):294 – 312, 2008. Special Section: New Research from the 2006 International Conference on Electronic Commerce.

[113] Dakshita Khurana, Rafail Ostrovsky, and Akshayaram Srinivasan. Round optimal black-box "commit-and-prove". In Amos Beimel and Stefan Dziembowski, editors, *Theory of Cryptography*, pages 286–313, Cham, 2018. Springer International Publishing.

[114] Craig Costello, Cédric Fournet, Jon Howell, Markulf Kohlweiss, Benjamin Kreuter, Michael Naehrig, Bryan Parno, and Samee Zahur. Geppetto: Versatile verifiable computation. In *2015 IEEE Symposium on Security and Privacy*, pages 253–270, May 2015.

[115] Helger Lipmaa. Prover-efficient commit-and-prove zero-knowledge snarks. In David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *Progress in Cryptology – AFRICACRYPT 2016*, pages 185–206, Cham, 2016. Springer International Publishing.

[116] Michel Abdalla, Fabrice Benhamouda, and David Pointcheval. Disjunctions for hash proof systems: New constructions and applications. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 69–100, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[117] Jan Camenisch, Nishanth Chandran, and Victor Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009*, pages 351–368, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[118] Alex Escala and Jens Groth. Fine-tuning groth-sahai proofs. In Hugo Krawczyk, editor, *Public-Key Cryptography – PKC 2014*, pages 630–649, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[119] Matteo Campanelli Daniel Benarroch and Dario Fiore. Proposal: Commit-and-prove zero-knowledge proof systems. 2nd edition of the ZK proof workshop, 2020.

[120] Evgene Vahlis. *Cryptography: Leakage Resilience, Black Box Separations, and Credential-free Key Exchange*. PhD thesis, Graduate Department of Computer Science University of Toronto, 2011.

[121] Victor Shoup. Sequences of games: A tool for taming complexity in security proofs. *IACR Cryptology ePrint Archive*, 2004:332, 01 2004.