ABSTRACT

Title of dissertation: A STUDY OF SEPARATIONS IN CRYPTOGRAPHY:
                            NEW RESULTS AND NEW MODELS

                            Arkady Yerukhimovich, Doctor of Philosophy, 2011

Dissertation directed by: Professor Jonathan Katz
                            Department of Computer Science

For more than 20 years, black-box impossibility results have been used to argue the

infeasibility of constructing certain cryptographic primitives (e.g., key agreement) from

others (e.g., one-way functions). In this dissertation we further extend the frontier of this

field by demonstrating several new impossibility results as well as a new framework for

studying a more general class of constructions.

Our first two results demonstrate impossibility of black-box constructions of two

commonly used cryptographic primitives. In our first result we study the feasibility

of black-box constructions of predicate encryption schemes from standard assumptions

and demonstrate strong limitations on the types of schemes that can be constructed. In

our second result we study black-box constructions of constant-round zero-knowledge

proofs from one-way permutations and show that, under commonly believed complex-

ity assumptions, no such constructions exist.

A widely recognized limitation of black-box impossibility results, however, is that

they say nothing about the usefulness of (known) *non*-black-box techniques. This state

of affairs is unsatisfying as we would at least like to rule out constructions using the set

of techniques we have at our disposal. With this motivation in mind, in the final result of this dissertation we propose a new framework for black-box constructions with a non-black-box flavor, specifically, those that rely on *zero-knowledge proofs* relative to some oracle. Our framework is powerful enough to capture a large class of known constructions, however we show that the original black-box separation of key agreement from one-way functions still holds even in this non-black-box setting that allows for zero-knowledge proofs.

A Study of Separations in Cryptography:
New Results and New Models

by

Arkady Yerukhimovich

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:
Professor Jonathan Katz, Chair/Advisor
Professor William Gasarch
Professor David Mount
Professor Abhi Shelat
Professor Lawrence Washington

Acknowledgments

First and foremost I would like to thank my advisor Jonathan Katz. I joined
Jonathan's group because of his enthusiasm about his research and his interest in and
willingness to work on problems outside his immediate area of expertise. I am now
certain that this was the right decision for me. Jonathan taught me how to approach a
difficult problem, to truly understand what is involved and to find the correct tools and
techniques to solve it. His suggestions and advice were always useful at each step of
this process, from selecting the problem to surmounting the many technical hurdles to
presenting the result in a clear and understandable way. His availability and willingness
to explain even basic concepts made the experience of working with him a pleasure and
I consider myself very lucky to have had Jonathan as an advisor.

I would like to thank Bill Gasarch for the many entertaining conversations and tid-
bits of advice that he has given me over the years. He was always happy to talk and give
frank answers to the many questions I had, both about my own research and the overall
graduate school experience. I also want to thank John Savage and Ramani Duraiswami
for introducing me to the world of academic research and for guiding me through my
first few attempts. Additionally, I want to thank to Abhi Shelat, Larry Washington and
Dave Mount for serving on my dissertation committee.

I would like to thank Dov Gordon for many years of friendship and collaboration.
We started graduate school together with similar interests, and from the beginning he
was there to discuss both research and non-research topics. I always enjoyed working
with him and have benefitted greatly from having such a good friend and colleague. I
would also like to thank my officemates Amy Alford, Adam Groce, Ranjit Kumaresan

and Martin Paraskevov for the many hours spent working together and for keeping the office a fun and productive place. Thank you to Dominique Schroeder, Seung Geol Choi, Hong-Sheng Zhou, and Vassilis Zikas for making this last year of my PhD such an exciting and productive year. I benefitted greatly from working with all of you and learning about your various areas of expertise and approaches to research. Special thanks to Dominique for reviewing an early version of this dissertation.

I have had the pleasure of many exciting collaborations during the course of my studies. I would like to thank Dr. Andrew Yao for inviting me to visit the Institute for Theoretical Computer Science. The summer I spent there broadened my horizon and introduced me to a variety of topics to which I had not previously been exposed. I would especially like to thank Andrej Bogdanov, Elad Verbin and David Xiao for making that summer a fun and educational experience. I would also like to thank my many co-authors and collaborators. Thank you to Zvika Brakerski, Gil Segev, Hoeteck Wee, David, Dominique, Seung Geol, Hong-Sheng, Adam, Ranjit and Dov. I have benefited greatly from working with all of you.

Most importantly I would like to thank my parents, Boris and Faina, and my grandmothers, Raisa and Musya, for encouraging me throughout the course of my studies. Even if they did not understand what I was working on, they were always ready to offer their full-fledged support. Finally, I would like to thank my wife, Priscilla, for her tremendous support during this process. Thank you for encouraging me when the going was tough and for sharing my joy upon my successes. Thank you for your patience and for the occasional push in the right direction when I would get distracted or discouraged. Thank you for always being there for me and for helping me maintain my sanity through

this lengthy process. I could not have done this without you.

<h1>Table of Contents</h1>

List of Figures

Chapter 1

Introduction

A central goal of theoretical cryptography is to explore relationships between various cryptographic primitives and, in particular, to show constructions of various "high-level" cryptographic objects (encryption schemes, key agreement protocols, etc.) based on "low-level" cryptographic tools (such as one-way functions). This line of research has been very successful, and we now know, for example, that one-way functions suffice for constructing all the primitives of private-key cryptography [121, 18, 59, 56, 73] as well as digital signature schemes [92, 109]. In other cases, however, constructions of certain primitives from others are unknown: for example, we do not currently know how to construct public-key encryption schemes based on one-way functions. Given this failure, it is natural to wonder whether such constructions are inherently *impossible*. Unfortunately, we cannot rule out *all* such constructions as long as we believe that the object in question exists in the real world: if we believe that RSA encryption (say) is secure, then a valid construction of public-key encryption from any one-way function f consists of simply ignoring f and outputting the code for the RSA encryption scheme. Yet this is clearly not what is intended.

In an effort to capture what is meant by a "natural" construction of one primitive from another, Impagliazzo and Rudich [76] formalized the notion of a *black-box* construction. Informally, a black-box construction of primitive Q from primitive P is a construction of Q that uses only the input/output characteristics of an implementation of P, but

does not rely on any internal details as to how P is implemented. Moreover, Q should be "secure" as long as P is "secure" (each in their respective senses). This notion allowed Impagliazzo and Rudich to reason about the existence of such constructions. They demonstrated the power of this model by showing that there does not exist a black-box construction of key agreement from one-way functions. Their work opened a wealth of research opportunities to study the relationships among the various primitives that are used in cryptography. This has led to many interesting results demonstrating separations between primitives [117, 85, 52, 54, 49, 53, 68, 9, 10]. We review the techniques used by these and other results in Chapter 3 and show two new separations in Chapters 4 and 5. As the majority of known constructions in cryptography are in fact black-box, such results give strong evidence that drastically new techniques will be needed for these constructions.

However, a recognized drawback of existing black-box impossibility results is that they say nothing regarding whether these results might be circumvented using *non-black-box* techniques. While it is true that most constructions in cryptography are black-box, we have examples of non-black-box constructions as well. One striking example is given by the observation that all known constructions of CCA-secure public-key encryption schemes based on trapdoor permutations [93, 37, 114, 88] are, in fact, not black-box. (Interestingly, a partial black-box separation is known [53].) Other non-black-box constructions include those of [39, 12, 11, 2, 44, 5]. For a more detailed summary see Section 6.2.

If black-box constructions are supposed to be representative of existing techniques, we should update our definition of what "black-box" means. In the final result of this dis-

sertation, we propose a framework to do exactly this, allowing us to go beyond black-box separations. Specifically, we suggest a model that incorporates a rich class of non-black-box techniques: those that rely on zero-knowledge proofs. We accomplish this by augmenting the basic, black-box model — in which there is only an oracle $\mathcal{O}$ implementing some primitive P — with a *zero-knowledge (ZK) oracle* that allows parties to prove statements relative to $\mathcal{O}$ in zero knowledge. (Technically, a ZK oracle allows zero-knowledge proofs for any language in $\mathsf{NP}^{\mathcal{O}}$.) We call any construction using black-box access to $\mathcal{O}$ *and its associated ZK oracle* an **augmented black-box** construction. Given primitives P and Q, we can then ask whether there exists an augmented black-box construction of Q from P; an impossibility result demonstrating that no such construction exists rules out a broader class of approaches to constructing one from the other. Since the technique of using zero-knowledge proofs is by far the most commonly used non-black-box construction technique, our framework captures a meaningfully larger class of constructions than [76]. Of course, as with all impossibility results, such a result says nothing about whether some *other* non-black-box techniques might apply (and, in fact, the non-black-box results of, e.g., [11, 2, 5] do not fall within our framework); nevertheless, impossibility results are still useful insofar as they show us where we must look and what dead ends we must avoid if we hope to circumvent them.

1.1 Summary of Contributions

In this dissertation, we study the power of black-box and augmented black-box constructions in relating several cryptographic primitives. We begin with some preliminary definitions in Chapter 2. Then, in Chapter 3, we review the definitions of black-box

constructions and the techniques used to prove black-box separation results. We additionally give a brief survey of the prior work in this area. Then in Chapters 4, 5 and 6 we present our results:

- In Chapter 4, we investigate the possibility of constructing secure predicate encryption schemes [24, 81] from trapdoor permutations or CCA-secure encryption. In a predicate encryption scheme every ciphertext is associated with an attribute I and every secret key corresponds to a predicate f. A secret key SK_f can decrypt a ciphertext associated with attribute I if and only if $f(I) = 1$. We identify a combinatorial property on the predicates and attributes of a predicate encryption scheme such that a black-box construction of predicate encryption from trapdoor permutations is impossible. To demonstrate the usefulness of this property we show that it is in fact satisfied by several important special cases of predicate encryption such as identity-based encryption [116, 21], forward-secure encryption [30] and broadcast encryption [42]. A preliminary version of this work has appeared previously [83].

- In Chapter 5, we investigate the round complexity of black-box constructions of zero-knowledge proofs [63]. Specifically, we look at the feasibility of constructing constant-round zero-knowledge proofs from one-way permutations. We identify the adaptivity of the simulator's queries to the cheating verifier as a key property for studying such constructions. We then show that, under widely believed complexity assumptions, there is no black-box construction of constant-round zero-knowledge proofs with constant (and even logarithmic) simulator adaptivity from one-way permutations. In fact, even if we do not restrict the simulator adaptivity,

4

we show that such a construction would lead to a major breakthrough in complexity theory and is thus likely to be difficult to find. A preliminary version of this work has appeared previously [65].

- In Chapter 6, we introduce a new framework for separation results that allows us to go beyond traditional black-box separations and prove separations for a richer class of constructions. Specifically, we define *augmented black-box* constructions to capture the class of non-black-box constructions using zero-knowledge proofs relative to a base primitive. We validate this model by demonstrating that it indeed captures known non-black-box constructions such as the construction of CCA-secure encryption from trapdoor permutations [93, 114]. Then, we initiate the study of augmented black-box separations by showing that there is no augmented black-box construction of secure (perfect completeness) key agreement from one-way functions. A preliminary version has appeared previously [26].

Chapter 2

Preliminaries

2.1 Notation

Throughout this thesis, we let $n \in \mathbb{N}$ denote the security parameter. Efficient computation is modeled by a probabilistic polynomial time (ppt) Turing machine M receiving 1^n as an input. A ppt Turing machine is one for which there exists a polynomial poly such that, for all inputs x and all random tapes r, $M(x; r)$ runs in time bounded by $\mathsf{poly}(|x|)$. Note that by giving M the string 1^n as an input, we guarantee that M can run in time at least $\mathsf{poly}(n)$. We will also need the following two definitions. An expected polynomial time Turing machine M is one for which there exists a polynomial poly such that, for all inputs x, the expected running time of $M(x; r)$ (over the choice of r) is bounded by $\mathsf{poly}(|x|)$. A non-uniform Turing machine is a pair $(M, \overline{a})$ where M is a two-input polynomial time Turing machine and $\overline{a} = a_1, a_2, \ldots$ is an infinite sequence of strings such that there exists a polynomial poly for which $|a_n| \leq \mathsf{poly}(n)$ for all n. On input x, we define the output of this machine to be $M(x, a_{|x|})$ where $a_{|x|}$ is a non-uniform advice string depending on the length of x. Since non-uniform Turing machines are equivalent to (non-uniform) families of circuits, we will often refer to such machines as circuits rather than Turing machines.

We write $\{0, 1\}^n$ to denote the set of binary strings of length n and $\{0, 1\}^*$ to indicate the set of all finite, binary strings. For a set S, we write $x \leftarrow S$ to indicate that the value

of x is sampled uniformly from S. We use $\langle A(x_a), B(x_b)\rangle(x) = (y_a, y_b)$ to represent an interactive protocol between two interactive Turing machines A and B on common input x, where A also has private input x_a and receives output y_a and B has private input x_b and private output y_b. When only one of the parties receives output we will abuse notation to write $\langle A, B\rangle = a$ to indicate this single output.

Negligible Functions: We will often need to argue that an event occurs with very low probability. For this purpose, we use the following definition of a negligible function to indicate a function that goes to 0 faster than any inverse polynomial.

Definition 2.1.1 (Negligible function) *We call a function* $\mathsf{negl} : \mathbb{N} \to \mathbb{R}^+$ negligible *if for every polynomial* poly, *there exists an N such that for all $n > N$,* $\mathsf{negl}(n) < \frac{1}{\mathsf{poly}(n)}$.

We say that a function $g : \mathbb{N} \to \mathbb{R}^+$ is *overwhelming* if $1 - g(\cdot)$ is negligible.

Additionally, we will need to argue that some events occur with noticeable probability. For this purpose, we use the following definition of a noticeable function to indicate a function that is lower bounded by an inverse polynomial (for large enough n). Note that it is possible for a function to be neither negligible nor noticeable.

Definition 2.1.2 (Noticeable function) *We say that a function* $f : \mathbb{N} \to \mathbb{R}^+$ *is* noticeable *if there exists a polynomial* poly *and an N such that for all $n > N$,* $f(n) > \frac{1}{\mathsf{poly}(n)}$.

Oracle Algorithms: In this dissertation we will often talk about oracle algorithms or oracle Turing machines. An oracle Turing machine is a Turing machine M that is allowed to make oracle queries to a function $f : \{0,1\}^* \to \{0,1\}^*$. Whenever M makes a query x to f, it receives the answer $f(x)$ in a single computation step. We write M^f to indicate

a machine M with oracle access to f. We will also use this notation to represent black-box access to another Turing machine. That is M^A will be used to indicate a machine M that may make "oracle" queries to a (possibly inefficient) machine A. Such a query will only take a single time step and we assume that M does not see how the computation is actually performed.

In this thesis, we make extensive use of the following oracles. A *random oracle*, denoted by $\mathcal{O} : \{0,1\}^* \rightarrow \{0,1\}^*$, is an oracle evaluating a random length-preserving function. That is, $\mathcal{O} \stackrel{\text{def}}{=} \{\mathcal{O}_n : \{0,1\}^n \rightarrow \{0,1\}^n\}_{n\in\mathbb{N}}$ where each $\mathcal{O}_n$ is chosen uniformly at random from the set of all length-preserving functions on $\{0,1\}^n$. A PSPACE-complete oracle, denoted by PSPACE, is an oracle deciding membership in some PSPACE-complete language such as Quantified Boolean Formula.

2.2   Probabilistic Lemmas

We now provide several probabilistic lemmas that we will use in this dissertation. The first such lemma is the Borel-Cantelli Lemma. This lemma says that for any infinite sequence of events if the sum of their probabilities is finite then the probability that infinitely many of them happen is 0. The following statement of the lemma is due to [75].

Lemma 2.2.1 (Borel-Cantelli Lemma) *Let* $B_1, B_2, \ldots$ *be a sequence of events on the same probability space. Then* $\sum_{n=1}^{\infty} \Pr[B_n] < \infty$ *implies that* $\Pr[\bigwedge_{k=1}^{\infty} \bigvee_{n\geq k} B_n] = 0.$

A second lemma that we will use is Markov's inequality. This inequality bounds the probability that a random variable significantly deviates from its expected value.

Lemma 2.2.2 (Markov's Inequality) *Let X be a random variable assuming only non-negative values. Then for any $t > 0$, $\Pr[X \geq t] \leq \frac{E[X]}{t}$, where $E[X]$ denotes the expectation of X.*

2.3 Cryptographic Primitives

In this section we define some basic cryptographic primitives that will be discussed throughout this dissertation. We leave the definitions of more complicated primitives to the corresponding chapters. Our definitions follow the presentation of [79].

2.3.1 Basic Primitives

We begin with the definitions of two very basic cryptographic primitives. The first of these primitive is a *one-way function* which is the most basic of all cryptographic primitives and is necessary for all constructions that we will discuss.

Definition 2.3.1 *A function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ is a* one-way function *(OWF) if the following two conditions hold:*

- *(Easy to Compute:) There exists a polynomial-time algorithm M_f such that $M_f(x) = f(x)$ for all x.*

- *(Hard to Invert:) For every* ppt *algorithm $\mathcal{A}$, there exists a negligible function* negl *such that*

$$\Pr_{x \leftarrow \{0,1\}^n}[\mathcal{A}(f(x)) \in f^{-1}(f(x))] \leq \mathsf{negl}(n).$$

If, for each n, f is a function from $\{0,1\}^n$ to $\{0,1\}^n$ then we say that n is a *length-preserving* one-way function. If, for each n, f is a permutation on $\{0,1\}^n$ then we call this a *one-way permutation* (OWP).

The next primitive we define is a pseudorandom generator [18, 121]. A *pseudorandom generator* (PRG) is a deterministic algorithm that receives a short truly random seed and stretches it into a long pseudorandom string, where a pseudorandom string is one that is computationally indistinguishable from a random string of the same length. To make this definition formal we first define what it means for two probability ensembles to be computationally indistinguishable.

Definition 2.3.2 *Two probability ensembles* $X \stackrel{\text{def}}{=} \{X_n\}_{n \in \mathbb{N}}$ *and* $Y \stackrel{\text{def}}{=} \{Y_n\}_{n \in \mathbb{N}}$ *are computationally indistinguishable if, for every* ppt *distinguisher D there exists a negligible function* negl *such that:*

$$|\Pr[D(1^n, X_n) = 1] - \Pr[D(1^n, Y_n) = 1]| \leq \mathsf{negl}(n)$$

where the notation $D(1^n, X_n)$ *means that* x *is chosen according to distribution* X_n *and then* $D(1^n, x)$ *is run.*

Additionally, we say that X *and* Y *are indistinguishable for non-uniform distinguishers, if the above holds for any non-uniform polynomial time distinguisher* D.

We let U_n denote the uniform distribution over $\{0,1\}^n$. We can now define a pseudorandom generator as follows.

Definition 2.3.3 *Let* $l(\cdot)$ *be a polynomial and let* G *be a deterministic polynomial-time algorithm such that for any* $s \in \{0,1\}^n$, *G outputs a string of length* $l(n)$. *G is a* pseudorandom generator *(PRG) if:*

- *(Expansion:) For every* $n \in \mathbb{N}$ *it holds that* $l(n) > n$.

- *(Pseudorandomness:) The ensemble* $\{G(U_n)\}_{n \in \mathbb{N}}$ *is computationally indistinguishable*

from the ensemble $U \stackrel{\text{def}}{=} \{U_{l(n)}\}_{n \in \mathbb{N}}$.

The random input s given to G is called the seed. The function $l(\cdot)$ is called the expansion factor of G.

2.3.2 Public-Key Primitives

We now shift to the public-key world and define a few commonly used primitives. First we define secure key agreement, which is a protocol allowing two parties, Alice and Bob, to agree on a secret key in the presence of an eavesdropper. This primitive will figure extensively in our discussion of black-box separations and also in our results in Chapter 6.

Definition 2.3.4 *A* key agreement *protocol* Π *is a pair of algorithm* (A, B) *for Alice and Bob respectively. On input* 1^n, A *and* B *choose independent random coins, participate in an interactive protocol and output* $k_A, k_B \in \{0,1\}^n$ *respectively. Using previously defined notation,* Π *is the protocol* $\langle A(r_A), B(r_B) \rangle (1^n) = (k_A, k_B)$. *We require that* Π *have* perfect completeness. *That is, for all random strings for Alice and Bob we have* $k_A = k_B$.

We say that a key agreement protocol Π *is* secure in the presence of an eavesdropper *if for every* ppt *eavesdropper Eve there exists a negligible function* negl *such that*

$$\Pr[KA_{\mathcal{A},\Pi}^{eav}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$

where $KA_{\mathcal{A},\Pi}^{eav}(n)$ *is the following experiment.*

1. *On input* 1^n, A *and* B *execute protocol* Π. *This results in output* (trans, k) *where* trans *is a transcript of all the messages exchanged and* k *is the output key.*

2. *A random bit $b \leftarrow \{0,1\}$ is chosen. If $b = 0$ then choose $\hat{k} \leftarrow \{0,1\}^n$ and if $b = 1$ set*

   $\hat{k} = k$.

3. *Eve is given $(\mathsf{trans}, \hat{k})$ and outputs a bit b'.*

4. *The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.*

Next, we define two standard security notions for public-key encryption. Namely, chosen-plaintext (CPA) and chosen-ciphertext (CCA) secure encryption. What we call CCA-security is also commonly known as CCA-2 security.

Definition 2.3.5 *A* public-key encryption scheme *is a tuple of* ppt *algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ such that:*

- *$\mathcal{G}$ takes as input the security parameter 1^n and outputs the public and secret keys (pk, sk).*

- *$\mathcal{E}$ takes as input a public key pk and a message m and outputs a ciphertext c. We write $c = \mathcal{E}_{pk}(m)$. If we wish to explicitly specify the randomness r used by $\mathcal{E}$ we will write $\mathcal{E}_{pk}(m; r)$.*

- *$\mathcal{D}$ takes as input a ciphertext c and the secret key sk and outputs a message $m = \mathcal{D}_{sk}(c)$.*

In this dissertation we require that the encryption scheme have perfect correctness. *That is, for any pair of keys (pk, sk) output by $\mathcal{G}$,*

$$\Pr[\mathcal{D}_{sk}(\mathcal{E}_{pk}(m)) = m] = 1.$$

First, we define what it means for a public-key encryption scheme to be secure under a chosen-plaintext attack.

Definition 2.3.6 *A public-key encryption scheme* $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ *is* CPA-secure *if for all* ppt *adversaries* $\mathcal{A}$, *there exists a negligible function* negl *such that:*

$$\Pr[PubK_{\mathcal{A},\Pi}^{cpa}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$

where, $PubK_{\mathcal{A},\Pi}^{cpa}(n)$ *is the output of the following experiment*

1. $\mathcal{G}(1^n)$ *is run to obtain* (pk, sk).

2. $\mathcal{A}$ *is given* pk *and outputs a pair of messages* (m_0, m_1) *of the same length. Note that* $\mathcal{A}$ *can evaluate* $\mathcal{E}_{pk}(\cdot)$ *since it knows* pk.

3. *A random* $b \leftarrow \{0, 1\}$ *is chosen and then* $c = \mathcal{E}_{pk}(m_b)$ *is given to the adversary.* c *is called the* challenge ciphertext.

4. $\mathcal{A}$ *outputs a bit* b'.

5. *The output of the experiment is 1 if* $b' = b$ *and 0 otherwise.*

A stronger notion of security for public-key encryption is that of chosen ciphertext security where the adversary is additionally given access to a decryption oracle. Formally,

Definition 2.3.7 *A public-key encryption scheme* $\Pi = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ *is* CCA-secure *if for all* ppt *adversaries* $\mathcal{A}$, *there exists a negligible function* negl *such that:*

$$\Pr[PubK_{\mathcal{A},\Pi}^{cca}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n)$$

where, $PubK_{\mathcal{A},\Pi}^{cca}(n)$ *is the output of the following experiment*

1. $\mathcal{G}(1^n)$ is run to obtain (pk, sk).

2. $\mathcal{A}$ is given pk and access to a decryption oracle $\mathcal{D}_{sk}(\cdot)$ and outputs a pair of messages (m_0, m_1) of the same length.

3. A random $b \leftarrow \{0, 1\}$ is chosen and then $c = \mathcal{E}_{pk}(m_b)$ is given to the adversary.

4. $\mathcal{A}$ continues to interact with the decryption oracle, but may not request a decryption of c. Finally, $\mathcal{A}$ outputs a bit b'.

5. The output of the experiment is 1 if $b' = b$ and 0 otherwise.

2.3.3 Zero-Knowledge Proofs

Here we give a definition of *zero-knowledge proofs* [63] for a language L that we will use in this dissertation. A zero-knowledge proof is a two-party protocol in which one party, the prover, can convince the other party, the verifier, that some statement x is in L without the verifier learning anything other than that $x \in L$. To demonstrate that the verifier does not learn too much a polynomial time simulator is given that (without having a witness w for the statement $x \in L$) can output a view indistinguishable from the verifier's view in his interaction with an honest prover. Since the verifier can just run the simulator himself, this guarantees that he did not learn anything additional from the proof. We focus on the restricted case of *black-box* zero-knowledge where the simulator only accesses the cheating verifier as a black-box while generating the simulated transcript. This definition of zero-knowledge was introduced in the works of Goldreich et al. [62, 58] and we follow the presentation of [55].

Definition 2.3.8 *Fix a language* $L \in$ NP *and a corresponding* NP *relation* R_L. *For* $n \in \mathbb{N}$, *let* $L_n \stackrel{\text{def}}{=} L \cap \{0,1\}^n$ *and* $R_n \stackrel{\text{def}}{=} \{(x,w) \mid (x,w) \in R_L \text{ and } x \in L_n\}$. *An* efficient prover interactive proof system *for* L *is a pair of* ppt *interactive algorithms* (P,V), *where only* V *has output, such that the following two conditions hold:*

- *(Perfect Completeness:) For every* $(x,w) \in R_L$,

$$\Pr[\langle P(w), V \rangle(x) = 1] = 1$$

- *(Negligible Soundness:) For every* $x \notin L$ *and any (possibly unbounded) adversarial prover* P^* *there exists a negligible function* negl *such that,*

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

We say that (P,V) *is* black-box zero-knowledge (BBZK) *if additionally the following holds:*

- *(Black-Box Zero-Knowledge:) There exists an expected polynomial time simulator* S *such that for any non-uniform polynomial time cheating verifier* V^* *the following two ensembles are indistinguishable by non-uniform polynomial time distinguishers.*

  - $\{\langle P(w), V^* \rangle(x)\}_{(x,w) \in R_L}$ *(the output of* V^* *after interacting with the honest prover on witness* w *for the fact that* $x \in L$*)*

  - $\{S^{V^*}(x)\}_{(x,w) \in R_L}$ *(the simulated output of* V^* *produced by the simulator* S *using black-box access to* V^**)*

Some discussion is in order. First, without loss of generality, we assume that the cheating verifier V^* outputs its entire view, so simulating its output is equivalent to simulating its view. By making both the verifier and distinguisher non-uniform, we also require zero-knowledge to hold with respect to verifiers (and distinguishers) receiving auxiliary input. This is necessary for zero-knowledge to be preserved under sequential composition (see [55] for a discussion).

Our definition differs from the standard one in the following ways. First, we only define zero-knowledge proofs for the case that $L \in \mathsf{NP}$. This allows us to make the honest prover P efficient when given an NP witness w. Second, we require that zero-knowledge proofs have perfect completeness and negligible soundness error. Finally, we require that the simulation be black-box. That is, we require a universal simulator S that is able to simulate the view of any cheating verifier V^* while only making black-box queries to V^*. Note that this notion of black-box simulation is quite different from the notion of black-box constructions discussed in this thesis. In black-box simulation, black-box refers to the access that the simulator has to the cheating verifier, whereas in a black-box construction, black-box access refers to the way the construction uses the underlying primitive.

Chapter 3

Black-Box Constructions and Separations

In this chapter, we introduce the concept of a *black-box* construction of a primitive Q from a primitive P. In Section 3.1, we review the definitions of several different types of black-box constructions and the relationships between them. Then, in Section 3.2, we define the concept of a *black-box separation* of Q from P and give a survey of the techniques and results proving such separations for various primitives.

3.1 Definitions of Black-Box Constructions

Intuitively, a construction of primitive Q from primitive P is *black-box* if it treats P as an oracle, only looking at the input/output behavior of P and not at how P is implemented. To make the above intuition into a formal definition we first need to define what a primitive is and exactly what it means to construct one primitive from another. In this section, we provide the necessary language for doing this. Following the definitions of Reingold et al. [108], we define several types of black-box constructions and discuss the relationships between them. We note that all the definitions in this section are presented for the case of uniform adversaries and do not necessarily apply to non-uniform or information theoretic notions of security.

3.1.1 Cryptographic Primitives

A cryptographic primitive consists of two components: a correctness requirement specifying what the primitive should do and a security requirement specifying what it means for an attacker to break the primitive's security. A secure implementation of the primitive should then satisfy the correctness requirement and also be secure against any polynomial time attacker. That is, no probabilistic polynomial time attacker should be able to break any of the security requirements. More formally,

Definition 3.1.1 *A primitive P is a pair (F_P, R_P), where F_P is a set of functions $f : \{0,1\}^* \to \{0,1\}^*$ and R_P is a relation over pairs (f, M) where $f \in R_P$ and M is a (possibly inefficient) Turing machine.*

- *We say that a function f* implements *P if $f \in F_P$. Additionally, we say that f* efficiently implements *P if $f \in F_P$ and f is computable by a* ppt *machine.*

- *A machine M P-*breaks *implementation $f \in F_P$ if the pair $(f, M) \in R_P$. Thus, a* secure implementation *of P is a function $f \in F_P$ such that no* ppt *machine P-breaks f.*

- *A primitive P* exists *if there exists an efficient and secure implementation f of P.*

The set F_P in the above definition is used to capture the correctness requirements. That is, any function $f \in F_P$ will have the correct input and output space and will have the proper behavior. For example, in the case when P is a length-preserving one-way function, any such f must map inputs of length n to outputs of length n. However, this does not say anything about the security of f. This is captured by the set R_P, which, for each $f \in F_P$, contains all the machines that will break the security of f as

an implementation of P. By carefully defining which machines M are in this set for a function f, we can capture the desired security property. For example, for the case of one-way functions we would define $(f, M) \in R_P$ if there is a polynomial p such that $\Pr_{x \leftarrow \{0,1\}^n}[M(f(x)) \in f^{-1}(f(x))] > 1/p(n)$ for infinitely many values of n.

We will often want to argue that some primitive P can be securely instantiated using an oracle $\mathcal{O}$. That is, there exists an efficient implementation of P using $\mathcal{O}$ such that no efficient adversary with oracle access to $\mathcal{O}$ can break its security. More formally,

Definition 3.1.2 *For an oracle $\mathcal{O} : \{0,1\}^* \rightarrow \{0,1\}^*$, we say that:*

- *$\mathcal{O}$ implements* primitive P *if there exists an implementation $f \in F_P$ that is computable by a* ppt *machine with oracle access to $\mathcal{O}$.*

- *Implementation f is* secure *relative to $\mathcal{O}$ if there is no* ppt *oracle machine M such that $M^{\mathcal{O}}$ P-breaks f.*

- *A primitive P exists relative to $\mathcal{O}$ if $\mathcal{O}$ implements P via implementation f which is secure relative to $\mathcal{O}$.*

3.1.2 Cryptographic Constructions

Now that we know what a cryptographic primitive is, we can define what it means to construct primitive Q from primitive P. A cryptographic construction consists of two algorithms, a *construction G* turning an instance of P into an instance of Q and a *security reduction S* showing that if we can break the construction of Q then we can also break the underlying instantiation of P. Here, we will only consider black-box constructions. That is, the construction of Q will treat P as an oracle. Following [108], we define several

variants of such constructions that vary in how the adversary breaking Q is used by the security reduction. We only give definitions relevant to this work. For additional definitions and discussion we refer the reader to [108].

We begin with the definition of a fully black-box construction of Q from P. In a *fully* black-box construction it is additionally required that the security reduction use the adversary breaking the security of Q as a black-box. More formally,

Definition 3.1.3 *There exists a* fully black-box *construction of primitive* $Q = (F_Q, R_Q)$ *from primitive* $P = (F_P, R_P)$, *if there exist* ppt *oracle machines G and S such that:*

- *For every implementation $f \in F_P$, $G^f \in F_Q$ (G^f implements Q)*

- *For every implementation $f \in F_P$ and every (possibly inefficient) machine M, if M Q-breaks G^f then $S^{M,f}$ P-breaks f.*

This construction consists of two components: the *construction G* and the *security reduction S*. As in all black-box constructions, we require that G use the primitive P as a black-box. In fact we require that a universal G work for all f. What makes this construction fully black-box is that the security reduction uses the (possibly inefficient) adversary M breaking the security of Q in a black-box way to break the security of P. In particular, S must work for any such adversary, even an inefficient one. Interestingly, the vast majority of known constructions in cryptography satisfy this very strong requirement.

Next, we define a less restricted type of construction called a *semi black-box* construction. In such a construction the security reduction S no longer uses the Q-adversary as a black-box and may be different for each M. However, since we now require that M

be polynomial-time it may no longer be able to evaluate the possibly inefficient implementation f. Thus, we give M access to an oracle evaluating f. More formally,

Definition 3.1.4 *There exists a* semi black-box construction *of primitive $Q = (F_Q, R_Q)$ from primitive $P = (F_P, R_P)$ if there exists a* ppt *oracle machine G such that:*

- *For every implementation $f \in F_P$, $G^f \in F_Q$ (G^f implements Q)*

- *For every implementation $f \in F_P$, if there exists a* ppt *oracle machine M such that M^f Q-breaks G^f, then there exists a* ppt *oracle machine S such that S^f P-breaks f.*

The key difference between a semi black-box and a fully black-box construction is the fact that the security reduction is defined based on the Q-adversary M. In particular, this means that a different reduction, dependent on the code of M, can be used for each M. However, this security reduction still has a black-box component because of M's oracle access to the possibly inefficient implementation f. The security reduction may not know the exact queries that M makes to f and must work regardless of what the answers to these queries are. For a complete discussion and for a definition of a black-box construction with a truly non-black-box security reduction (a *weakly black-box* construction) we refer the reader to [108].

We can actually relax the definition a little more. This time, we leave the security reduction alone and instead focus on the construction G. In the previous two definitions, we required a universal construction G that worked for every f. However, it is sufficient that, for every implementation f, there exists a construction G. We still require that such a construction be black-box in that it only accesses f as a black-box. More formally,

Definition 3.1.5 *There exists a $\forall\exists$semi black-box construction of primitive $Q = (F_Q, R_Q)$ from primitive $P = (F_P, R_P)$ if, for every implementation $f \in F_P$, there exists a* ppt *oracle machine G such that:*

- *$G^f \in F_Q$ (G^f implements Q)*

- *If there exists a* ppt *oracle machine M such that M^f Q-breaks G^f, then there exists a* ppt *oracle machine S such that S^f P-breaks f.*

We also define a related notion from complexity theory of a relativizing construction [4]. These are constructions that remain secure relative to any oracle. It turns out that relativizing constructions are very useful for reasoning about black-box constructions. Formally,

Definition 3.1.6 *There exists a relativizing construction of primitive $Q = (F_Q, R_Q)$ from primitive $P = (F_P, R_P)$ if for any oracle $\mathcal{O} : \{0, 1\}^* \to \{0, 1\}^*$, if P exists relative to $\mathcal{O}$ then so does Q.*

We conclude with the following lemma due to [108] demonstrating some simple relationships between the above definitions. These relationships are summarized in Figure 3.1. The proof of this lemma follows easily from the definitions and is omitted.

Lemma 3.1.7 *For any two primitives P and Q, we have the following:*

1. *If there exists a fully black-box construction of Q from P, then there exists a semi black-box construction of Q from P and a relativizing construction of Q from P.*

2. *If there exists either a semi black-box or a relativizing construction of Q from P, then there exists a $\forall\exists$semi black-box construction of Q from P.*
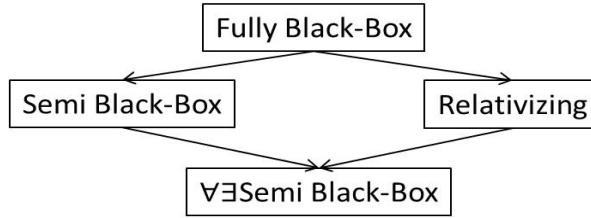
Figure 3.1: Relationships Between Black-Box Constructions

3.2 Black-Box Separation Techniques and Results

As mentioned earlier, an important goal in cryptography is to construct "high level" primitives from "low level" ones. This line of research has been very successful, and we have many results showing constructions of cryptographic primitives from each other. However, other constructions, such as the construction of public-key encryption from one-way functions, have been more elusive. This phenomenon has caused people to ask whether such constructions are inherently impossible. Unfortunately, as noted earlier, it is impossible to rule out such a construction as long as we believe that public-key encryption exists. Thus, a general impossibility result ruling out all constructions of public-key encryption from one-way functions is not achievable.

Instead, Impagliazzo and Rudich [76] suggested looking at the restricted class of black-box constructions discussed in the previous section. Since the vast majority of cryptographic constructions are black-box, this rules out most known approaches. Under this restriction, they were able to show the first known *black-box separation* between two primitives. They did this by reasoning about relativizing constructions. Specifically, they showed the following theorem.

Theorem 3.2.1 ([76]) *There is no relativizing construction of secure key agreement from one-*

way functions.

We note that Impagliazzo and Rudich [76] actually prove the stronger result that there is no relativizing construction of key agreement with noticeable (rather than perfect) completeness from one-way permutations. Since fully black-box constructions imply relativizing constructions this also implies a separation under fully black-box constructions. We introduce some language to talk about such black-box separations.

Definition 3.2.2 *We say that there is a* fully black-box separation *of primitive Q from primitive P if there does not exist a fully black-box construction of Q from P.*

Analogously, we can define separations under all the types of black-box constructions discussed in Section 3.1. Thus, Theorem 3.2.1 shows that there is a *relativizing separation* and a *fully black-box separation* of key agreement from one-way functions.

This result initiated a very active and successful field of study demonstrating black-box separations between various cryptographic primitives showing that, at least for the case of black-box constructions, the world of cryptographic primitives is quite complex. We now review some of the most common techniques used to prove these separations and the various results that they have been used for. These reviews are meant to give a sketch of the techniques and results and we refer the reader to the cited works for the complete details.

3.2.1 One-Oracle Techniques

A Separating Oracle:

This is the original technique suggested by Impagliazzo and Rudich [76] and to this day

remains the most commonly used technique for proving black-box separations. This technique proceeds by proving that there is no relativizing construction of primitive Q from primitive P. This is done by demonstrating an oracle $\mathcal{O}$ such that P exists relative to $\mathcal{O}$, but no construction $G^{\mathcal{O}}$ of primitive Q is secure. We call $\mathcal{O}$ a *separating oracle*. More formally, these separations rely on the following fact.

Fact 3.2.3 *To show that there is no relativizing construction from primitive P to primitive Q, it suffices to show a separating oracle $\mathcal{O}$ such that:*

- *P exists relative to $\mathcal{O}$.*

- *There exists a* ppt *oracle machine M, such that $M^{\mathcal{O}}$ Q-breaks any* ppt *oracle construction $G^{\mathcal{O}}$.*

Impagliazzo and Rudich [76] prove their separation as follows. First, they use the Borel-Cantelli lemma to show that a random oracle $\mathcal{O}$ is one-way with probability 1 (over the choice of $\mathcal{O}$). This holds even against an unbounded adversary making at most polynomially many queries to $\mathcal{O}$. Next, they show an (inefficient) adversary Eve who breaks the security of any construction, $(A^{\mathcal{O}}, B^{\mathcal{O}})$, of key agreement while making $O(n^6)$ queries to a random $\mathcal{O}$, where the probability that Eve succeeds is again over the choice of $\mathcal{O}$. Roughly, this Eve works by finding all queries made to $\mathcal{O}$ by both A and B (see [76] for a discussion of why this is enough). Since Eve only makes polynomially many queries to $\mathcal{O}$, this is sufficient to rule out a fully-black-box construction, as Eve cannot be used to invert a random $\mathcal{O}$. In fact, this weaker result is the stopping point of most follow up work. (We discuss this approach in more detail in Section 3.2.5.) To achieve a relativizing separation, Impagliazzo and Rudich next show that Eve is efficient if P = NP. Then they

apply the Borel-Cantelli lemma again to argue that there is a fixed oracle $\hat{\mathcal{O}}$ relative to which Eve breaks any construction of key-agreement, but $\hat{\mathcal{O}}$ is one-way against any adversary asking polynomially many queries. Finally, using the fact that P = NP relative to a PSPACE oracle [4], the joint oracle $(\hat{\mathcal{O}}, \text{PSPACE})$ gives the necessary separating oracle.

Note, that this also suffices to prove the stronger statement that, if P = NP there is no ∀∃semi black-box construction of KA from OWF's, since in this case Eve is efficient as required and the additional PSPACE oracle is not necessary. This requirement that P = NP was subsequently removed by Reingold et al. [108] using an "embedding technique" originally due to Simon [117]. This technique embeds both $\hat{\mathcal{O}}$ and the PSPACE oracle into a single oracle while preserving the one-wayness of $\hat{\mathcal{O}}$. Relative to this oracle, one-way functions exist but there is an efficient adversary that breaks the security of key agreement. Thus, combined they prove,

Theorem 3.2.4 ([76] and [108]) *There is no ∀∃semi black-box construction of secure key agreement from one-way functions.*

This result was recently improved by Barak and Mahmoody-Ghidary [10], who showed a more efficient adversary demonstrating a fully black-box separation of key agreement from one-way functions (and even one-way permutations). Namely, they show an attacker Eve that breaks any construction of key agreement for a random oracle $\mathcal{O}$ while making only $O(n^2)$ queries to the oracle. The improved attacker and analysis have since been used by several works [34, 82] to separate fair coin tossing and blind signatures from one-way functions.

Many other works have used this technique to prove black-box separation results. We review some of them here. We do not define all of the discussed primitives and refer the reader to the cited works for the necessary definitions. In the setting of secret-key cryptography, Rudich [112], Kahn et al. [77] and Chang et al. [31] showed that one-way permutations can not be constructed from a variety of primitives. Additionally, Simon [117] showed a separation of collision resistant hash-functions from one-way functions and Fischlin [43] showed a separation of non-interactive statistically-hiding commitments from one-way permutations and even one-to-one trapdoor functions.

In the setting of public-key cryptography, Rudich [113] showed a separation between k-round and $(k + 1)$-round key agreement and his techniques were extended by Gertner et al. [52] to show separations between key agreement, CPA-secure public-key encryption and oblivious transfer. A number of works have also looked at the possibility of constructing CCA-secure encryption. Specifically, Gertner et al. [53] showed a partial separation of CCA-secure encryption from CPA-secure encryption. Addressing specific techniques for constructing CCA-secure encryption, Vahlis [119] showed a separation of trapdoor functions secure under correlated inputs from trapdoor permutations and Kiltz et al. [84] showed a separation of correlation secure trapdoor functions from adaptive trapdoor functions. Additionally, Boneh et al. [22] showed a separation of identity-based encryption from trapdoor permutations.

All of the above results argue about the feasibility of a certain construction. A somewhat different line of work, also using the same technique, has looked instead at bounding the efficiency of black-box constructions. Specifically, Kim et al. [85] and Barak and Mahmoody-Ghidari [9] use this technique to prove lower bounds on the efficiency

of black-box constructions of universal one-way hash functions and digital signature schemes from one-way functions.

An Oracle For Each Construction:

A twist on this technique was proposed by Brakerski et al. [25]. Rather than showing a single separating oracle, the authors give a different oracle for each potential construction. That is, for each construction G they show an oracle $\mathcal{O}$ relative to which primitive P exists, but $G^{\mathcal{O}}$ is not a secure implementation of Q. More formally, their result is summarized by the following,

Fact 3.2.5 *To show that there is no semi black-box construction from primitive P to primitive Q, it suffices to show that for any* ppt *construction G, there exists an oracle $\mathcal{O}$ such that:*

- *P exists relative to $\mathcal{O}$.*

- *There is a* ppt *oracle machine M such that $M^{\mathcal{O}}$ breaks the Q-security of $G^{\mathcal{O}}$.*

The authors use this technique to show that there is no construction of weak verifiable random functions from one-way permutations. Note that unlike the separating oracle technique, this only rules out semi black-box rather than relativizing (and $\forall\exists$semi black-box) constructions as it can only prove that for each construction there is some oracle for which it fails rather than an oracle for which all constructions fail.

3.2.2 Two-Oracle Techniques

A Breaker Oracle:

A different technique for black-box separations was first formalized by Hsiao and

Reyzin [75] who used it to separate public-coin collision-resistant hash functions from private-coin ones. To separate primitive Q from primitive P, this technique makes use of two oracles: a "helper" oracle A to guarantee P-security and a "breaker" oracle B to break the Q-security of any construction using A. More formally, the separation is captured by the following fact,

Fact 3.2.6 *To show that there is no fully black-box construction of primitive Q from primitive P, it suffice to show two oracles* A *and* B *such that,*

- *There is an* ppt *oracle machine L such that L^A implements P.*

- *For any* ppt *oracle machine G, if G^A implements Q then there exists a* ppt *adversary M such that $M^{A,B}$ breaks the Q-security of G^A.*

- *There is no* ppt *oracle machine S such that $S^{A,B}$ breaks the P-security of L^A.*

This big difference between this technique and the one-oracle techniques is that the construction G is not given access to the breaker oracle B. In fact, if G were given both A and B then this would indeed become a one-oracle separation with oracle (A, B). However, not giving B to G allows B to be defined dependent on the oracles used by the construction (in this case just A) without having to worry about any "self-referencing". Specifically, much care must be taken to design an oracle B that can break constructions that may use B (as is done in [117]) and this can be avoided by using this technique. A major drawback of this approach is that it only rules out fully black-box constructions. This is due to the fact that the adversary $M^{A,B}$ may not be efficient when only given access to A.

This approach has also seen a fair amount of use and we now review some of the results. The most well known of these are the works of Haitner et al. [68, 69] which, building on the work of Wee [120], show lower bounds on the round complexity of statistically hiding commitments and the communication complexity of private information retrieval protocols based on trapdoor permutations. These works define a breaker oracle Sam which finds collisions in interactive protocols. This oracle has since been used by a number of works (e.g. [111, 65, 105, 71, 104]) and will be discussed in more detail in Chapter 5. Some other results using this approach include Boldyreva et al. [19] who rule out constructions of non-malleable hash-functions from one-way permutations and Matsuda et al. [90] who rule out constructions of one-way permutations from injective length-increasing one-way functions.

This technique was also used by Hofheinz [74] and Haitner and Holenstein [70] to demonstrate *strongly*-black-box separations of commitment schemes secure under selective opening (SO-COM) and key-dependent message (KDM) secure encryption schemes from *any* cryptographic assumption. What makes these separations strongly-black-box is that they only rule out constructions that, besides being black-box in the underlying primitive, are also black-box in their use of some additional function given as part of the definition of the primitive. Specifically, in the case of SO-COM [74], it is required that the construction be black-box in the underlying message distribution, while in the case of KDM-secure encryption [70], it is required that the construction treat the KDM query function as a black-box.

A Construction Dependent Breaker Oracle:

A somewhat different two-oracle technique was introduced by Gertner et al. [54] to sep-

arate trapdoor functions from public-key encryption. Rather than define a universal "breaker" oracle as in [75], they allow the breaker oracle to depend on the construction. Formally, their separation is captured by the following.

Fact 3.2.7 *To show that there is no fully black-box construction of primitive Q from primitive P, it suffices to show that there exists an oracle* A *such that:*

- *There is an* ppt *oracle machine L such that L^A implements P.*

- *For any* ppt *oracle machine G, if G^A implements Q then there exists an oracle* B *such that:*

  - *There exists a* ppt *adversary M such that $M^{A,B}$ breaks the Q-security of G^A.*

  - *There is no* ppt *oracle machine S such that $S^{A,B}$ breaks the P-security of L^A.*

Just as in the previous two-oracle technique, this approach only rules out fully black-box constructions. The major advantage of this technique is that a different breaker oracle can be used for every construction and thus can be tailor-made to break a specific construction rather than giving a general attack against all constructions.

3.2.3 Simulation Based Techniques

Simulating the Helper Oracle:

We now discuss a significantly different technique for showing black-box separations originally proposed by Gennaro and Trevisan [50]. This technique was further refined by Gennaro et al. [48] and our presentation follows the merged version of these two works [49]. The main result of these works is to give lower bounds on the efficiency of black-box constructions of various cryptographic primitives (e.g. pseudorandom gener-

ators, universal one-way hash functions, encryption and signatures schemes) from one-way and trapdoor permutations. Specifically, they prove that if a black-box construction of one of these primitives makes "few" queries to the one-way function then there exists an unconditional instantiation of the primitive in question. Since the unconditional existence of any of these primitives would imply that $P \neq NP$, this is viewed as strong evidence that such a construction will be very hard to find. Note that, unlike the separations discussed previously, this does not prove that such a construction does not exist, only that it will be hard to find. Our presentation here focuses on the result lower-bounding the efficiency of weakly black-box constructions of pseudorandom generators (PRGs) from one-way permutations. For the other results we refer the reader to [49].

The first step, as before, is to prove that a random permutation is one-way. Here, a very different technique called *the reconstruction lemma* is used to show that a random permutation on n-bit strings is, with high probability, one-way even against non-uniform adversaries of size $2^{\Omega(n)}$. This reconstruction lemma has since been used in a number of works (e.g. [120, 68, 69]) and we will elaborate on it a little later.

Using this result, the intuition behind the proof for the case of PRGs is as follows. Let $S = 2^{n/5}$ (the constant here is arbitrary) and let $G^{\mathcal{O}}$ be a secure construction of a PRG using oracle access to a one-way permutation $\mathcal{O}$. We let $\mathcal{O}$ be a permutation that is a random permutation on its first $t = \Theta(\log S)$ bits and is the identity function on the remaining $n - t$ bits. From the above result about random permutations, it follows that $\mathcal{O}$ is one-way against non-uniform adversaries of size S, and hence G must be secure against any such adversary when instantiated with $\mathcal{O}$. Let q be the number of queries that G makes to $\mathcal{O}$. Notice that the answer to every $\mathcal{O}$ query can be exactly described

by t random bits as we only need to know the output on the first t bits of the query (the remaining bits of the output are the same as the corresponding bits in the query). Thus, it is possible to answer all of G's queries using at most $q \cdot t$ random bits. Therefore, G can be converted into a PRG G' that does not use $\mathcal{O}$. Instead G' uses the first $q \cdot t$ bits of its random seed to "simulate" the oracle $\mathcal{O}$ for G. If q is sufficiently small and the expansion factor of G is sufficiently large, the output of G' is still longer than its seed making G' an unconditional PRG secure against (non-uniform) adversaries of size S. Since the existence of such a PRG implies that $\mathsf{P} \neq \mathsf{NP}$ such a black-box construction is likely to be hard to find. Note also that this proof technique does not use the security reduction at all and thus only requires that the construction be black-box without any restriction on how the security reduction accesses the adversary. Thus, this technique can rule out even weakly black-box constructions.

Meta-Reductions: Simulating the Adversary

A somewhat similar approach for demonstrating fully black-box separations between two primitives is the technique of using *meta-reductions*. The main idea of this approach is to build "a reduction against the reduction". That is, we show that if there exists a security reduction from the security of a primitive Q to a primitive P then we can use this reduction to break P-security. In a little more detail, we start by assuming that primitive P exists. Now assume that a black-box security reduction S exists from the security of primitive Q to the security of primitive P. That is, given any adversary $\mathcal{A}$ that Q-breaks an instantiation of Q, $S^{\mathcal{A}}$ P-breaks the underlying instantiation of P. Now, for any such S, we construct a meta-reduction S' that "simulates" the adversary $\mathcal{A}$ to the real reduction S. Specifically, S' runs S while simulating the answers to any queries S makes to $\mathcal{A}$.

As long as S can not distinguish this simulated adversary from the real one S' yields a ppt procedure for breaking the security of P without any oracles. However, the existence of such a procedure contradicts the assumption that P is secure. The black-box separation is then proven by showing a meta-reduction that works for any security reduction S. Meta-reductions have two important properties that distinguish them from the techniques described earlier. First, meta-reductions do not depend on the construction of Q from P but only on the security reduction. Thus, they can rule out the existence of any construction as long as the security reduction is black-box. Second, meta-reductions allow the assumed primitive P to be arbitrary making it possible to prove that Q can not be securely instantiated based on any cryptographic assumption.

This technique was originally proposed by Boneh and Venkatesan [23] in the context of algebraic reductions from factoring to low-exponent RSA. Since then meta-reductions have received a fair amount of use. Some examples of problems studied using this technique include the feasibility of instantiating efficient signature schemes in the standard model [32, 36, 35, 96, 47], the possibility of CCA-secure encryption based on factoring [97], relations between one-more style cryptographic assumptions [28, 29] and the feasibility of three-move blind signature schemes [45].

3.2.4 Security of the Base Primitive

All but one of the approaches mentioned above rely on building an oracle such that primitive P exists relative to that oracle. We now review the techniques for finding such an oracle.

Using The Borel-Cantelli Lemma:

This technique was introduced by Impagliazzo and Rudich [76] for the case of uniform adversaries. We describe it here for the case of proving that a random oracle $\mathcal{O}$ is one-way.

Recall that a random oracle $\mathcal{O} = \{\mathcal{O}_n\}_{n \in \mathbb{N}}$ is a collection of oracles where each $\mathcal{O}_n$ is chosen uniformly from the space of functions from $\{0,1\}^n$ to $\{0,1\}^n$. The first step is to show that for any fixed ppt oracle adversary $\mathcal{A}$ there exists a polynomial poly such that for any $n \in \mathbb{N}$ and any $x \in \{0,1\}^n$ we have that

$$\Pr_{\mathcal{O}} \left[\mathcal{A}^{\mathcal{O}}(\mathcal{O}(x)) \in \mathcal{O}^{-1}(\mathcal{O}(x)) \right] \leq \frac{\mathsf{poly}(n)}{2^n} \quad . \tag{3.1}$$

where the above probability is taken over the choice of $\mathcal{O}$ and $\mathsf{poly}(n)$ is a bound on the running time of $\mathcal{A}$. This is proven by a simple lazy-sampling argument where we view any point of $\mathcal{O}$ that has not been queried by $\mathcal{A}$ as unfixed. Thus, the only way to invert $\mathcal{O}$ is to query it on a point that returns $y = \mathcal{O}(x)$. However, on each query asked the probability that it returns y is exactly $1/2^n$ giving the above.

Next, for any $n \in \mathbb{N}$, any fixed ppt adversary $\mathcal{A}$ running in time $\mathsf{poly}(n)$ and any fixed oracle $\mathcal{O}$, let $E_{n,\mathcal{A},\mathcal{O}}$ denote the event that $\mathcal{O}$ is such that

$$\Pr_{x \leftarrow \{0,1\}^n} \left[\mathcal{A}^{\mathcal{O}}(\mathcal{O}(x)) \in \mathcal{O}^{-1}(\mathcal{O}(x)) \right] > \frac{n^2 \cdot \mathsf{poly}(n)}{2^n} \quad .$$

By Equation 3.1, we have that the expected probability (over the choice of $\mathcal{O}$) that $\mathcal{A}$ inverts $\mathcal{O}(x)$ is at most $\frac{\mathsf{poly}(n)}{2^n}$. Thus, using Markov's inequality, we get that for any fixed $\mathcal{A}$ running in time $\mathsf{poly}(n)$

$$\Pr_{\mathcal{O}}[E_{n,\mathcal{A},\mathcal{O}}] \leq \frac{1}{n^2} \quad .$$

Then, since $\sum_{n=1}^{\infty} \frac{1}{n^2} < \infty$, the Borel-Cantelli lemma implies that the probability over the choice of $\mathcal{O}$ that $E_{n,\mathcal{A},\mathcal{O}}$ occurs for infinitely many n is zero. Thus, for measure 1 of oracles $\mathcal{O}$ we have that for any $\mathcal{A}$ there exists a negligible function negl such that

$$\Pr_{x \leftarrow \{0,1\}^n} \left[\mathcal{A}^{\mathcal{O}}(\mathcal{O}(x)) \in \mathcal{O}^{-1}(\mathcal{O}(x)) \right] \leq \frac{n^2 \cdot \mathsf{poly}(n)}{2^n} < \mathsf{negl}(n) \ .$$

Thus, by removing a set of measure 0 oracles for each of the (countably many) machines $\mathcal{A}$, we get that for measure 1 of random oracles $\mathcal{O}$ it holds that for *all* ppt adversaries $\mathcal{A}$, the probability that $\mathcal{A}$ inverts $\mathcal{O}$ is negligible. To summarize, we have proven the following theorem.

Theorem 3.2.8 *With probability 1 over the choice of $\mathcal{O}$, $\mathcal{O}$ is one-way against all* ppt *adversaries.*

In fact, since the above proof relativizes and thus must also hold relative to a PSPACE oracle, we get that this is true even for computationally unbounded adversaries as long as they only make polynomially many oracle queries to $\mathcal{O}$.

The Reconstruction Lemma:

An alternative approach for proving that a base primitive P exists was given by Gennaro et al. [49]. Specifically, they show that, with overwhelming probability, a random permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ is one-way against any non-uniform adversary of size $S = 2^{n/5}$ for sufficiently large n. A major benefit of this approach is that we can get a concrete bound on the probability that a random permutation on n bits is one way. Additionally, this allows one to prove that π is one-way even against non-uniform adversaries. This result is summarized by the following theorem.

Theorem 3.2.9 *For all sufficiently large n, a random $\pi \leftarrow \Pi_n$ is $2^{n/5}$-hard with probability at least $1 - 2^{-2^{n/2}}$.*

Here Π_n is the set of all permutations over n bits and a permutation π is S-hard if for any circuit $\mathcal{A}$ of size less than or equal to S, $\Pr_{y \leftarrow \{0,1\}^n}[\mathcal{A}^\pi(y) = \pi^{-1}(y)] \leq \frac{1}{S}$.

This theorem is proven using the *reconstruction lemma*. This is an argument that shows that if there exists a small circuit $\mathcal{A}$ that can invert π with high probability, then $\mathcal{A}$ can be used to give a short description of the random permutation π. Since there are many permutations on $\{0,1\}^n$ this leads to a contradiction. More formally, the reconstruction lemma says,

Lemma 3.2.10 (Reconstruction Lemma) *Let $\mathcal{A}$ be a circuit that makes q queries to a permutation $\pi : \{0,1\}^n \rightarrow \{0,1\}^n$, and for which $\Pr_y[\mathcal{A}^\pi(y) = \pi^{-1}(y)] \geq \epsilon$. Then π can be described using at most*

$$2 \log \binom{2^n}{a} + \log\left((2^n - a)!\right)$$

bits (given $\mathcal{A}$), where $a = \frac{\epsilon 2^n}{q+1}$.

We now sketch the proof of Theorem 3.2.9. Here, we only provide a high-level outline of the proof and refer the reader to [49] for the details. Let $\mathcal{A}$ be a circuit of size $S = 2^{n/5}$. Clearly, $\mathcal{A}$ makes at most $q = 2^{n/5}$ queries to π. It is easy to show that any such circuit has a (relatively) short description. Thus, by Lemma 3.2.10, any permutation π that can be inverted by $\mathcal{A}$ must also have a short description. However, since there are very many permutations on n-bits, only a tiny fraction of them can have such a short description. Thus, the probability over the choice of $\pi \leftarrow \Pi_n$ that $\mathcal{A}$ succeeds in inverting

π must be very small. Finally, taking a union bound over all the possible circuits of size S, we get that the probability (over the choice of $\pi \leftarrow \Pi_n$) that there exists a circuit $\mathcal{A}$ of size at most S that inverts π is also very small, proving the theorem.

3.2.5 On The Existence of a Separating Oracle

In many of the techniques described in the previous section we talk about a fixed separating oracle such that primitive P exists relative to this oracle. For the case of one-way functions or permutations the existence of such a fixed oracle is, in fact, implied by either of the results from Section 3.2.4. For example, consider the case of one-way permutations. Theorem 3.2.9 shows that for any polynomial poly there exists a negligible function negl such that

$$\Pr_{\pi \leftarrow \Pi_n} \left[\exists \mathcal{A} \text{ s.t. } \Pr_{y \leftarrow \{0,1\}^n} [\mathcal{A}^\pi(y) = \pi^{-1}(y)] \geq 1/\mathsf{poly}(n) \right] < \mathsf{negl}(n) \tag{3.2}$$

where $\mathcal{A}$ is of size $\mathsf{poly}(n)$.

This statement implies the existence of an oracle $\hat{\Pi} \overset{\text{def}}{=} \{\hat{\pi}_n\}_{n \in \mathbb{N}}$ such that $\hat{\Pi}$ is a one-way permutation. Such an oracle can be constructed by taking, for each n, some permutation $\hat{\pi}_n$ that satisfies the above equation. The existence of such an oracle is clearly necessary if one wants to prove a relativizing separation and it also gives a fixed oracle that can be used in further constructions. However, if the goal is only to prove a fully black-box separation, it is not necessary for such a fixed oracle to exist. A technique formalized by Barak and Mahmoody-Ghidari [9] (although already used many times before this) shows that it is sufficient to prove that a *random* permutation oracle is one-

way rather than that any fixed permutation oracle is one-way. That is, it is enough to prove that for any polynomial poly there exists a negligible function negl such that for any $\mathcal{A}$ of size $\text{poly}(n)$

$$\Pr_{\pi \leftarrow \Pi_n, y \leftarrow \{0,1\}^n}[\mathcal{A}^\pi(y) = \pi^{-1}(y)] \leq \text{negl}(n) \tag{3.3}$$

We now demonstrate why the above is sufficient to prove a fully black-box separation. Consider the case of a fully black-box construction of a key-agreement protocol from one-way permutation. To simplify presentation we assume that, on input 1^n, the construction only queries the one-way permutation on inputs of length n. Thus, to analyze the probability that Eve succeeds on security parameter n, it is enough to consider the probability space over the choice of permutation π_n. We omit the subscript n for the rest of this discussion.

To prove a separation, we need to show an adversary E that breaks the security of any construction $G^\pi = (A^\pi, B^\pi)$ for a random permutation oracle π. Let $T \stackrel{\text{def}}{=} \langle A^\pi(r_a), B^\pi(r_b)\rangle(1^n)$ be the transcript of an execution of G^π where A has randomness r_a and B has randomness r_b and let k be the output key. Then for any adversary E breaking the security of G^π, there exists a polynomial poly_1 such that

$$\Pr_{\pi \leftarrow \Pi_n, r_a \leftarrow \{0,1\}^n, r_b \leftarrow \{0,1\}^n}[E^\pi(T) = k] > 1/\text{poly}_1(n)$$

By an averaging argument, we get that

$$\Pr_{\pi \leftarrow \Pi_n}\left[\Pr_{r_a, r_b \leftarrow \{0,1\}^n}[E^\pi(T) = k] > \frac{1}{2\mathsf{poly}_1(n)}\right] > \frac{1}{2\mathsf{poly}_1(n)}$$

Now, for a fixed adversary E, we say that an oracle π is "bad" with respect to E (denoted by BAD_E) if $\Pr[E^\pi(T) = k] > \frac{1}{2\mathsf{poly}_1(n)}$. For any such oracle, the assumed fully black-box construction guarantees the existence of a polynomial size security reduction S such that $S^{E,\pi}$ inverts π. That is, there exists a polynomial poly_2 such that,

$$\Pr_{y \leftarrow \{0,1\}^n}[S^{E,\pi}(y) = \pi^{-1}(y)] > 1/\mathsf{poly}_2(n)$$

Note that S makes polynomially many queries to E and each of these can be simulated using at most polynomially many queries to π (since E is polynomial size). Thus, there is a polynomial size adversary $\hat{S}$ that runs S simulating the answers to all of S's queries to E such that $\hat{S}$ inverts any π that is BAD_E. However, since $\Pr_{\pi \leftarrow \Pi_n}[\pi$ is $\mathsf{BAD}_E] \geq \frac{1}{2\mathsf{poly}_1(n)}$, we get that there is a polynomial poly such that $\hat{S}$ is of size at most $\mathsf{poly}(n)$ and

$$\Pr_{\pi \leftarrow \Pi_n, y \leftarrow \{0,1\}^n}\left[\hat{S}(y) = \pi^{-1}(y)\right] > \frac{1}{2\mathsf{poly}_1(n) \cdot \mathsf{poly}_2(n)} > \frac{1}{\mathsf{poly}(n)}$$

This, however, contradicts equation 3.3 and thus no such construction exists.

However, we wish to point out that this approach leads to a strictly weaker result than what is achieved by using Equation 3.2. Specifically, Equation 3.3 only shows that P is secure for a random oracle but fails to guarantee that there exists any fixed oracle relative to which primitive P is secure. To see this consider the following primitive. We

say that an adversary $\mathcal{A}$ 0^n-*inverts* a permutation π if it outputs $\pi^{-1}(0^n)$. Now, it is easy to prove, by a lazy-sampling argument, that for any polynomial poly there exists a negligible function negl such that for any non-uniform $\mathcal{A}$ of size at most $\mathsf{poly}(n)$

$$\Pr_{\pi \leftarrow \Pi_n} [\mathcal{A} \ 0^n\text{-inverts } \pi] \leq \mathsf{negl}(n).$$

However, for any fixed random permutation oracle $\hat{\Pi} \overset{\text{def}}{=} \{\hat{\pi}_n\}_{n\in\mathbb{N}}$ there exists a non-uniform polynomial size adversary $\mathcal{A}$ (defined dependent on $\hat{\Pi}$) that 0^n-inverts $\hat{\pi}_n$ with probability 1 for all n. Such an $\mathcal{A}$ is given by an adversary that receives as non-uniform advice the sequence of strings $\bar{a} = \hat{\pi}_1^{-1}(0), \hat{\pi}_2^{-1}(00), \dots$ and on input 1^n simply outputs $a_n = \hat{\pi}_n^{-1}(0^n)$. Thus, even though 0^n-uninvertability holds for a random permutation oracle π, it can not hold for any fixed oracle $\hat{\pi}$. Similar considerations arise in the case of collision-resistance and several other primitives. For a more detailed discussion of such issues having to do with oracle-dependent auxiliary-input see [118]. Due to this counter-example, this approach should not be used whenever a fixed oracle implementing P is desired.

Chapter 4

Black-Box Constructions of Predicate Encryption

4.1   Introduction

In this chapter we present the first of our black-box separation results. Specifically, we study the possibility of black-box constructions of predicate encryption from trapdoor permutations.

In a *predicate encryption* scheme [24, 81] an authority generates a master public key and a master secret key, and uses the master secret key to derive personal secret keys for individual users. A personal secret key corresponds to a predicate in some class $\mathcal{F}$, and ciphertexts are associated (by the sender) with an attribute in some set $\mathbb{A}$; a ciphertext associated with the attribute $I \in \mathbb{A}$ can be decrypted by a secret key SK_f corresponding to the predicate $f \in \mathcal{F}$ if and only if $f(I) = 1$. The basic security guarantee provided by such schemes is that a ciphertext associated with attribute I hides all information about the underlying message unless one holds a personal secret key giving the explicit ability to decrypt; i.e., if an adversary $\mathcal{A}$ holds keys $SK_{f_1}, \ldots, SK_{f_\ell}$, then $\mathcal{A}$ learns nothing about the message if $f_1(I) = \cdots = f_\ell(I) = 0$. (A formal definition is given later.)

By choosing $\mathcal{F}$ and $\mathbb{A}$ appropriately, predicate encryption yields as special cases many notions that are interesting in their own right. For example, by taking $\mathbb{A} = \{0,1\}^n$ and letting $\mathcal{F} = \{f_{ID}\}_{ID \in \{0,1\}^n}$ be the class of point functions (so that $f_{ID}(ID') = 1$ iff $ID = ID'$) we recover the notion of identity-based encryption (IBE) [116, 21]. It can

be similarly seen that predicate encryption encompasses fuzzy IBE [115], forward-secure (public-key) encryption [30], (public-key) broadcast encryption [42], attribute-based encryption [66, 16, 95], and more.

Most (though not all) existing constructions of predicate encryption schemes rely on bilinear maps. A natural question is: *what are the minimal assumptions on which predicate encryption can be based?* Of course, the answer will depend on the specific predicate class $\mathcal{F}$ and attribute set $\mathbb{A}$ we are interested in; in particular, Boneh and Waters [24] show that if $\mathcal{F}$ is polynomial size then (for any $\mathbb{A}$) one can construct a predicate encryption scheme for $(\mathcal{F}, \mathbb{A})$ from any (standard) public-key encryption scheme. On the other hand, Boneh et al. [22] have recently shown that there is no *black-box* construction of IBE from trapdoor permutations.

4.1.1   Our Results

The specific question we consider is: *for which $(\mathcal{F}, \mathbb{A})$ can we give a (black-box) construction of a predicate encryption scheme over $(\mathcal{F}, \mathbb{A})$ based on CPA-secure encryption?* We show a characterization of $(\mathcal{F}, \mathbb{A})$ under which no such construction exists. Before describing our results in more detail, we provide some background intuition.

A natural combinatorial construction of a predicate encryption scheme for some $(\mathcal{F}, \mathbb{A})$ from a CPA-secure encryption scheme (Gen, Enc, Dec) is as follows: The authority includes several public keys $pk_1, \ldots, pk_q$ from the underlying encryption scheme in the master public key, and each personal secret key is some appropriate subset of the corresponding secret keys $sk_1, \ldots, sk_q$. Encryption of a message m with respect to an attribute I requires "sharing" m in some way to yield $m_1, \ldots, m_q$, and the resulting ciphertext is

$\mathsf{Enc}_{pk_1}(m_1), \ldots, \mathsf{Enc}_{pk_q}(m_q)$. Intuitively, this works if:

Correctness: Let $SK_f = \{sk_{i_1}, \ldots, sk_{i_t}\}$ be a personal secret key with $f(I) = 1$. Then the set of "shares" $m_{i_1}, \ldots, m_{i_t}$ should enable recovery of m.

Security: Let $\{sk_{i_1}, \ldots, sk_{i_k}\} = \bigcup_{f \in \mathcal{F}: f(I)=0} SK_f$. Then the set of "shares" $m_{i_1}, \ldots, m_{i_k}$ should leak no information about m.

Roughly, our result can be interpreted as showing that this is essentially the *only* way to construct predicate encryption (in a black-box manner) from CPA-secure encryption (or even trapdoor permutations). We now provide further details.

Impossibility results. Our negative results are in the same model used by Boneh et al. [22], which builds on the model used in the seminal work of Impagliazzo and Rudich [76]. Specifically, as in [22] our negative results hold relative to a *random* oracle (with trapdoor) and so rule out black-box constructions from trapdoor permutations as well as from any (standard) public-key encryption scheme secure against chosen-*ciphertext* attacks. All the separations in this chapter are *fully* black-box under the definitions from Chapter 3.

A slightly informal statement of our result follows. Fix $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$, a sequence of predicate classes and attribute sets indexed by the security parameter n. We say that $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ can be *q-covered* if for every set system $\{S_f\}_{f \in \mathcal{F}_n}$ with $S_f \subseteq [q(n)]$ ($[q] \overset{\text{def}}{=} \{1, \ldots, q\}$), there are polynomially-many predicates $f^*, f_1, \ldots, f_p \in \mathcal{F}_n$ such that, with high probability:

1. $S_{f^*} \subseteq \bigcup_{i=1}^p S_{f_i}$.

2. There exists an $I \in \mathbb{A}_n$ with $f_1(I) = \cdots = f_p(I) = 0$ but $f^*(I) = 1$.

44

$\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ is *easily covered* if it is q-covered for every polynomial q. We show:

Main Theorem (informal). *If $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ is easily covered, there is no black-box construction of a predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ based on trapdoor permutations (or CCA-secure encryption).*

Intuitively, if $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ is easily covered then the combinatorial approach discussed earlier cannot work: letting $q(n)$ be the (necessarily) polynomial number of keys for the underlying (standard) encryption scheme, no matter how the secret keys $\{sk_i\}_{i=1}^q$ are apportioned to the personal secret keys $\{SK_f\}_{f \in \mathcal{F}_n}$, an adversary can carry out the following attack (cf. Definition 4.2.2 , below):

1. Request the keys $SK_{f_1}, \ldots, SK_{f_p}$, where each $SK_{f_i} = \{sk_1, \ldots, \} \subseteq \{sk_i\}_{i=1}^q$.

2. Request the challenge ciphertext C to be encrypted using an attribute I for which
$$f_1(I) = \cdots = f_p(I) = 0 \text{ but } f^*(I) = 1.$$

3. Compute the key $SK_{f^*} \subseteq \bigcup_i SK_{f_i}$ and use this key to decrypt C.

This constitutes a valid attack since SK_{f^*} suffices to decrypt C yet the adversary only requested the keys $SK_{f_1}, \ldots, SK_{f_p}$, none of which suffices on its own to decrypt C.

Turning this intuition into a formal proof must, in particular, implicitly show that the combinatorial approach sketched earlier is essentially the *only* black-box approach to building predicate encryption schemes from trapdoor permutations. Moreover, we actually prove a stronger *quantitative* version of the above theorem showing, roughly, that if $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ is q-covered then any predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ must use at least $q + 1$ underlying encryption keys.

One might wonder whether the "easily covered" condition is useful for determining whether there exist black-box constructions of predicate encryption schemes over $\{(\mathcal{F}_n, \mathbb{A}_n)\}_n$ of interest. We show that it is, in that the following corollary can be proven fairly easily given the above:

Corollary *There are no black-box constructions of (1) identity-based encryption[1], (2) forward-secure encryption (for a super-polynomial number of time periods), or (3) broadcast encryption (where a super-polynomial number of users can be excluded) from trapdoor permutations.*

The first result was already proved in [22]; the point is that our impossibility result serves as a strict generalization of theirs. To the best of our knowledge, results (2) and (3) do not follow from result (1), as we do not know a construction of IBE from forward-secure encryption or broadcast encryption with small (but super-polynomial) number of revoked users. We also show quantitative versions of the above corollary that bound, e.g., the number of encryption keys needed to construct forward-secure encryption for any $N = \mathsf{poly}(n)$ time periods.

4.1.2 Comparison to the Results of Boneh et al.

Our proof relies heavily on the impossibility result from [22] for IBE, and indeed our proofs share the same high-level structure. Our contribution lies in finding the right abstraction and generalization (specifically, the "easily covered" property described above) of the *specific* property used by Boneh et al. in the particular case of IBE, adapting their proof to our setting, and applying their ideas to the more general case of predicate encryption. Our generalization, in turn, allows us to show impossibility for several

---

[1]Of course, anything that implies IBE — e.g., attribute-based encryption — is also ruled out.

cryptosystems of interest besides IBE (cf. the corollary stated earlier).

4.2 Definitions

4.2.1 Predicate Encryption

We provide a functional definition of predicate encryption, followed by a weak definition of security that we use in proving impossibility (thus making the result stronger) as well as the standard definition of security [81].

Definition 4.2.1 *Fix $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$, where each $\mathcal{F}_n$ is a set of (efficiently computable) predicates over the set of attributes $\mathbb{A}_n$. A* predicate encryption scheme *over $\{\mathcal{F}_n, \mathbb{A}_n\}_{n \in \mathbb{N}}$ consists of four* PPT *algorithms* PE = (Setup, KeyGen, Enc, Dec) *such that:*

- Setup *is a deterministic algorithm that takes as input a master secret key $MSK \in \{0,1\}^n$ and outputs a master public key MPK.*

- KeyGen *is a deterministic algorithm that takes as input the master secret key MSK and a predicate $f \in \mathcal{F}_n$ and outputs a secret key $SK_f = \mathsf{KeyGen}_{MSK}(f)$. (The assumption that KeyGen is deterministic is without loss of generality, since MSK may include a key for a pseudorandom function.)*

- Enc *takes as input the public key MPK, an attribute $I \in \mathbb{A}_n$, and a bit b. It outputs a ciphertext $C \leftarrow \mathsf{Enc}_{MPK}(I, b)$.*

- Dec *takes as input a secret key SK_f and ciphertext C. It outputs either a bit b or the distinguished symbol $\bot$.*

It is required that for all n, all $MSK \in \{0,1\}^n$ and $MPK = \mathsf{Setup}(MSK)$, all $f \in \mathcal{F}_n$ and $SK_f = \mathsf{KeyGen}_{MSK}(f)$, all $I \in \mathbb{A}_n$, and all $b \in \{0,1\}$, that if $f(I) = 1$ then $\mathsf{Dec}_{SK_f}(\mathsf{Enc}_{MPK}(I, b)) = b$.

For our impossibility result, we rule out constructions achieving even a weak definition of security:

Definition 4.2.2 *A predicate encryption scheme over $(\mathcal{F}, \mathbb{A})$ is* weakly payload hiding *if the advantage of any* PPT *adversary $\mathcal{A}$ in the following game is negligible:*

1. *$\mathcal{A}(1^n)$ outputs $I^* \in \mathbb{A}_n$ and $(f_1, \ldots, f_p) \in \mathcal{F}_n$ such that $f_i(I^*) = 0$ for all i.*

2. *A random $MSK \in \{0,1\}^n$ is chosen; let $MPK := \mathsf{Setup}(MSK)$ and $SK_{f_i} := \mathsf{KeyGen}(MSK, f_i)$ for all i. A random $b \in \{0,1\}$ is chosen, and a random ciphertext $C^* \leftarrow \mathsf{Enc}_{MPK}(I^*, b)$ is computed. $\mathcal{A}$ is given $(MPK, SK_{f_1}, \ldots, SK_{f_p}, C^*)$.*

3. *$\mathcal{A}$ outputs b' and succeeds if $b' = b$.*

The advantage of $\mathcal{A}$ is defined as $\left| \Pr[\mathcal{A} \text{ succeeds}] - \frac{1}{2} \right|$.

The standard security definition [81] follows:

Definition 4.2.3 *A predicate encryption scheme over $(\mathcal{F}, \mathbb{A})$ is* payload hiding *if the advantage of any* PPT *adversary $\mathcal{A}$ in the following game is negligible:*

1. *A random $MSK \in \{0,1\}^n$ is chosen, and $\mathcal{A}$ is given $MPK := \mathsf{Setup}(MSK)$.*

2. *$\mathcal{A}$ may adaptively request keys $SK_{f_1}, \ldots$ corresponding to the predicates $f_1, \ldots \in \mathcal{F}_n$.*

3. *At some point, $\mathcal{A}$ outputs $I^* \in \mathbb{A}_n$. A random $b \in \{0,1\}$ is chosen and $\mathcal{A}$ is given the ciphertext $C^* \leftarrow \mathsf{Enc}_{MPK}(I^*, b)$. $\mathcal{A}$ may continue to request keys for predicates of its choice.*

4. *$\mathcal{A}$ outputs b' and succeeds if (1) $\mathcal{A}$ never requested a key for a predicate f with $f(I^*) = 1$, and (2) $b' = b$.*

The advantage of $\mathcal{A}$ is defined as $\left| \Pr[\mathcal{A} \text{ succeeds}] - \frac{1}{2} \right|$.

4.2.2   A Random Trapdoor Permutation Oracle

We demonstrate our separations by showing a distribution over oracles such that, with overwhelming probability over the choice of $\mathcal{O}$ from this distribution, trapdoor permutations and CCA-secure encryption exist relative to $\mathcal{O}$ (even against non-uniform adversaries) yet any construction of a predicate encryption scheme (for certain $(\mathcal{F}, \mathbb{A})$) using black-box access to $\mathcal{O}$ can be broken with noticeable probability (over the choice of $\mathcal{O}$) by a polynomial time adversary given oracle access to $\mathcal{O}$ and a PSPACE oracle. We refer the reader to Section 3.2.5 for a discussion on why this suffices to prove the separation. The distribution over oracles $\mathcal{O} = (g, e, d)$ is defined as follows, for each $n \in \mathbb{N}$:

- g is chosen uniformly from the space of permutations on $\{0,1\}^n$. We view g as taking a secret key sk as input, and returning a public key pk.

- $e \colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ maps a public key pk and a "message" $m \in \{0,1\}^n$ to an output "ciphertext" $c \in \{0,1\}^n$. It is chosen uniformly subject to the constraint that, for every pk, the function $e(pk, \cdot)$ is a permutation on $\{0,1\}^n$.

- d: $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ maps a secret key sk and a ciphertext c to a message m.

  We require that $d(sk, c)$ outputs the unique m for which $e(g(sk), m) = c$.

One can show [49, 22] that with overwhelming probability over the choice of $\mathcal{O}$ from this distribution, $\mathcal{O}$ is a trapdoor permutation even against an unbounded and non-uniform adversary making at most polynomially many queries to $\mathcal{O}$. Moreover, since the components of $\mathcal{O}$ are chosen at *random* subject to the above constraints (and not with some "defect" as in, e.g., [49]), oracle $\mathcal{O}$ also implies CCA-secure encryption [15].

We denote a query α to $\mathcal{O}$ as, e.g., $\alpha \stackrel{\mathrm{def}}{=} [g(sk) = pk]$ and similarly for e and d queries. In describing our attack in the next section, we often use a partial oracle $\mathcal{O}'$ that is defined only on some subset of the possible inputs. We always enforce that such oracles be *consistent*:

Definition 4.2.4 *A partial oracle $\mathcal{O}' = (g', e', d')$ is* consistent *if:*

1. *For every $pk \in \{0,1\}^n$, the (partial) function $e'(pk, \cdot)$ is one-to-one.*

2. *For every $sk \in \{0,1\}^n$, the (partial) function $d'(sk, \cdot)$ is one-to-one.*

3. *For all $x \in \{0,1\}^n$, and all sk such that $g'(sk) = pk$ is defined, the value $e'(pk, x) = c$ is defined if and only if $d'(sk, c) = x$ is defined.*

4.3 A General Impossibility Result for Predicate Encryption

Here we define a combinatorial property on $(\mathcal{F}_n, \mathbb{A}_n)$ and formally state our impossibility result. Then, in Section 4.4, we describe an adversary attacking any black-box construction of a predicate encryption scheme satisfying the conditions of our theorem

and analyze its probability of success.

Fix a set $\mathcal{F}$ and a positive integer q, and let $[q] \stackrel{\text{def}}{=} \{1, \ldots, q\}$. An $\mathcal{F}$-*set system over* $[q]$ is a collection of sets $\{S_f\}_{f \in \mathcal{F}}$ where each $f \in \mathcal{F}$ is associated with a set $S_f \subseteq [q]$.

Definition 4.3.1 *Let* $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$ *be a sequence of predicates and attributes. We say* $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$ *can be* q-covered *if there exist* ppt *algorithms* (A_1, A_2, A_3), *where* $A_2(1^n, f)$ *is deterministic and outputs* $I \in \mathbb{A}_n$ *with* $f(I) = 1$, *such that for* n *sufficiently large:*

For any $\mathcal{F}_n$-*set system* $\{S_f\}_{f \in \mathcal{F}_n}$ *over* $[q(n)]$, *if we compute*

$$f^* \leftarrow A_1(1^n); \; I^* = A_2(1^n, f^*); \; f_1, \ldots, f_p \leftarrow A_3(1^n, f^*),$$

then with probability at least $4/5$,

1. *$S_{f^*} \subseteq \bigcup S_{f_i}$;*

2. *$f_i(I^*) = 0$ for all i.*

$\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$ *is* easily covered *if it can be* q-covered *for every polynomial* q.

Although the above definition may seem rather complex and hard to use, we show in Section 4.5 that it can be applied quite easily to several interesting classes of predicate encryption schemes. Moreover, the definition is natural given the attack we will describe in the following section.

A black-box construction of a predicate encryption scheme PE $=$ $(\mathsf{Setup}^{\mathcal{O}}, \mathsf{KeyGen}^{\mathcal{O}}, \mathsf{Enc}^{\mathcal{O}}, \mathsf{Dec}^{\mathcal{O}})$ is *q-bounded* if each of its algorithms makes at most q queries to $\mathcal{O}$. We now state our main result, a proof of this theorem appears in Section 4.4:

Theorem 4.3.2 (Main Theorem) *If $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ can be q-covered, there is no q-bounded (fully) black-box construction of a weakly payload hiding predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ from trapdoor permutations (or CCA-secure encryption).*

Since each algorithm defining the predicate encryption scheme can make at most polynomially-many queries to its oracle, we have

Corollary 4.3.3 *If $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ is easily covered, there is no (fully) black-box construction of a weakly payload hiding predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ from trapdoor permutations (or CCA-secure encryption).*

4.4 Proof of Main Theorem

We now prove Theorem 4.3.2 by demonstrating an adversary $\mathcal{A}$ that breaks the security of any black-box construction of a predicate encryption scheme for an easily covered family of predicates and attributes. We first describe the adversary $\mathcal{A}$ in Section 4.4.1 and then analyze its success probability in the sections that follow.

4.4.1 The Attack

Fix an $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ that can be q-covered, and let $\mathsf{PE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a predicate encryption scheme over $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ each of whose algorithms makes at most $q = \mathsf{poly}(n)$ queries to $\mathcal{O} = (g, e, d)$. We assume, without loss of generality, that before any algorithm of PE makes a query of the form $[d(sk, \star)]$, it first makes the query $[g(sk)]$. We additionally assume that PE only queries $\mathcal{O}$ on inputs of length polynomially related to n.

We begin the proof of Theorem 4.3.2 by describing an adversary $\mathcal{A}$ attacking PE. Adversary $\mathcal{A}$ is given access to $\mathcal{O}$ and makes a polynomial number of calls to this oracle; as described, $\mathcal{A}$ is not efficient but it runs in polynomial time given access to a PSPACE oracle (or if P = NP). We then prove that $\mathcal{A}$ succeeds with non-negligible probability over the choice of $\mathcal{O}$, the randomness of $\mathcal{A}$ and the security game. This suffices to prove black-box impossibility as explained in Section 3.2.5.

Let A_1, A_2, and A_3 be as guaranteed by Definition 4.3.1, and let $p = \mathsf{poly}(n)$ bound the number of predicates output by A_3. Throughout $\mathcal{A}$'s execution, when it makes a query to $\mathcal{O}$ it stores the query and the response in a list L. We also require that before $\mathcal{A}$ makes any query of the form $[d(sk, \star)]$, it first makes the query $[g(sk)]$. Furthermore, once the query $[g(sk) = pk]$ has been made then $[e(pk, x) = y]$ is added to L if and only if $[d(sk, y) = x]$ is added to L.

Setup and challenge. $\mathcal{A}(1^n)$ does the following

1. $\mathcal{A}$ computes $f^* \leftarrow A_1(1^n)$, $I^* = A_2(1^n, f^*)$ and $(f_1, \ldots, f_p) \leftarrow A_3(1^n, f^*)$.

    (a) If $f_i(I^*) = 0$ for all i, then $\mathcal{A}$ outputs $(I^*, f_1, \ldots, f_p)$ and receives in return the values $(MPK, SK_{f_1}, \ldots, SK_{f_p}, C^*)$ from the challenger (cf. Definition 2).

    (b) Otherwise, $\mathcal{A}$ aborts and outputs a random bit $b' \leftarrow \{0, 1\}$.

Step 1: Discovering important public keys. For $i = 1$ to p, adversary $\mathcal{A}$ does the following:

1. Compute $I_{f_i} = A_2(1^n, f_i)$, and choose random $b \leftarrow \{0, 1\}$ and $r \leftarrow \{0, 1\}^n$.

2. Compute $\mathsf{Dec}^{\mathcal{O}}_{SK_{f_i}}\left(\mathsf{Enc}^{\mathcal{O}}_{MPK}(I_{f_i}, b; r)\right)$, storing all $\mathcal{O}$-queries in the list L.

Step 2: Discovering frequent queries for I^*. $\mathcal{A}$ repeats the following $q \cdot p^3$ times: Choose random $b \leftarrow \{0, 1\}$ and $r \leftarrow \{0, 1\}^n$; compute $\mathsf{Enc}^{\mathcal{O}}_{MPK}(I^*, b; r)$, storing all $\mathcal{O}$-queries in L.

Step 3: Discovering secret queries and decrypting the challenge. $\mathcal{A}$ chooses $k \leftarrow [q \cdot p^3]$ and runs the following k times.

1. $\mathcal{A}$ uniformly generates a secret key MSK' and a consistent partial oracle $\mathcal{O}'$ for which $\mathsf{Setup}^{\mathcal{O}'}(MSK') = MPK$; for all i it holds that $\mathsf{KeyGen}^{\mathcal{O}'}_{MSK'}(f_i) = SK_{f_i}$; the oracle $\mathcal{O}'$ is consistent with L; and $SK'_{f^*} \stackrel{\text{def}}{=} \mathsf{KeyGen}^{\mathcal{O}'}_{MSK'}(f^*)$ is defined.

   We denote by L' the set of queries in $\mathcal{O}'$ that are not in L (the "invented queries"). Note that $|L'| \leq q \cdot (p + 2)$, at most q queries made by Setup and q queries for each of $SK_{f^*}, SK_{f_1}, \ldots, SK_{f_p}$ made by $\mathsf{KeyGen}(f)$.

2. $\mathcal{A}$ chooses $b \leftarrow \{0, 1\}$ and $r \leftarrow \{0, 1\}^n$, and computes $C = \mathsf{Enc}^{\mathcal{O}}_{MPK}(I^*, b; r)$ (storing all $\mathcal{O}$-queries in L). Then:

   (a) In iteration $k' < k$, adversary $\mathcal{A}$ computes $\mathsf{Dec}^{\mathcal{O}''}_{SK'_{f^*}}(C)$ (where $\mathcal{O}''$ is defined below).

   (b) In iteration k, adversary $\mathcal{A}$ computes $b' = \mathsf{Dec}^{\mathcal{O}''}_{SK'_{f^*}}(C^*)$ (where $\mathcal{O}''$ is defined below).

Output: $\mathcal{A}$ Outputs the bit b' computed in the kth iteration of step 3.

Before defining the oracle $\mathcal{O}''$ used above, we introduce some notation. Let L, $\mathcal{O}'$, and MSK' be as above, and note that we can view L and $\mathcal{O}'$ as a tuple of (partial) functions (g, e, d) and (g', e', d') where g', e', and d' extend g, e, and d, respectively. Define the following:

- $\mathcal{Q}'_S$ is the set of pk for which $[g'(sk) = pk]$ is queried during computation of $\mathsf{Setup}^{\mathcal{O}'}(MSK')$.

- $\mathcal{Q}'_K$ is the set of pk for which $[g'(sk) = pk]$ is queried during computation of $\mathsf{KeyGen}^{\mathcal{O}'}_{MSK'}(f)$ for some $f \in \{f^*, f_1, \ldots, f_p\}$.

- $\mathcal{Q}'_{K-S} = \mathcal{Q}'_K \setminus \mathcal{Q}'_S$.

- L_g is the set of pk for which the query $[g(sk) = pk]$ is in L.

Note that $\mathcal{A}$ can compute each of these sets from its view. Note further that $\mathcal{Q}'_S, \mathcal{Q}'_K, \mathcal{Q}'_{K-S}, \mathcal{O}'$ are fixed throughout an iteration of step 3, but L_g may change as queries are answered.

Oracle $\mathcal{O}''$ is defined as follows. For any query whose answer is defined by $\mathcal{O}'$, return that answer. Otherwise:

1. For an encryption query $e(pk, x)$ with $pk \in \mathcal{Q}'_{K-S} \setminus L_g$, return a random y consistent with the rest of $\mathcal{O}''$ (i.e., ensuring that e remains one-to-one). Act analogously for a decryption query $d(sk, y)$ with $pk \in \mathcal{Q}'_{K-S} \setminus L_g$ (where $pk = g(sk)$).

2. For a decryption query $d(sk, y)$, if there exists a pk such that $[g(sk) = pk] \in \mathcal{O}'$ but[2] there exists an $sk' \neq sk$ with $[g(sk') = pk] \in L$, then use $\mathcal{O}''$ to answer the query $d(sk', y)$.

3. In any other case, query the real oracle $\mathcal{O}$ and return the result. Store the query/answer in L (note that this might affect L_g as well).

---

[2]While $\mathcal{O}'$ is initially chosen to be consistent, a conflict can occur since L is updated as $\mathcal{A}$ makes additional queries to the real oracle $\mathcal{O}$.

The following lemma completes the proof of Theorem 4.3.2:

Lemma 4.4.1 *The probability, over the choice of $\mathcal{O}$, the randomness of $\mathcal{A}$ and the security game, that $\mathcal{A}$ succeeds is $\frac{29}{50} - O\left(\frac{1}{p^2}\right)$, which is noticeably greater than $1/2$ for n sufficiently large.*

We now give a full proof of the above in Sections 4.4.2 - 4.4.6. Specifically, in Section 4.4.2 we describe a series of experiments that aid in our analysis and then analyze the properties of these experiments in the remaining sections. The proof is largely similar to the one from [22] (the full proof of this result is in Papakonstantinou's thesis [98]), with the main difference being Claim 4.4.5. This claim is where we make use of the "easily covered" property of the predicates.

4.4.2 Defining Four Experiments

We now begin the proof of Lemma 4.4.1 by analyzing the success probability of the adversary $\mathcal{A}$. Toward this end, we describe a series of experiments, the first of which corresponds to adversary $\mathcal{A}$ interacting in the experiment from Definition 4.2.2. We show that, as long as no "bad" events (to be defined later) occur, the statistical distance between the transcripts generated in each of these experiments is not too large. This allows us to bound the adversary's success probability by comparing it to an appropriate event in the final experiment. We note that, unless specified otherwise, all probabilities in the remainder of this section are over the choice of random oracle $\mathcal{O}$ (from the distribution in Section 4.2.2) as well as the randomness of $\mathcal{A}$ and the security game.

$Expt_0$**:** This corresponds to adversary $\mathcal{A}$ interacting in the experiment from Definition 4.2.2.

Expt_1: This is the same as Expt_0 except that $\mathcal{O}''$ (as defined after the kth repetition of step 3) is used instead of $\mathcal{O}$ to compute the challenge ciphertext C^*.

Expt_2: This is the same as Expt_1 except that $\mathcal{O}''$ never queries $\mathcal{O}$ (cf. step 3 in the definition of $\mathcal{O}''$); instead, any such queries are answered randomly (subject to ensuring that $\mathcal{O}''$ remains consistent).

Expt_3: This is the following experiment with no adversary and using the real oracle $\mathcal{O}$:

Setup and challenge.

1. Compute $f^* \leftarrow A_1(1^n)$, $I^* = A_2(1^n, f^*)$, and $\{f_1, \dots, f_p\} \leftarrow A_3(1^n, f^*)$.

2. Choose at random $MSK \leftarrow \{0,1\}^n$ and compute $MPK = \mathsf{Setup}^{\mathcal{O}}(MSK)$. If $f_i(I^*) = 1$ for some i, abort and output a random bit.

3. For every predicate $f \in \{f^*, f_1, \dots, f_p\}$ compute $SK_f = \mathsf{KeyGen}^{\mathcal{O}}_{MSK}(f)$.

Step 1: Discovering important public keys. For $i = 1$ to p do:

1. Compute $I_{f_i} \leftarrow A_2(1^n, f_i)$, and choose random $b \leftarrow \{0,1\}$ and $r \leftarrow \{0,1\}^n$.

2. Compute $\mathsf{Dec}^{\mathcal{O}}_{SK_{f_i}}(\mathsf{Enc}^{\mathcal{O}}_{MPK}(I_{f_i}, b; r))$.

Step 2: Decrypting the challenge.

1. Choose $r \leftarrow \{0,1\}^n$, $b \leftarrow \{0,1\}$ and compute $C^* = \mathsf{Enc}^{\mathcal{O}}_{MPK}(I^*, b; r)$.

2. Compute $b' = \mathsf{Dec}^{\mathcal{O}}_{SK_{f^*}}(C^*)$ and output b'. Note that $b' = b$ always.

This completes the description of Expt_3.

For $i \in \{0, 1, 2\}$ we will be interested in the following transcripts defined in the course of Expt_i. These transcripts contain, in particular, all oracle queries/answers made and received.

- trans^i_{setup}: The transcript of the setup phase. This includes the computation of MPK and $SK_{f_1}, \ldots, SK_{f_p}$, as well as the computation of SK_{f^*} for the f^* chosen by the adversary. (Even though SK_{f^*} is not computed in the experiment, SK_{f^*} is well defined given f^*, MSK and $\mathcal{O}$.) Note that the adversary never sees this transcript.

- trans^i_{pks}: The transcript of step 1 ("discovering important public keys").

- trans^i_{freq}: The transcript of step 2 ("discovering frequent queries for I^*").

- $\mathsf{trans}^i_{sim\text{-}setup}$: This is the transcript defined by the adversary's choice of MSK' and $\mathcal{O}'$ in the kth repetition of step 3, and can be viewed as the adversary's "guess" for trans^i_{setup}.

- trans^i_*: The transcript of the encryption of C/decryption of C^* in the kth repetition of step 3.

- $\mathsf{trans}^i = (\mathsf{trans}^i_{setup}, \mathsf{trans}^i_{pks}, \mathsf{trans}^i_{sim\text{-}setup}, \mathsf{trans}^i_*)$.

For Expt_3 we define

- $\mathsf{trans}^3_{sim\text{-}setup}$: The transcript of the "setup and challenge" step.

- trans^3_{pks}: The transcript of step 1 ("discovering important public keys").

- trans^3_*: The transcript of step 2 ("decrypting the challenge").

- $\mathsf{trans}^3 = (\mathsf{trans}^3_{pks}, \mathsf{trans}^3_{sim\text{-}setup}, \mathsf{trans}^3_*)$.

For a given transcript, we partition the set of public keys used (i.e., the set of pk's for which $[g(\cdot) = pk] \in$ trans) into the following sets:

- We let $\mathcal{Q}_S(\text{trans})$ denote the public keys queried during execution of Setup:

$$\mathcal{Q}_S(\text{trans}) \stackrel{\text{def}}{=} \{pk \mid \text{the query } [g(\cdot) = pk] \in \text{trans is asked by Setup}\}.$$

Intuitively, these are the pk's whose corresponding sk's are "useful" for decrypting ciphertexts.

- We let $\mathcal{Q}_K(\text{trans})$ denote the public keys queried by the KeyGen algorithm when some personal secret key is derived:

$$\mathcal{Q}_K(\text{trans}) \stackrel{\text{def}}{=} \{pk \mid [g(\cdot) = pk] \in \text{trans is asked by KeyGen}_{MSK}(\cdot)\}$$

$$\mathcal{Q}_{K-S}(\text{trans}) \stackrel{\text{def}}{=} \mathcal{Q}_K(\text{trans}) \setminus \mathcal{Q}_S(\text{trans}).$$

- Finally, we will also look at the public keys "discovered" during encryption and decryption (cf. step 3 of the experiments):

$$\mathcal{Q}_{ENC+DEC}(\text{trans}, I, f) \stackrel{\text{def}}{=} \{pk \mid [g(\cdot) = pk] \text{ asked by } \mathsf{Dec}_{SK_f}(\mathsf{Enc}_{MPK}(I, \cdot; \cdot))\}$$

4.4.3 Probabilistic Lemmas

Before analyzing the probability that $\mathcal{A}$ succeeds, we prove three simple facts that will be useful in our analysis. The first two of these are just simple probabilistic facts and the last one shows an important property of a random permutation oracle g.

Lemma 4.4.2 *Let $X_1, \ldots, X_{n+1}$ be independent $0, 1$ random variables, where $\Pr[X_i = 1] = p$.*

Let E be the event that $X_1, \ldots, X_n = 1$, but $X_{n+1} = 0$. Then $\Pr[E] \leq \frac{1}{e \cdot n}$

Proof   By independence of the variables, we see that $\Pr[E] = p^n(1 - p)$. This quantity

is maximized at $p = \frac{n}{n+1}$, giving $\Pr[E] \leq \frac{1}{e \cdot n}$. ∎

Lemma 4.4.3 *For any probability space Ω and any function f with domain Ω let x, x' be sampled*

from Ω as follows. First x is sampled from Ω, then x' is sampled from Ω conditioned on $f(x') = $

$f(x)$. Then for every $y \in \Omega$, $\Pr[x = y] = \Pr[x' = y]$.

Proof   For any $y \in \Omega$, let Ball_y be the set of values $y' \in \Omega$ such that $f(y') = f(y)$.

Then, $\Pr[x = y] = \Pr[x \in \mathsf{Ball}_y] \cdot \Pr[x = y \mid x \in \mathsf{Ball}_y]$. Also, $\Pr[x' = y] = \Pr[x' \in$

$\mathsf{Ball}_y] \cdot \Pr[x' = y \mid x' \in \mathsf{Ball}_y]$. However, since x' is chosen conditioned on $f(x') = f(x)$,

$\Pr[x' \in \mathsf{Ball}_y] = \Pr[x \in \mathsf{Ball}_y]$, so $\Pr[x = y] = \Pr[x' = y]$. ∎

Lemma 4.4.4 *For $n \in \mathbb{N}$, let $g : \{0, 1\}^n \to \{0, 1\}^n$ be sampled uniformly from the space of*

permutations on $\{0, 1\}^n$. Then, for any computationally unbounded machine B making $\mathsf{poly}(n)$

oracle queries to g,

$$\Pr[(x, y) \leftarrow B^g(1^n); x \in \{0, 1\}^n \wedge y = g(x) \wedge B \text{ did not query } g(x)] \leq \mathsf{negl}(n)$$

where the probability is over the choice of g.

Proof   Let Y be the set of y values returned by B's queries to g. If B has not queried $g(x)$

then this value is distributed uniformly in the set $Z = \{0, 1\}^n \setminus Y$. Since $|Y| \leq \mathsf{poly}(n)$,

the probability that $g(x) = y$ is bounded by $\frac{1}{2^n - \mathsf{poly}(n)} \leq \mathsf{negl}(n)$. ∎

4.4.4 Bounding Probabilities of Bad Events

To analyze the probability that $\mathcal{A}$ succeeds, we define four "bad" events that would prevent $\mathcal{A}$'s success, and then bound the probabilities of each of them.

E_{NC}^i is the event that either of the following is true (in Expt_i):

1. $\exists f_i \in \{f_1, \ldots, f_p\}$ such that $f_i(I^*) = 1$.

2. The following condition holds:

$$\mathcal{Q}_{ENC+DEC}(\mathsf{trans}_*^i, I^*, f^*) \bigcap \mathcal{Q}_S(\mathsf{trans}_{sim\text{-}setup}^i)$$

$$\not\subseteq \left(\bigcup_{f \in \{f_1, \ldots, f_p\}} \mathcal{Q}_{ENC+DEC}(\mathsf{trans}_{pks}^i, I_f, f) \right) \bigcap \mathcal{Q}_S(\mathsf{trans}_{sim\text{-}setup}^i),$$

where $I_f := A_2(1^n, f)$.

Intuitively, the second condition in the definition of E_{NC} is the event that the set of public keys that are "useful" for $f_1, \ldots, f_p$ does not contain the set of public keys that are "useful" for f^*.

We bound the probability of event E_{NC}^3 using the assumed easily-covered property of $\{(\mathcal{F}_n, \mathbb{A}_n)\}$; this represents the crux of our proof, and motivates our Definition 4.3.1.

Claim 4.4.5 *For any $\mathcal{O}$ and any $MSK \in \{0, 1\}^n$, $\Pr[E_{NC}^3] \le 1/5$, where the probability is over the randomness of A_1 and A_3.*

Proof Fix $\mathcal{O}$ and $MSK \in \{0, 1\}^n$, thus fixing $\mathsf{trans}_{sim\text{-}setup}^3$. In addition, for each predicate $f \in \mathcal{F}_n$, fix a string r_f that is sufficiently long to run $\mathsf{Dec}_{SK_f}(\mathsf{Enc}_{MPK}(I, b; r))$ (where

$I \stackrel{\text{def}}{=} A_2(f))$, then this defines, for each f, the set

$$S_f \stackrel{\text{def}}{=} \{pk \mid [g(\cdot) = pk] \text{ is asked by } \mathsf{Dec}_{SK_f}(\mathsf{Enc}_{MPK}(I, b; r))\} \cap \mathcal{Q}_S(\mathsf{trans}^3_{sim\text{-}setup}).$$

Numbering the (at most q) public keys in $\mathcal{Q}_S(\mathsf{trans}^3_{sim\text{-}setup})$ in lexicographic order, we can view these $\{S_f\}_{f \in \mathcal{F}_n}$ as an $\mathcal{F}_n$-set system over $[q]$. The fact that $\{(\mathcal{F}_n, \mathbb{A}_n)\}$ can be q-covered then implies that there exists a polynomial p such that

$$\Pr\left[\begin{array}{c} f^* \leftarrow A_1, \ I^* = A_2(1^n, f^*) \\ \{f_1, \ldots, f_p\} \leftarrow A_3(f^*) \end{array} : \left(S_{f^*} \subseteq \bigcup_{i=1}^p S_{f_i}\right) \bigwedge (\forall i : f_i(I^*) = 0)\right] \geq \frac{4}{5}. \quad (4.1)$$

The above is exactly a lower bound on the probability that E^3_{NC} does not occur. $\blacksquare$

Let abort be the event that there is an abort in the "setup and challenge" step of any of the experiments. (It is not hard to see that the probability of abort is the same in all the experiments since it only depends on A_1 and A_3.) Note that $\Pr[\mathsf{abort}] \leq \Pr[E^3_{NC}]$.

E^i_{HQ} is the event that a *hidden query* appears in trans^i_*. A query α is hidden if one of the following holds:

1. $\alpha \in \mathsf{trans}^0_{setup} \setminus L$.

2. α is of the form $[e(pk, x)]$, and there exists sk, y such that $[g(sk) = pk], [d(sk, y) = x] \in \mathsf{trans}^0_{setup} \setminus L$.

3. α is of the form $[d(sk, x)]$, and there exists an x such that $[g(sk) = pk], [e(pk, x) = y] \in \mathsf{trans}^0_{setup} \setminus L$.

Intuitively, E_{HQ} is the event that a query used by Setup that is necessary to encrypt or

decrypt the challenge ciphertext is not found in the attack.

Claim 4.4.6 *For any $\mathcal{O}$, $\Pr[E_{HQ}^0 \mid \neg\mathsf{abort}] \leq \frac{3\cdot(p+2)}{p^3} = O\left(\frac{1}{p^2(n)}\right)$, where the probability is over the randomness of $\mathcal{A}$ and the security game.*

Proof In step 3 of the attack, $\mathcal{A}$ chooses a random round $k \leq q \cdot p^3$ in which to decrypt the challenge ciphertext. In each of the k repetitions of step 3 the ciphertext is computed exactly the same way as the challenge, using the real oracle $\mathcal{O}$, a random bit b, and randomness r. Note that $|\mathsf{trans}_{setup}^0| \leq q \cdot (p+2)$, since Setup makes at most q queries, and at most q queries are made for each of the $p+1$ keys $SK_{f^*}, SK_{f_1}, \ldots, SK_{f_p}$. For each query in trans_{setup}^0, there are at most 3 hidden queries; thus, there are at most $3q \cdot (p+2)$ hidden queries. By definition of $\mathcal{O}''$, any hidden queries found in step 3 are queried to the real oracle $\mathcal{O}$ and stored in L; i.e., each hidden query is only found once. We conclude that there are at most $3q \cdot (p+2)$ rounds in which a hidden query is found. The probability that the kth round is such a round is at most $\frac{3q\cdot(p+2)}{q\cdot p^3} = \frac{3(p+2)}{p^3}$. ∎

E_{KG} is the event that there exists a public key $pk \in \mathcal{Q}_{K-S}(\mathsf{trans}_{sim\text{-}setup}^0) \setminus L_g$ such that $[e(pk, \cdot) = \cdot] \in \mathsf{trans}_{freq}^0$. Intuitively, E_{KG} is the event that one of the executions of Enc uses a public key that was generated by KeyGen for some predicate f but was not generated by Setup.

Claim 4.4.7 $\Pr[E_{KG}^0 \mid \neg\mathsf{abort}]$ *is negligible in n, where the probability is over the choice of $\mathcal{O}$, the randomness of $\mathcal{A}$ and the security game.*

Proof The proof shows that if $\Pr[E_{KG}^0]$ is not negligible, then there is an adversary $\mathcal{A}'$ that makes polynomially-many queries to a random oracle g and outputs with non-

negligible probability a pair (sk, pk) where $pk = g(sk)$ but $g(sk)$ was never queried. By Lemma 4.4.4, this can only happen with negligible probability.

We start by analyzing the probability of the following, related, event E'_{KG} defined with respect to trans_{setup} instead of $\mathsf{trans}_{sim\text{-}setup}$. Let L_{freq} be the subset of the queries in L_g that is found during step 2 of the attack ("discovering frequent queries"). Now, let E'_{KG} be the event that there exists $pk \in \mathcal{Q}_{K-S}(\mathsf{trans}^0_{setup}) \setminus L_{freq}$ such that $[e(pk, \cdot) = \cdot] \in \mathsf{trans}^0_{freq}$. We bound the probability of E'_{KG}.

Let $\mathcal{A}'$ be the following adversary. $\mathcal{A}'$, given a random random oracle $g : \{0,1\}^n \to \{0,1\}^n$ simulates the TDP oracle as follows. Whenever a query $[g(sk)]$ is made he queries it to his random oracle g. When a query $[e(pk, \cdot)]$ or $[d(sk, \cdot)]$ is made he just returns a random string in $\{0,1\}^n$ making sure that $e(pk, \cdot)$ is a permutation for every pk. This simulated oracle is distributed exactly like a random TDP oracle $\mathcal{O}$.

Now, $\mathcal{A}'$ simulates the setup and challenge as well as the discovering frequent queries steps of $\mathcal{A}$'s attack in Expt_0 using this simulated oracle. In his simulation, $\mathcal{A}'$ reverses the order of encryption and key generation, running all the encryptions for f^* first and only then generating the secret keys for $f_1, \ldots, f_p$. At a random query α in the key generation process $\mathcal{A}'$ stops (without answering α). He selects at random a query β asked during the encryptions for f^*.

Assume that E'_{KG} occurred. Then, with probability at least $\left(\frac{1}{p \cdot q}\right)\left(\frac{1}{q^2 \cdot p^3}\right) \geq \frac{1}{\mathsf{poly}(n)}$ (for some polynomial poly), α is a query $[g(sk)]$ and β is a query $[e(pk, \cdot)]$ such that $g(sk) = pk$, as guaranteed to exist by E'_{KG}. If this is the case, then $\mathcal{A}'$ outputs the pair (sk, pk). Note that $g(sk) = pk$ for the random oracle g and $\mathcal{A}'$ never queried $g(sk)$. Therefore, $\Pr[E'_{KG}] \leq \frac{\mathsf{poly}(n)}{2^n} \leq \mathsf{negl}(n)$.

To bound $\Pr[E_{KG}]$ we need to relate it to $\Pr[E'_{KG}]$. To do this, we consider the pairs of transcripts $t_1 = (\mathsf{trans}^0_{setup}, \mathsf{trans}^0_{freq})$ and $t_2 = (\mathsf{trans}^0_{sim\text{-}setup}, \mathsf{trans}^0_{freq})$. The pair t_2 is the adversary's guess at t_1 after he sees some information about it. That is, he sees all of trans^0_{freq}, the secret keys of $f_1, \ldots, f_p$ and the public parameters. Let r be the additional randomness that determines $\mathcal{A}$'s information about t_1. More formally, $\mathcal{A}$'s view is $\delta = F(t_1, r)$ for some function F. $\mathcal{A}$ then chooses t_2 and r' conditioned on them resulting in the same view, $F(t_2, r') = \delta$. Therefore, by Lemma 4.4.3, $t_2 = t_1$.

The above shows that the probability that there exists a pk such that $[g(sk) = pk] \in \mathcal{Q}_{K-S}(\mathsf{trans}^0_{setup}) \setminus L_{freq}$ is negligible in n. Since, $L_{freq} \subseteq L_g$, the probability of there being such a pk in $\mathcal{Q}_{K-S}(\mathsf{trans}^0_{setup}) \setminus L_g$ is also bounded by the above, proving the claim. ∎

Finally, E^i_{FK} is the event that there exists a public key pk such that $[e(pk, \cdot) = \cdot] \in \mathcal{O}'$, $pk \notin \mathcal{Q}_S(\mathsf{trans}^i_{sim\text{-}setup}) \cup \mathcal{Q}_K(\mathsf{trans}^i_{sim\text{-}setup})$, and there exists an sk such that $[g(sk) = pk] \in \mathsf{trans}^i_*$. Intuitively, this means that after $\mathcal{A}$ invents the answers to some $e(pk, \cdot)$ queries during the attack, the transcript trans_* ends up containing a query $[g(sk) = pk]$. This is a bad event because this may result in invalid decryption using sk.

Claim 4.4.8 $\Pr[E^i_{FK} \mid \neg\mathsf{abort}]$ *is negligible in n for $i \in \{0, 1, 2\}$, where the probability is over the choice of $\mathcal{O}$, the randomness of $\mathcal{A}$ and the security game.*

Proof The proof for $i \in \{0, 1\}$ is similar to the proof of Claim 4.4.7. For $i = 2$, remember that, in Expt_2, whenever a query $[g(sk)]$ is made to $\mathcal{O}''$ the answer is chosen uniformly from $\{0, 1\}^n$ ($\mathcal{O}$ is never queried in step 3 of Expt_2). There are at most $2q$ queries in trans^2_* and at most $q \cdot (p + 2)$ queries in L' (the list of made up queries in $\mathcal{O}'$). Therefore, the probability that one of the $2q$ random pk returned in answer to a query in trans^2_* equals

one of the pk in L' is negligible in n.

∎

4.4.5 Analyzing the Experiments

Now we are ready to compare the transcripts of the experiments, always conditioned on the event that abort did not occur.

Claim 4.4.9 $SD(\mathsf{trans}^0, \mathsf{trans}^1) \leq \frac{3(p+2)}{ep^3} + \mathsf{negl}(n) = O\left(\frac{1}{p^2(n)}\right)$ *(even conditioned on ¬abort).*

Proof Expt_0 and Expt_1 only differ in the way that C^* is computed. In Expt_0 it is computed using the real oracle $\mathcal{O}$, while in Expt_1 it is computed using the hybrid oracle $\mathcal{O}''$. Thus it is enough to prove the following claim.

Claim 4.4.10 *Choose $MSK \leftarrow \{0,1\}^n$ and a random oracle $\mathcal{O}$ (from the distribution in Section 4.2.2). Let $MPK = \mathsf{Setup}^{\mathcal{O}}(MSK)$ and let f^* be the challenge predicate, $I^* = A_2(f^*)$ be the challenge attribute and $\mathcal{O}''$ be the oracle created by the adversary in the kth repetition of step 3. Then*

$$\Pr[\mathsf{Enc}^{\mathcal{O}}_{MPK}(I^*, b; r) \neq \mathsf{Enc}^{\mathcal{O}''}_{MPK}(I^*, b; r)] \leq \frac{3(p+2)}{ep^3} + \mathsf{negl}(n)$$

where the probability is over the choice of MSK and $\mathcal{O}$, the randomness of $\mathcal{A}$, and the choice of random tape $r \leftarrow \{0,1\}^n$ and $b \leftarrow \{0,1\}$.

Proof Consider a query α to $\mathcal{O}''$ that is asked by Enc. Let E_α be the event that α is not asked during step 2 of the attack, but is asked by $\mathsf{Enc}^{\mathcal{O}''}_{MPK}(I^*, b; r)$ (in the kth repetition of step 3). For these $q \cdot p^3 + 1$ encryptions, let X_i be an indicator random variable such

that $X_i = 1$ if α is not asked in the ith encryption. Clearly, E_α is the event that $X_i = 1$ for $i \leq q \cdot p^3$ and $X_{q \cdot p^3 + 1} = 0$. Using Lemma 4.4.2 we get that $\Pr[E_\alpha] \leq \frac{1}{eq \cdot p^3}$.

Let $\mathcal{O}'$ be the invented oracle in iteration k of step 3 and let L' be the list of queries in $\mathcal{O}' \setminus L$ (the invented queries). Remember that $|L'| \leq q \cdot (p + 2)$. The only way the two encryptions in the statement of the claim can differ is if $\mathsf{Enc}_{MPK}^{\mathcal{O}''}(I^*, b; r)$ asks a query α that is dependent on the invented set of queries L'. We show that this happens with probability at most $\frac{3(p+2)}{e \cdot p^3} + \mathsf{negl}(n)$. There are three ways the event in question can occur:

1. $\alpha \in L'$. By definition of L', α is not asked in step 2 of the attack, implying that E_α has occurred. But E_α occurs with probability at most $\frac{1}{eq \cdot p^3}$ for each query $\alpha \in L'$. Since there are at most $q \cdot (p + 2)$ queries in L', the probability such an α is asked is at most $\frac{q \cdot (p+2)}{eq \cdot p^3} = \frac{p+2}{e \cdot p^3}$.

2. α is of the form $[e(pk, \cdot)]$ such that $[g(\cdot) = pk] \in \mathcal{Q}_{K-S} \setminus L$. (Remember that $\mathcal{O}''$ answers such queries randomly.) By Claim 4.4.7, such a query α is in L_{freq} with at most negligible probability. However, if $\alpha \notin L_{freq}$ then E_α has occurred implying that such a query is asked with probability at most $\frac{p+2}{e \cdot p^3} + \mathsf{negl}(n)$.

3. α is of the form $[d(sk, \cdot)]$ such that query $\beta = [g(sk) = \cdot] \in L'$. Note that this means that the query $\beta \notin L$. Also remember that whenever α is queried in the experiment, we also query β to $\mathcal{O}''$. However, this would imply that event E_β has occurred and thus happens with probability at most $\frac{p+2}{e \cdot p^3}$ as above.

Summing these up completes the proof of Claim 4.4.10, and hence Claim 4.4.9 as well. ■

■

67

Next, we compare Expt_2 and Expt_3. We prove

Claim 4.4.11 $(\mathsf{trans}^2_{pks}, \mathsf{trans}^2_{sim\text{-}setup}, \mathsf{trans}^2_*) = \mathsf{trans}^3$.

Proof Since, in Expt_2, $\mathcal{O}''$ never queries $\mathcal{O}$, $\mathcal{O}''$ is just a random TDP oracle. Therefore, it is easy to see that the marginal distributions $(\mathsf{trans}^2_{setup}, \mathsf{trans}^2_{pks})$ and $(\mathsf{trans}^3_{sim\text{-}setup}, \mathsf{trans}^3_{pks})$ are identical. To see that the distribution $(\mathsf{trans}^2_{sim\text{-}setup}, \mathsf{trans}^2_{pks})$ is also identical to these, note that $\mathcal{A}$ gets some partial information $\delta = F(\mathsf{trans}^2_{setup}, \mathsf{trans}^2_{pks}, r)$ for some randomness r, and then uniformly generates $(\mathsf{trans}^2_{sim\text{-}setup}, \mathsf{trans}^2_{pks})$ consistent with δ. This is the same argument as in the proof of Claim 4.4.7.

To see that $\mathsf{trans}^2_* = \mathsf{trans}^3_*$, note that the oracle $\mathcal{O}''$ used in Expt_2 is just a random extension of the oracle $\mathcal{O}'$ (in Expt_2 all queries not in $\mathcal{O}'$ are answered randomly). Since $\mathcal{O}'$ contains all the queries from $(\mathsf{trans}^2_{sim\text{-}setup}, \mathsf{trans}^2_{pks})$ these are all consistent with $\mathcal{O}''$. In Expt_3, all the queries are answered consistently with the random oracle $\mathcal{O}$ which contains all the queries in $(\mathsf{trans}^2_{sim\text{-}setup}, \mathsf{trans}^2_{pks})$. Since $\mathcal{O}$ and $\mathcal{O}''$ are identically distributed random oracles, $\mathsf{trans}^2_* = \mathsf{trans}^3_*$, implying the claim. ∎

To complete the comparisons of the experiments we need to compare the transcripts of Expt_1 and Expt_2.

Claim 4.4.12 $SD(\mathsf{trans}^1, \mathsf{trans}^2) \leq \frac{2}{5} + O\left(\frac{1}{p^2(n)}\right)$ *(even conditioned on ¬abort).*

Proof Expt_1 and Expt_2 only differ in the way $\mathcal{O}''$ works. It therefore follows immediately that $(\mathsf{trans}^1_{sim\text{-}setup}, \mathsf{trans}^1_{pks}) = (\mathsf{trans}^2_{sim\text{-}setup}, \mathsf{trans}^2_{pks})$, and we only need to compare trans^1_* and trans^2_*.

The transcripts trans^1_* and trans^2_* consist of the (at most $2q$) queries asked during $\mathsf{Enc}_{MPK}(I^*, b; r)$ and $\mathsf{Dec}_{SK_{f^*}}(C^*)$. Let α^j_i be the distribution on the ith query asked in trans^j_* for $1 \leq i \leq 2q$ (we will use $\overline{\alpha}^j_i$ to indicate a value sampled from this distribution). Let β^j_i be the distribution of answers to α^j_i. When there are less than i queries in trans^j_* then we set $(\overline{\alpha}^j_i, \overline{\beta}^j_i) = (\perp, \perp)$.

In order to compare Expt_1 and Expt_2 we introduce the following two intermediate experiments. Let $E^k = E^k_{FK} \vee E^k_{HQ} \vee E^k_{NC}$ for $k \in \{1, 2\}$. $\mathsf{Expt}_{1'}$ proceeds exactly like Expt_1 up to the first query $\overline{\alpha}_i$ for which E^1 occurs. When E^1 occurs $\mathsf{Expt}_{1'}$ stops and sets $(\overline{\alpha}^{1'}_j, \overline{\beta}^{1'}_j) = (\perp, \perp)$ for all $j \geq i$. Similarly, we define $\mathsf{Expt}_{2'}$ to run as Expt_2 until E^2 occurs. All random variables defined in Expt_i are also defined in $\mathsf{Expt}_{i'}$ and we use a prime superscript to differentiate the variables.

The proof of the following is identical to the one given by [22], but we include it here for completeness:

Claim 4.4.13 $SD(\mathsf{trans}^{1'}, \mathsf{trans}^{2'}) = \mathsf{negl}(n)$ *(even conditioned on $\neg\mathsf{abort}$).*

Proof To compare $\mathsf{trans}^{1'}$ and $\mathsf{trans}^{2'}$ we only need to compare the distributions $(\alpha^{1'}_i, \beta^{1'}_i)_{i \in [2q]}$ and $(\alpha^{2'}_i, \beta^{2'}_i)_{i \in [2q]}$. Let $1 \leq i \leq 2q$, and suppose that $\alpha^{1'}_j = \alpha^{2'}_j$ and $\beta^{1'}_j = \beta^{2'}_j$ for all $j < i$. Note, that this implies that $\alpha^{1'}_i = \alpha^{2'}_i$ since a query is determined by all the previous queries and answers. We show that the distributions on the answers given in the two experiments, $\beta^{1'}_i$ and $\beta^{2'}_i$, are statistically close. We define $\mathcal{O}''_i = \mathcal{O}' \cup \{\overline{\alpha}_j \to \overline{\beta}_j \mid 1 \leq j < i\}$.

We prove the claim by bounding the statistical distance of $\beta^{1'}_i$ and $\beta^{2'}_i$ for every possible query $\overline{\alpha}^{1'}_i$. Note that $\mathcal{O}''_i$ and $\mathcal{O}'$ are always equal in $\mathsf{Expt}_{1'}$ and $\mathsf{Expt}_{2'}$ since we

assumed that $\alpha_j^{1'} = \alpha_j^{2'}$ and $\beta_j^{1'} = \beta_j^{2'}$ for all $j < i$. We split up the possible queries into two sub-categories as follows.

1. The answers $\beta_i^{2'}$ and $\beta_i^{1'}$ are determined by $\mathcal{O}_i''$

   (a) If $\overline{\alpha}_i^{1'} = \perp$ then $\overline{\beta}_i^{1'} = \overline{\beta}_i^{2'} = \perp$ so $\beta_i^{1'} = \beta_i^{2'}$.

   (b) If there exists an answer $\overline{\beta}$ such that $[\overline{\alpha}_i^{1'} = \overline{\beta}] \in \mathcal{O}_i''$ then $\overline{\beta}_i^{1'} = \overline{\beta}_i^{2'} = \overline{\beta}$ so $\beta_i^{1'} = \beta_i^{2'}$.

   (c) If $\overline{\alpha}_i^{1'}$ is a query $[g(sk) = pk] \in \mathcal{O}_i'' \setminus \mathcal{O}'$. Since, $\overline{\alpha}_i^{1'} \neq \perp$ we know that[3] $\neg E_{FK}^{1'}$, implying that there is no query of the form $[e(pk, \cdot) = \cdot] \in \mathcal{O}'$. Also, since $[g(sk) = pk] \notin \mathcal{O}'$ there is no query $[d(sk, \cdot) = \cdot] \in \mathcal{O}'$. Note that since $\overline{\alpha}_i^{1'} \in \mathcal{O}_i''$ there is already an answer, $\overline{\beta} = \overline{\beta}_j^{1'}$, defined for this query in $\mathcal{O}''$ for some $j < i$. Remember that we assumed $\beta_j^{1'} = \beta_j^{2'}$. By definition of $\mathsf{Expt}_{1'}$ and $\mathcal{O}''$, $\overline{\beta}_j^{1'}$ must have come from the real oracle $\mathcal{O}$ and $\overline{\alpha}_i^{1'}$ will thus be answered consistently. In $\mathsf{Expt}_{2'}$, this query $\overline{\alpha}_i^{2'}$ answered according to a random $\mathcal{O}''$ consistent with all previous queries, implying that $\beta_i^{1'} = \beta_i^{2'}$.

   (d) Consider all $pk \in \mathcal{Q}_S(\mathsf{trans}_{sim\text{-}setup}^{1'}) \setminus L_g$. Since $\overline{\alpha}_i^{1'} \neq \perp$ we know that $\neg E_{NC}^{1'}$. Therefore, no query $\overline{\alpha}_i^{1'}$ of the form $[d(sk, y)]$ such that $[g(sk) = pk] \in \mathcal{O}'$ is asked in $\mathsf{trans}_*^{1'}$.

   (e) Consider all $pk \in \mathcal{Q}_{K-S}(\mathsf{trans}_{sim\text{-}setup}^{1'}) \setminus L_g$. In both $\mathsf{Expt}_{1'}$ and $\mathsf{Expt}_{2'}$ such queries are answered randomly without ever using $\mathcal{O}$. Therefore, the distributions $\beta_i^{1'} = \beta_i^{2'}$.

   (f) Consider all $pk \in (\mathcal{Q}_S(\mathsf{trans}_{sim\text{-}setup}^{1'}) \cup \mathcal{Q}_K(\mathsf{trans}_{sim\text{-}setup}^{1'})) \cap L_g$. Now consider

---

[3]In this proof, when we say $\neg E$, we mean that event E has not occurred through query i

the case that $\bar{\alpha}_i^{1'}$ is of the form $[d(sk, y)]$ such that $[g(sk) = pk] \in \mathcal{O}'$ and there exists a query $\alpha' = [e(pk, x) = y] \in \mathcal{O}_i''$. Note that the answer to α', may be from the real oracle $\mathcal{O}$, or made up by the adversary in $\mathcal{O}'$. In either case, since $pk \in L_g$, there exists an sk' such that $[g(sk') = pk] \in L$. This means that before $\bar{\alpha}_i^{1'}$ is asked to $\mathcal{O}''$ it is first modified to $[d(sk', y)]$.

Now, we analyze the possible answers to $\bar{\alpha}_i^{1'}$. If α' was answered by $\mathcal{O}$, then we get that $\bar{\beta}_i^{1'} = d(sk', y) = x$ by the correctness of $\mathcal{O}$. Also, $\bar{\beta}_i^{2'} = x$, because this is the answer contained in $\mathcal{O}''$. If, on the other hand, the answer to α' is made up by $\mathcal{A}$ in $\mathcal{O}'$, we know that the query $[d(sk, y) = x]$ is also in $\mathcal{O}'$. This is because $[g(sk') = pk] \in L \subseteq \mathcal{O}'$ implying that $[e(pk, x) = y] \in \mathcal{O}' \iff [d(sk', y) = x] \in \mathcal{O}'$. Since $\bar{\beta}_i^{1'}$ and $\bar{\beta}_i^{2'}$ are both answered consistently with $\mathcal{O}'$, we get that $\bar{\beta}_i^{1'} = \bar{\beta}_i^{2'} = x$. A symmetric argument works for the case when $\bar{\alpha}_i^{1'}$ is of the form $[e(pk, x)]$ except that the query is never modified.

2. The answers $\beta_i^{2'}$ and $\beta_i^{1'}$ are not determined in $\mathcal{O}_i''$

    (a) If $\bar{\alpha}_i^{1'}$ is of the form $[g(sk)]$. Since $\bar{\alpha}_i^{1'} \neq \bot$, we know that $\neg E_{HQ}^{1'}$ implying that there are no hidden queries in $\text{trans}_*^{1'}$ so $[g(sk) = \cdot] \notin \text{trans}_{setup}^{1'}$. This means that this is the first time that $[g(sk)]$ is queried to $\mathcal{O}$. Therefore, the answer is distributed uniformly in $\{0, 1\}^n$ in both $\text{Expt}_{1'}$, where it is queried to the random oracle $\mathcal{O}$, and in $\text{Expt}_{2'}$, where it is generated at random. So, the distributions $\beta_i^{1'} = \beta_i^{2'}$.

    (b) If $\bar{\alpha}_i^{1'}$ is of the form $[e(pk, x)]$. Since $\bar{\alpha}_i^{1'} \neq \bot$, we know that $\neg E_{HQ}^{1'}$ implying that there are no hidden queries in $\text{trans}_*^{1'}$. Therefore, $\bar{\alpha}_i^{1'} \notin \text{trans}_{setup}^{1'}$ and there do

not exist a secret key sk and a value y such that $[g(sk) = pk]$, $[d(sk, y) = x] \in$ $\mathsf{trans}^{1'}_{setup}$. Therefore, $\overline{\alpha}^{1'}_i$ is queried to the real oracle $\mathcal{O}$, returning a uniformly random value $\overline{\beta}^{1'}_i$ that has not been previously assigned to $e(pk, \cdot)$ by $\mathcal{O}$. Thus $\beta^{1'}_i$ is the uniform distribution over the set $B_{pk} = \{y \in \{0, 1\}^n$ not assigned by $\mathcal{O}$ as the answer to any query $[e(pk, x')]\}$.

Since we also know that $\neg E^{2'}_{HQ}$ ($\overline{\alpha}^{2'}_i = \overline{\alpha}^{1'}_i \neq \perp$), we know that the value $\overline{\beta}^{2'}_i$ is distributed uniformly in such a way as to keep $e(pk, \cdot)$ a permutation relative to $\mathcal{O}''_i$. Thus, $\beta^{2'}_i$ is the uniform distribution over the set $A_{pk} = \{y \mid \nexists x'$ s.t. $[e(pk, x') = y] \in \mathcal{O}''_i\}$.

Note that the sets A_{pk} and B_{pk} consist of all possible strings in $\{0, 1\}^n$ except for the ones that were queried to either $\mathcal{O}$ or $\mathcal{O}''$ during the attack. Therefore, $|A_{pk} \cap B_{pk}| \geq 2^n - \mathsf{poly}(n)$. This gives us that $|A_{pk} \triangle B_{pk}| \leq \mathsf{poly}(n)$ implying that $SD(\beta^{1'}_i, \beta^{2'}_i) \leq \frac{2\mathsf{poly}(n)}{2^n}$.

(c) If $\overline{\alpha}^{1'}_i$ is of the form $[d(sk, y)]$ then a symmetric argument shows $SD(\beta^{1'}_i, \beta^{2'}_i) \leq \frac{\mathsf{poly}(n)}{2^n}$

We now know that for any $i \in [2q]$ s.t $\alpha^{2'}_j = \alpha^{1'}_j$ and $\beta^{2'}_j = \beta^{1'}_j$ for all $j < i$, $SD(\beta^{1'}_i, \beta^{2'}_i) \leq \frac{2\mathsf{poly}(n)}{2^n}$. Taking a union bound over all possible values of i we get that, $SD(\mathsf{trans}^{1'}_*, \mathsf{trans}^{2'}_*) \leq \frac{2q \cdot 2\mathsf{poly}(n)}{2^n}$ implying the claim. ∎

To finish the proof of Claim 4.4.12 we prove the following lemma bounding the statistical distance between trans^1 and $\mathsf{trans}^{1'}$ as well as trans^2 and $\mathsf{trans}^{2'}$.

Claim 4.4.14 $SD(\mathsf{trans}^2, \mathsf{trans}^{2'}) \leq \frac{1}{5} + O\left(\frac{1}{p^2(n)}\right)$ and $SD(\mathsf{trans}^1, \mathsf{trans}^{1'}) \leq \frac{1}{5} + O\left(\frac{1}{p^2(n)}\right)$. *(In each case, this holds even conditioned on $\neg$abort.)*

Proof   We implicitly condition on ¬abort in everything that follows. Experiments Expt_2 and $\mathsf{Expt}_{2'}$ proceed identically unless event $E^2 = E_{FK}^2 \vee E_{HQ}^2 \vee E_{NC}^2$ occurs in Expt_2, or $E^{2'}$ occurs in $\mathsf{Expt}_{2'}$. Therefore, $\Pr[E^2] = \Pr[E^{2'}]$. In addition, we have that $\Pr[E_{FK}^{2'}] \leq \Pr[E_{FK}^2]$, $\Pr[E_{HQ}^{2'}] \leq \Pr[E_{HQ}^2]$ and $\Pr[E_{NC}^{2'}] \leq \Pr[E_{NC}^2]$. To see this, note that $E_{FK}^{2'} \Rightarrow E_{FK}^2$ (this also holds for E_{HQ} and E_{NC}) since $E_{FK}^{2'}$ only when E_{FK}^2 occurs. Note, however, that the reverse implication, $E_{FK}^2 \Rightarrow E_{FK}^{2'}$, is not necessarily true as E_{FK}^2 may occur after another bad event has already occurred, while $E_{FK}^{2'}$ can not occur in this case since $\mathsf{Expt}_{2'}$ aborts. Similar arguments hold for Expt_1 and $\mathsf{Expt}_{1'}$. Now:

1. From Claim 4.4.8, we know $\Pr[E_{FK}^{2'}] \leq \Pr[E_{FK}^2] = \mathsf{negl}(n)$ and $\Pr[E_{FK}^{1'}] \leq \Pr[E_{FK}^1] = \mathsf{negl}(n)$.

2. From Claim 4.4.6, we know $\Pr[E_{HQ}^0] = O\left(\frac{1}{p^2(n)}\right)$.

3. Applying Claim 4.4.9, we know $\Pr[E_{HQ}^{1'}] \leq \Pr[E_{HQ}^1] \leq \Pr[E_{HQ}^0] + O\left(\frac{1}{p^2(n)}\right) = O\left(\frac{1}{p^2(n)}\right)$.

4. Applying Claim 4.4.13, we get that $\Pr[E_{HQ}^{2'}] \leq \Pr[E_{HQ}^{1'}] + \mathsf{negl}(n) = O\left(\frac{1}{p^2(n)}\right)$.

5. From Claim 4.4.5, we know that $\Pr[E_{NC}^{2'}] \leq \Pr[E_{NC}^2] = \Pr[E_{NC}^3] \leq 1/5$.

6. Applying Claim 4.4.13, we get that $\Pr[E_{NC}^{1'}] \leq \Pr[E_{NC}^{2'}] + \mathsf{negl}(n) \leq 1/5 + \mathsf{negl}(n)$.

Therefore,

$$SD(\mathsf{trans}^2, \mathsf{trans}^{2'}) = \Pr[E^{2'}] = \Pr[E_{FK}^{2'}] + \Pr[E_{HQ}^{2'}] + \Pr[E_{NC}^{2'}] \leq \frac{1}{5} + O\left(\frac{1}{p^2(n)}\right)$$

$$SD(\mathsf{trans}^1, \mathsf{trans}^{1'}) = \Pr[E^{1'}] = \Pr[E_{FK}^{1'}] + \Pr[E_{HQ}^{1'}] + \Pr[E_{NC}^{1'}] \leq \frac{1}{5} + O\left(\frac{1}{p^2(n)}\right),$$

as desired. ∎

Combining Claims 4.4.13 and 4.4.14, we get Claim 4.4.12. ∎

4.4.6   Completing the Proof

Let abort be the event that there is an abort in the "setup and challenge" step of the experiment, and let correct^i be the event that $b' = b$ in Expt$_i$. Claims 4.4.9, 4.4.12, and 4.4.11 together imply that $SD(\text{trans}^0, \text{trans}^3) \leq 2/5 + O\left(\frac{1}{p^2(n)}\right)$ even conditioned on $\neg$abort, and Claim 4.4.5 shows that $\Pr[\text{abort}] \leq 1/5$. It follows that

$$
\begin{aligned}
\Pr[\mathcal{A} \text{ succeeds}] &= \frac{1}{2} \cdot \Pr[\text{abort}] + \Pr[\text{correct}^i \mid \neg\text{abort}] \cdot \Pr[\neg\text{abort}] \\
&\geq \frac{1}{2} \cdot \frac{1}{5} + \left(1 - \frac{2}{5} - O\left(\frac{1}{p^2(n)}\right)\right) \cdot \frac{4}{5} \\
&= \frac{29}{50} - O\left(\frac{1}{p^2(n)}\right),
\end{aligned}
$$

which is noticeably larger than $1/2$ for n sufficiently large.

4.5   Impossibility for Specific Cases

We now show how we can use Theorem 4.3.2 to rule out black-box constructions of predicate encryption schemes in several specific cases of interest. We begin with the following lemma.

Lemma 4.5.1 *Fix $q = q(n)$, and assume $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$ has the following property: For sufficiently large n, there exist $f_1, \ldots, f_{5q} \in \mathcal{F}_n$ and $I_1, \ldots, I_{5q} \in \mathbb{A}_n$ such that:*

For all $i \in \{1, \ldots, 5q\}$ it holds that $f_i(I_i) = 1$ but $f_j(I_i) = 0$ for $j > i$.

Then $(\mathcal{F}_n, \mathbb{A}_n)_{n \in \mathbb{N}}$ can be q-covered.

If the above holds for every polynomial q, then $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$ is easily covered.

Proof   We show that, under the stated assumption, $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$ satisfies Definition 4.3.1. Fix q and n large enough so that the condition of the lemma holds, and let $f_1, \ldots, f_{5q}$ and $I_1, \ldots, I_{5q}$ be as stated. Define algorithms A_1, A_2, A_3 as follows:

1. $A_1(1^n)$ chooses $i \leftarrow \{0, \ldots, 5q\}$ and outputs $f^* = f_i$.

2. $A_2(1^n, f^*)$ finds i for which $f^* = f_i$ and outputs $I^* = I_i$.

3. $A_3(1^n, f^*)$ finds i for which $f^* = f_i$ and outputs $f_{i+1}, \ldots, f_{5q}$. (If $i = 5q$ then output nothing.)

Note that $A_2(1^n, f^*)$ always outputs I^* with $f^*(I^*) = 1$. We show that for *any* $\mathcal{F}_n$-set system $\{S_f\}_{f \in \mathcal{F}_n}$ over $[q]$, the conditions of Definition 4.3.1 hold. We begin with the following claim:

Claim 4.5.2 *For any $\mathcal{F}_n$-set system $\{S_f\}_{f \in \mathcal{F}_n}$ over $[q]$, there are at most q values $i \in \{1, \ldots, 5q\}$ for which $S_{f_i} \not\subseteq \bigcup_{i < j \leq 5q} S_{f_j}$. (By convention, the union is the empty set if $j = 5q$.)*

Proof   Define $\mathbf{S}_i \overset{\text{def}}{=} \bigcup_{i < j \leq 5q} S_{f_j}$, with $\mathbf{S}_{5q} = \emptyset$. Note that $\mathbf{S}_{i-1} = \mathbf{S}_i \cup S_{f_i}$, and so $S_{f_i} \not\subseteq \bigcup_{i < j \leq 5q} S_{f_j} = \mathbf{S}_i$ iff $\mathbf{S}_i \subsetneq \mathbf{S}_{i-1}$. Since

$$\mathbf{S}_{5q} \subseteq \mathbf{S}_{5q-1} \subseteq \cdots \subseteq \mathbf{S}_1 \subseteq [q],$$

there can be at most q indices i where this occurs.   ∎

75

Fixing an arbitrary $\mathcal{F}_n$-set system $\{S_f\}_{f \in \mathcal{F}_n}$ over $[q]$, let $I \subset \{1, \ldots, 5q\}$ be the set of indices for which $S_{f_i} \subseteq \bigcup_{i < j \leq q} S_{f_j}$; the claim above shows that $|I| \geq 4q$. If A_1 chooses $i \in I$ then:

1. $S_{f^*} = S_{f_i} \subseteq \bigcup_{i < j \leq q} S_{f_j}$.

2. $f_j(I^*) = f_j(I_i) = 0$ for all the predicates $f_{i+1}, \ldots, f_q$ output by A_3.

Since A_1 chooses $i \in I$ with probability $4/5$, this proves the lemma. ∎

We now apply this lemma to several specific cases.

Identity-based encryption. It is easy to see that IBE for identities $\{\mathcal{I}_n\}$ can be viewed as an instance of predicate encryption by setting $\mathbb{A}_n = \mathcal{I}_n$ and $\mathcal{F}_n = \{f_{ID}\}_{ID \in \mathcal{I}_n}$ where

$$f_{ID}(ID') \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } ID' = ID \\ 0 & \text{otherwise} \end{cases}.$$

Let $N = |\mathcal{I}_n|$ denote the size of the identity space. Boneh et al. [22] already rule out black-box constructions of IBE from trapdoor permutations for $N = \omega(\text{poly}(n))$; the next theorem shows that our Theorem 4.3.2 generalizes their result:

Theorem 4.5.3 *There is no black-box construction (from trapdoor permutations or CCA-secure encryption) of an IBE scheme for $5N$ identities where each algorithm makes fewer than N queries to its oracle.*

As a corollary, there is no black-box construction of an IBE scheme (from trapdoor permutations or CCA-secure encryption) for a super-polynomial number of identities.

Proof Let $\mathcal{I}_n = \{ID_1, \ldots, ID_{5N}\}$. It is not hard to see that $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$ can be N-covered: take $f_{ID_1}, \ldots, f_{ID_{5N}}$ and set $I_i = ID_i$ for all i. Then apply Theorem 4.3.2. ■

Forward-secure encryption. In a forward-secure public-key encryption scheme [30] secret keys are associated with time periods; the secret key at time period i enables decryption for ciphertexts encrypted at any time $j \geq i$. (We refer the reader to [30] for further discussion.) A forward-secure encryption scheme supporting $N = N(n)$ time periods can be cast as a predicate encryption scheme by letting $\mathbb{A}_n = \{1, \ldots, N\}$ and $\mathcal{F}_n = \{f_i\}_{1 \leq i \leq N}$ where

$$f_i(j) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } j \geq i \\ 0 & \text{otherwise} \end{cases}.$$

(A forward-secure encryption scheme imposes the additional requirement that $SK_{f_{i+1}}$ can be derived from SK_{f_i}; since we do not impose this requirement our impossibility result is even stronger.) A black-box construction of a forward-secure encryption scheme from any CPA-secure encryption scheme exists for any $N = poly(n)$: the master public key contains public keys $\{pk_1, \ldots, pk_N\}$, and the secret key at period i is $SK_{f_i} = \{sk_i, \ldots, sk_N\}$; encryption at period j uses pk_j. While such a scheme is trivial as far as forward-secure encryption goes (since the public/secret key lengths are linear in N), it satisfies the definition. The next theorem indicates that, in some sense, this trivial construction is almost optimal as far as black-box constructions are concerned; moreover, there is no black-box construction supporting a super-polynomial number of time periods. (In contrast, existing schemes based on specific assumptions [30, 20] support an unbounded number of time periods.)

77

Theorem 4.5.4 *There is no black-box construction (from trapdoor permutations or CCA-secure encryption) of a forward-secure encryption scheme for $5N$ periods where each algorithm in the scheme makes fewer than N queries to its oracle.*

Thus, there is no black-box construction of a forward-secure encryption scheme (from trapdoor permutations or CCA-secure encryption) supporting a super-polynomial number of time periods.

Proof   It is easy to see that $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n\in\mathbb{N}}$ can be N-covered, as taking $f_1, \ldots, f_{5N}$ and setting $I_i = i$ for all i satisfies the conditions of Lemma 4.5.1. Then apply Theorem 4.3.2.

∎

Broadcast encryption. Finally, we look at the case of (public-key) broadcast encryption [42]. Here, there is a fixed public key and a set of users $\mathcal{U} = \{1, \ldots, U\}$ each with their own personal secret key; it should be possible for a sender to encrypt a message in such a way that only some subset $\mathcal{U}' \subset \mathcal{U}$ of users can decrypt. Consider the case where at most $k = k(n) < U$ users are excluded; we refer to this as *k-exclusion broadcast encryption*. This can also be modeled by predicate encryption, if we let $\mathbb{A}_n = \{\mathcal{U}' \subseteq \mathcal{U} \mid |\mathcal{U}'| \geq U - k\}$ and define $\mathcal{F}_n = \{f_i\}_{i\in\mathcal{U}}$ where

$$f_i(\mathcal{U}') \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } i \in \mathcal{U}' \\ 0 & \text{otherwise} \end{cases}.$$

Theorem 4.5.5 *There is no black-box construction (from trapdoor permutations or CCA-secure encryption) of a $(5k)$-exclusion broadcast encryption scheme where each algorithm in the scheme makes k or fewer queries to its oracle.*

Thus, there is no black-box construction of a k-exclusion broadcast encryption scheme (from

trapdoor permutations or CCA-secure encryption) for super-polynomial k.

Proof We show that $\{(\mathcal{F}_n, \mathbb{A}_n)\}_{n \in \mathbb{N}}$ can be k-covered. Take $f_1, \ldots, f_{5k}$ and define

$$I_i \stackrel{\text{def}}{=} \mathcal{U} \setminus \{i, \ldots, 5k\}$$

for $i \in \{1, \ldots, 5k\}$. (So $I_{5k} = \mathcal{U}$.) Note that $|I_i| \geq U - 5k$ always, and these satisfy the

conditions of Lemma 4.5.1. Applying Theorem 4.3.2 concludes the proof. ∎

Chapter 5

Black-Box Constructions of Constant-Round Zero-Knowledge Proofs

5.1   Introduction

In this chapter we present our second black-box separation result. Specifically, we study the round complexity of black-box constructions of black-box zero-knowledge proofs from one-way permutations.

A *zero-knowledge proof* is a protocol wherein one party, the prover, convinces another party, the verifier, of the validity of an assertion while revealing no additional knowledge. Introduced by Goldwasser, Micali and Rackoff in the 1980s [63], zero-knowledge proofs have played a central role in the design and study of cryptographic protocols. In these applications, the main measure of efficiency is the *round complexity* of the proof system, and it is important to construct constant-round zero-knowledge proofs (with negligible soundness) for NP under minimal assumptions. In many cases, a computational zero-knowledge argument system (where both the zero-knowledge and soundness guarantees hold against computationally bounded adversaries) suffices, and we know how to construct such protocols for NP under the minimal assumption of one-way functions [41, 94]. However, in this chapter, we focus on computational zero-knowledge proof systems, where the soundness guarantee must hold against computationally unbounded adversaries.

A common intuition in constructing zero knowledge protocols (typically based on

some form of commitments) is that statistical (resp. computational) soundness corresponds to using a statistically (resp. computationally) binding commitment, while statistical (resp. computational) zero knowledge corresponds to using statistically (computationally) hiding commitments. One might also expect that the round complexity of the resulting zero knowledge protocol is roughly the same as the round complexity of the underlying commitment scheme.

However, the best known construction of computational zero-knowledge proofs from one-way permutations has $\omega(1)$ rounds [61, 17], and the minimal assumption from which we know how to construct constant-round computational zero-knowledge proofs for NP is constant-round statistically *hiding* commitments [57, 110], which seem to be a stronger assumption than one-way permutations [120, 68]. There are no known constructions of constant-round computational zero knowledge proofs from constant-round statistically *binding* commitments. We note that the latter may be constructed from one-way permutations [17] and one-way functions [91, 73]. This raises the following intriguing open problem:

> Can we base constant-round zero-knowledge proofs for NP on the
>
> existence of one-way permutations?

We briefly survey what's known in this regard for constant-round black-box zero-knowledge protocols (that is, those using a black-box simulation strategy). We clarify that while we do know of non-black-box zero-knowledge protocols [5, 67], these protocols are all zero-knowledge arguments and not proofs.

Unconditional Constructions. The only languages currently known to have constant-round zero-knowledge proofs from assumptions weaker than statistically hiding commitment schemes are those that admit statistical zero-knowledge proofs, which do not require any computational assumption at all. Even though this includes languages believed to be outside of BPP such as graph isomorphism and graph non-isomorphism [61, 14], all languages with statistical zero knowledge proofs lie in AM ∩ coAM [1, 46] (and therefore do not include all of NP unless the polynomial hierarchy collapses).

Lower Bounds. Lower bounds for constructions of zero-knowledge protocols were initiated by the work of Goldreich and Oren [62] who showed that 2-round zero-knowledge proofs only exists for languages in BPP. Extending their result, Goldreich and Krawczyk [58] showed that 3-round zero-knowledge proofs and public-coin constant-round zero-knowledge proofs with black-box simulators exist only for languages in BPP. Katz [78] showed that 4-round black-box zero-knowledge proofs only exist for languages in MA ∩ coMA. Haitner et al. [68] ruled out fully black-box constructions of constant-round statistically hiding commitment schemes (in fact, any $O(n/\log n)$-round protocol) from one-way permutations, which means that we are unlikely to obtain constant-round zero-knowledge proofs from one-way permutations via the approach in [57]. More recently, Haitner et al. [72] established a partial converse to [57], namely that any constant-round zero-knowledge proof for NP that remains zero-knowledge under parallel composition implies the existence of constant-round statistically hiding commitments. Unlike the case for stand-alone zero-knowledge, we do not know if there exists a $\omega(1)$-round zero-knowledge proof system for NP that remains zero-knowledge under parallel com-

position, assuming only the existence of one-way permutations. Indeed, zero-knowledge under parallel composition appears to be a qualitively much stronger security guarantee than stand-alone zero-knowledge.

5.1.1 Our Result.

In this chapter, we establish new barriers towards constructing zero-knowledge proof systems from one-way permutations for all of NP:

> **Main Theorem (informal).** Only languages in AM∩coAM admit a fully black-box construction of zero-knowledge proofs starting from one-way permutations where the construction relies on a black-box simulation strategy with constant adaptivity.

As defined in Chapter 3, a fully black-box construction is one that uses the underlying primitive as a black-box. Additionally, any adversary breaking the black-box zero-knowledge of the construction can be used as a black-box to break the security of the underlying primitive. Adaptivity is a measure of how much the black-box simulator relies on responses from previous queries to the cheating verifier in order to generate new queries. We point out that all known constructions of black-box simulators achieve adaptivity that is linear in the round complexity of the protocol and therefore constant adaptivity is a fairly natural restriction for constant-round protocols. Apart from the restriction on adaptivity, this is essentially the best one could hope for in lieu of various positive results mentioned earlier:

- Our result only applies to constant-round protocols – running the $O(\log n)$-fold

parallel repetition of Blum's Hamiltonicity protocol [17] sequentially yields a $\omega(1)$-round black-box zero-knowledge proof system for NP.

- Our result applies only to proofs, but not arguments – there exists a fully black-box construction of constant-round computational zero-knowledge arguments with constant adaptivity from one-way functions for all of NP [41, 106].

- We have unconditional constructions of constant-round statistical black-box zero-knowledge proofs for graph isomorphism and graph non-isomorphism, languages which are in AM ∩ coAM but are commonly believed to lie outside BPP.

Limitations of Our Impossibility Result.   Our impossibility result imposes three main restrictions on the construction: black-box simulation strategy, black-box access to the one-way permutation, and bounded adaptivity of the black-box simulator, amongst which adaptivity appears to be the greatest limitation. Our current ability to prove general lower bounds for zero-knowledge (without limitation to black-box simulation) is relatively limited [62, 8]; moreover, non-black-box simulation strategies so far only yield arguments and not proof systems. In the context of zero-knowledge protocols, there is no indication whether non-black-box access to the underlying primitive has an advantage over black-box access to the primitive.

Extensions to Higher Adaptivity.   The formal statement of our result (Theorem 5.3.4) is slightly more general than stated above and, in particular, allows us to obtain non-trivial consequences even when the simulator's adaptivity is polynomial.

Generalized Main Theorem (informal). If a language L admits a fully black-box construction of zero-knowledge proofs starting from one-way permutations where the construction relies on a black-box simulation strategy with adaptivity t, then both L and $\overline{L}$ have $O(t)$-round public coin interactive proofs where the honest prover strategy can be implemented in $\mathsf{BPP}^{\mathsf{NP}}$.

For the case $t = O(1)$ this is just our main theorem. If we now let L be an NP-complete language, then for $t = O(\log n)$ this implies a collapse in the *quasi-polynomial hierarchy* [107], which one can view as a weakened version of a collapse in the polynomial hierarchy. For $t = o(n)$ this would improve on the best known round complexity for an interactive proof for a coNP-complete language (the best known is linear [89]), and even for $t = \mathrm{poly}(n)$ this would improve on the best known honest prover complexity for an interactive proof for a coNP-complete language (the best known is $\mathcal{P}^{\#\mathcal{P}}$ [89]). We view these results as evidence that such constructions will be hard to find.

5.1.2 Proof Overview

Recall that we start out with a constant-round zero-knowledge proof system $(\mathcal{P}, \mathcal{V})$ with constant adaptivity for a language L and we want to show that L lies in AM ∩ coAM. The high level strategy is to extend the Goldreich-Krawczyk lower bound for constant-round public-coin proofs [58] to the private-coin setting. Following [58] (also [100, 78, 72]), we consider a cheating verifier $\mathcal{V}_{\mathsf{GK}}^*$ that "resamples" new messages that are distributed identically to the real verifier's messages (conditioned upon the partial transcript) every time it is rewound by the simulator. We will need to address the fact that we do not know how to simulate such a $\mathcal{V}_{\mathsf{GK}}^*$ efficiently for general private-coin proofs.

The computational complexity of $\mathcal{V}^*_{\mathsf{GK}}$ comes up in two ways in [58]: first to deduce that the zero-knowledge property holds against such a $\mathcal{V}^*_{\mathsf{GK}}$, and second to derive an efficient AM proof for the underlying language L and its complement $\overline{L}$.

To address the first issue, we rely on a result of Haitner et al. [68], which, roughly speaking, demonstrates the existence of a one-way permutation π secure in the presence of a $\mathcal{V}^*_{\mathsf{GK}}$ oracle (as long as the zero-knowledge proof has bounded round complexity, which is the case here). We will then instantiate the zero-knowledge proof $(\mathcal{P}, \mathcal{V})$ with the permutation π. This will remain zero-knowledge against the cheating verifier $\mathcal{V}^*_{\mathsf{GK}}$ since π is one-way against $\mathcal{V}^*_{\mathsf{GK}}$. Following [58, 78, 72], we may then deduce a $\mathsf{BPP}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}$ algorithm for L. (Such a statement was obtained independently by Pass and Venkitasubramaniam [105].) Along the way, we will exploit (as with [78, 72]) the fact that $(\mathcal{P}, \mathcal{V})$ is a proof system as we need soundness to hold against a cheating prover that is able to simulate $\mathcal{V}^*_{\mathsf{GK}}$.

Next, we will essentially show that $\mathsf{BPP}^{\pi, \mathcal{V}^*_{\mathsf{GK}}} \subseteq \mathsf{AM} \cap \mathsf{coAM}$ from which our main result follows. We do this by constructing a AM proof for L and $\overline{L}$. The strategy is to have the AM prover (Merlin) and verifier (Arthur) jointly simulate π and $\mathcal{V}^*_{\mathsf{GK}}$. In more detail, Arthut will pick the permutation π at random from a space of $\mathsf{poly}(T^m)$ permutations, where T is an upper bound on the running time of the reduction in the zero-knowledge proof and m is the round complexity of the proof; this turns out to suffice as a one-way permutation for the result in [68].[1] Next, we will have Arthur and Merlin jointly simulate each oracle computation of $\mathcal{V}^*_{\mathsf{GK}}$ using a (constant-round public-coin) random sampling protocol from [71]. Note that naively having Merlin perform the computation

---

[1]Having Arthur sample a random permutation "on the fly" does not work because the permutation π needs to be defined everywhere for $\mathcal{V}^*_{\mathsf{GK}}$ to be well-defined.

of $\mathcal{V}_{\mathsf{GK}}^*$ fails for two reasons: a cheating Merlin may resample messages from a distribution different from the uniform distribution, and may not answer all of the $\mathcal{V}_{\mathsf{GK}}^*$ queries "independently". Finally, we rely on the constant adaptivity requirement of $(\mathcal{P}, \mathcal{V})$ to to guarantee that the final proof for $\overline{L}$ has constant round complexity.

As mentioned previously, in a recent work, Pass et al. [105] independently obtained results similar to ours. They also show that any language L for which there exists a fully black-box construction of constant-round zero-knowledge proofs from one-way functions is in $\mathsf{BPP}^{\pi, \mathcal{V}_{\mathsf{GK}}^*}$. Their techniques for doing this are different from ours. They use a generic transformation from private-coin proofs into $\mathcal{V}_{\mathsf{GK}}^*$-relativized public-coin proofs, upon which the result then follows from the (relativized) lower bound for constant-round public-coin proofs in [58]. They then argue that if such proofs exist for all of NP, this would imply unlikely properties for the complexity class $\mathsf{BPP}^{\pi, \mathcal{V}_{\mathsf{GK}}^*}$. Our techniques, on the other hand, allow us to relate the existence of such proofs to old questions in complexity such as whether $\mathsf{NP} \subseteq \mathrm{coAM}$ or whether coNP has interactive proofs with a $\mathsf{BPP}^{\mathsf{NP}}$ prover, whereas $\mathsf{BPP}^{\pi, \mathcal{V}_{\mathsf{GK}}^*}$ is a new and less well-understood notion.

5.2 Preliminaries

5.2.1 Basic Definitions

We need the following definitions due to [49].

Definition 5.2.1 *A permutation $\pi : \{0,1\}^n \to \{0,1\}^n$ is T-hard if for any circuit C of size at most T, and for y chosen uniformly at random, $\Pr[C(y) = \pi^{-1}(y)] \leq \frac{1}{T}$, where the probability is taken over the choice of y. If, given x, $\pi(x)$ is also efficiently computable then we call such a*

permutation a one way permutation *(OWP).*

Definition 5.2.2 *Let* Π_n *be the set of all permutations from* $\{0,1\}^n \rightarrow \{0,1\}^n$. *Then, using the notation of [49], we define* $\Pi_{k,n} \subseteq \Pi_n$ *as* $\{\pi_{k,n} \mid \pi_{k,n}(a,b) = (\pi_k(a),b) \text{ for some } \pi_k \in \Pi_k\}$ *In other words, a uniform element of* $\Pi_{k,n}$ *is a random permutation on the first* k *bits, and fixes the last* $n-k$ *bits.*

5.2.2 Complexity Classes.

We now review the definitions of several complexity classes that are used in this chapter. We let BPP denote the class of languages that can be recognized by a ppt Turing machine. Formally,

Definition 5.2.3 *A language* L *is in* BPP *if there exists a* ppt *Turing machine* M *such that:*

- *For every* $x \in L$ *it holds that* $\Pr[M(x) = 1] \geq \frac{2}{3}$.

- *For every* $x \notin L$ *it holds that* $\Pr[M(x) = 1] < \frac{1}{3}$.

For any oracle $\mathcal{O}$, we let BPP$^{\mathcal{O}[k]}$ denote the class of languages that are decidable by efficient randomized algorithms using at most k rounds of adaptive queries to an oracle $\mathcal{O}$. One round of adaptivity is a set of queries $x_1, \ldots, x_k$ the algorithm asks to the oracle such that the x_i can only depend on oracle answers to queries asked in previous rounds.

Additionally, we recall the definition of Arthur-Merlin (public-coin) proofs [3]. We let AM$[k]$ denote the class of languages that have $O(k)$-round public-coin interactive proofs (recall that public-coins are equivalent to private coins by [64] in this setting). Namely:

88

Definition 5.2.4 *$L \in$ AM$[k]$ if there is a $O(k)$-round public-coin interactive proof between an efficient verifier V and an all-powerful prover P such that:*

- *(Perfect Completeness:) For all $x \in L$, V always accepts when interacting with P.*

- *(Negligible Soundness Error:) For all $x \notin L$ and all possibly cheating prover strategies P^*, V accepts when interacting with P^* with only negligible probability.*

We note that, this definition is equivalent to one only requiring an inverse polynomial gap between completeness and soundness error.

We let AM $\stackrel{\text{def}}{=}$ AM$[O(1)]$. Additionally, we let MA denote the class defined similarly for 1-round proofs (sent from P to V). We say that a protocol (P, V) has an *honest prover strategy* of complexity $\mathcal{C}$ if the prover algorithm can be implemented by a machine in the class $\mathcal{C}$. We recall that coNP is in AM$[n]$ (where n is the length of the instance) with an honest prover strategy complexity of $\mathcal{P}^{\#\mathcal{P}}$ [89], and it is an open question whether the round complexity or the honest prover strategy complexity can be improved. From here on, we will call the prover in an AM proof Merlin and the verifier Arthur.

5.2.3 Zero-Knowledge

We have previously given a definition of black-box computational zero-knowledge proofs in Definition 2.3.8. Here we give a special case of this definition that we will use in this chapter. Specifically, we define a fully black-box construction of *weak* computational zero knowledge (wCZK) from one way permutations. As usual, we let negl(n) be a negligible function.

Notation: we will use the following notation for interactive protocols. For any interactive

protocol between a prover P and a verifier V, we let $2m$ denote the total number of rounds of communication, where a round consists of one message, either from P to V or from V to P. We let α_i denote the ith message sent from P to V, and β_i the ith response from V to P. Note that α_i is sent in round $2i-1$ and β_i is sent in round $2i$. Also, having P always send the first message is without loss of generality as we can set $\alpha_1 = \perp$ to model a proof where V goes first. For $i \in \{1\dots,m\}$, we let $\alpha_{[i]} = (\alpha_1, \dots, \alpha_i)$. Let $V = (V_1, \dots V_m)$ be the decomposition of V into its next-message functions. Here $V_i(x, \alpha_{[i]}, \omega)$ outputs β_i, the ith message sent by V when using input x, random coins ω, and receiving messages $\alpha_{[i]}$ from P. Let $\langle P, V \rangle(x)$ denote the verifier's view of an execution of the interactive protocol on an input x. This view includes all messages $\alpha_{[m]}$ sent by the prover, the verifier's random coins ω, and (if V is allowed access to an oracle) the answers to any oracle queries V may have made. We say that $\langle P, V \rangle(x)$ accepts if $V_m(x, \alpha_{[m]}, \omega) = 1$. We will use calligraphic $\mathcal{P}, \mathcal{V}, \mathcal{S}$ to denote the prover, verifier, and simulator in a zero-knowledge protocol.

Definition 5.2.5 *A fully black-box construction of a (weak) computational zero-knowledge proof system from one-way permutations for a language L is a tuple of oracle procedures $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$ such that there exists a polynomial $T(n)$ satisfying the following properties for every family of permutations $\pi = \{\pi_n\}_{n \geq 1}$:*

Efficiency. *The running times of $\mathcal{V}, \mathcal{S}, M$ are bounded by $T = T(n)$.*

Completeness. *For all $x \in L$:* $\Pr[\langle \mathcal{P}^\pi, \mathcal{V}^\pi \rangle(x) \text{ accepts}] \geq 1 - \mathsf{negl}(n)$.

Soundness. *For all $x \notin L$ and for all (possibly computationally unbounded) $\mathcal{P}^*$,*

$$\Pr[\langle \mathcal{P}^*, \mathcal{V}^\pi \rangle(x) \text{ accepts}] \leq \mathsf{negl}(n).$$

Black-Box Zero-Knowledge. *For all (possibly unbounded) $\mathcal{V}^*, D$ and for all $x \in L$:*

if

$$\left| \Pr[D(\langle \mathcal{P}^\pi, \mathcal{V}^* \rangle(x)) = 1] - \Pr[D(\mathcal{S}^{\pi, \mathcal{V}^*}(x)) = 1] \right| > 1/n$$

then M can invert π, namely:

$$\Pr_{y \leftarrow \{0,1\}^n}[M^{\pi, \mathcal{V}^*, D}(y) = \pi^{-1}(y)] > 1/T$$

We note that completeness and soundness hold even if the given permutations are not one-way. Also, $\mathcal{V}^*, D$ are quantified after π is fixed and therefore may depend on π.

Comparison with standard definitions of zero-knowledge: The property that makes the above definition *weak* zero knowledge is that we only require the distinguishing advantage to be smaller than $1/n$, rather than negligible (the choice of $1/n$ was arbitrary; any non-negligible function will do). This enables us to consider simulators that run in *strict* polynomial time; it is known that in the standard definition of zero knowledge where the distinguishing advantage is negligible, no strict polynomial-time black-box simulators exist for constant-round protocols [7], although there are examples of non-black-box simulators [5]. It is useful for us to consider strict polynomial-time simulators

because defining adaptivity is more straight-forward for such simulators than for expected polynomial-time simulators. This is discussed in the next section.

Nevertheless, we note here that any zero knowledge proof satisfying the standard definition also satisfies the weak definition above: if a simulator $\mathcal{S}'$ satisfies the standard definition and runs in expected time T', then a simulator $\mathcal{S}$ satisfies the weak definition by running $\mathcal{S}'$ for at most $2nT'$ steps, and halting with a failure symbol if $\mathcal{S}'$ does not produce an output in that time. This is true since, by Markov's inequality, the probability that $\mathcal{S}'$ runs for more than $2nT'$ steps is bounded by $\frac{1}{2n}$. Thus, by ruling out black-box constructions of weak zero knowledge proofs from one-way permutations, we also rule out proofs satisfying the standard definition. We note that the same discussion applies to the runtime of the reduction algorithm M.

Simplifying assumptions: we assume for simplicity that on inputs of length n, $\mathcal{V}$ and $\mathcal{S}$ only query π on inputs of length n. We assume that in an honest interaction of the protocol, the last message is from the verifier $\mathcal{V}$ to the prover $\mathcal{P}$ and contains the verifier's random coins. Clearly this does not affect either zero knowledge or soundness since it occurs after all "meaningful" messages are sent. This assumption allows us to define a transcript to be accepting if the honest verifier would accept that transcript using the coins output in the last message, and this definition remains meaningful even for transcripts generated by cheating verifiers. We assume without loss of generality that the simulator $\mathcal{S}$ never asks the same query twice and that it only asks *refinement* queries. Namely, for $i > 1$ and for every query $\alpha_{[i]} = (\alpha_{[i-1]}, \alpha_i)$ that the simulator queries to its cheating verifier black box $\mathcal{V}^*$, it must have previously queried $\alpha_{[i-1]}$ as well. We direct the reader to [57] for a discussion of why this holds without loss of generality.

5.2.4 Adaptivity

Here we define the *adaptivity* of the simulator, namely how much it uses responses from previous queries to the verifier black-box in order to generate new queries. All of the black-box simulators for constant-round zero knowledge in the literature intuitively work the following way: repeatedly query the cheating verifier with dummy queries enough times until it leaks some secret, then rewind and use this secret to output a simulated transcript [57, 13, 27, 41, 110]. The simulator may use the verifier's answers to determine whether to continue with dummy queries or to proceed to the next step of the simulation. If the simulator runs in *expected polynomial time* (rather than strict polynomial time), this procedure lasts indefinitely, making it hard to define the degree of the simulator's adaptivity. This is why we choose to work with *weak* zero knowledge, where the simulation is strict polynomial time; the definition of adaptivity becomes much simpler and more intuitive in this setting. We stress again that this only strengthens our result, as any zero-knowledge proof system satisfying the standard definition also satisfies the weak definition.

Definition 5.2.6 *A simulator $\mathcal{S}$ running in time T is said to be t-adaptive if it can be decomposed into $t + 1$ oracle machines $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_t, \mathcal{S}_{t+1})$ with the following structure. Let x, ω (respectively) be the input and random coins for $\mathcal{S}$. For all permutations π and all cheating verifiers $\mathcal{V}^*$, $\mathcal{S}^{\pi, \mathcal{V}^*}$ operates as follows:*

1. *$\mathcal{S}_1^{\pi, \mathcal{V}^*}(x; \omega)$ generates at most T queries $q_1^{(1)}, \ldots, q_T^{(1)}$ using x, ω. It sends these queries to $\mathcal{V}^*$ and gets back answers $\vec{a}_1 = (a_1^{(1)}, \ldots, a_T^{(1)})$.*

2. *For each phase $j, 1 < j \leq t$, $\mathcal{S}_j^{\pi, \mathcal{V}^*}(x; \omega, \vec{a}_{j-1})$ generates at most T queries $q_1^{(j)}, \ldots, q_T^{(j)}$*

*using x, ω and $\vec{a}_{j-1}$ which is the concatenation of all oracle answers from phases $1, \ldots, j -$
1. $\mathcal{S}_j^{\pi, \mathcal{V}^*}$ sets $\vec{a}_j$ to be the oracle answers $a_1^{(j)}, \ldots, a_T^{(j)}$ for the j'th phase, concatenated with
$\vec{a}_{j-1}$.*

3. *After obtaining $\vec{a}_t$, $\mathcal{S}_{t+1}^{\pi}(x; \omega, \vec{a}_t)$ computes the final output (notice it does so without call-
ing $\mathcal{V}^*$).*

5.2.5 The Sam Oracle

In our separation, we make extensive use of the Sam oracle defined in [68]. Here we provide a brief description of this oracle. A more formal description can be found in [68].

Description of Sam_d: Sam_d takes as input a query $q = (i, C_{\mathsf{next}}, C_{\mathsf{prev}}, z)$ and outputs (ω', z'), such that:

1. ω' is chosen uniformly at random from:

   - the domain of C_{next} if $i = 1$.

   - the set $\{\omega \mid C_{\mathsf{prev}}(\omega) = z\}$ if $i > 1$.

2. $z' = C_{\mathsf{next}}(\omega')$.

   The inputs to Sam_d are subject to the following restrictions:

1. The root query in every tree must include a security parameter 1^n such that $d = d(n)$ is the maximum depth query.

2. Queries with $i > d$ receive output $\perp$.

94

3. If $i > 1$, then the input $(i - 1, C_{\mathsf{prev}}, \cdot, \cdot)$ was previously queried and resulted in output (ω, z) for some ω. Note that this restriction imposes a forest structure on the queries.

4. C_{next} is a *refinement* of C_{prev}. Formally: $C_{\mathsf{next}} = (C_{\mathsf{prev}}, \widetilde{C})$ for some circuit $\widetilde{C}$.

For our purposes, it is easier to think of Sam_d as being stateful, in which case the above restrictions can easily be implemented. Technically however Sam_d must be stateless, and so the above restrictions are enforced in [68] by giving Sam_d access to a signature protocol, and having him sign the output to every query, as well as the depth of the query, before returning a response. New queries are required to include a signature on a prior query, demonstrating that the first and third requirements have been met. (The refinement property can be verified by Sam_d independently.) Any query not meeting these restrictions receives output $\perp$. We direct the reader to [68] for the complete details (see also [71] for a precise statement about how to remove state), and we work with a stateful Sam_d for the remainder of this paper.

We will also consider Sam_d in a relativized world with a random permutation $\pi = \{\pi_n\}_{n \in \mathbb{N}}$, where $\pi_n : \{0, 1\}^n \to \{0, 1\}^n$ is chosen at random from all permutations on $\{0, 1\}^n$. We let Sam_d^π denote Sam_d in this relativized world. Sam_d^π is defined exactly as Sam_d, except it accepts circuits $C_{\mathsf{prev}}^\pi, C_{\mathsf{next}}^\pi$ that can possibly contain π gates to represent queries to π.

We will abuse notation and write Sam to denote Sam_d for some $d = O(1)$. Our results will apply to all constant d so this slight abuse does not affect the correctness of our statements. Next, we state a theorem that will be very useful to us showing that Sam

can be simulated by a public-coin proof.

Using Merlin to Simulate Sam**:**

Let $\mathsf{BPP}^{\mathsf{Sam}[t]}$ denote the class of languages that can be decided efficiently by a machine making at most t adaptive rounds of queries to the oracle Sam. We use the following theorem from [71] which shows that one can simulate this Sam oracle by a constant-round public-coin protocol.

Theorem 5.2.7 ([71]) *For any $L \in \mathsf{BPP}^{\mathsf{Sam}[t]}$, it holds that both L and $\overline{L}$ have $\mathsf{AM}[t]$ proofs with an honest prover strategy complexity of $\mathsf{BPP}^{\mathsf{NP}}$.*

5.3 Proof of Main Theorem

5.3.1 Overview

As discussed in the Introduction, our proof involves using a particular cheating verifier, $\mathcal{V}^*_{\mathsf{GK}}$ defined in Section 5.3.2, with the following properties:

- $\mathcal{V}^*_{\mathsf{GK}}$ cannot invert a random permutation π. By definition 5.2.5, this implies that the view $\langle \mathcal{P}^\pi, \mathcal{V}^*_{\mathsf{GK}} \rangle(x)$ can be simulated by a simulator $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(x)$ whenever $x \in L$. (Section 5.3.3)

- The simulator $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(x)$ cannot produce an accepting transcript whenever $x \notin L$. Together with the previous property, this gives a way of deciding L. (Section 5.3.3)

- One can efficiently generate a transcript according to $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(x)$ in a constant number of rounds with the help of an all-powerful (but possibly cheating) prover Merlin. Since, using the output of $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(x)$, one can efficiently decide whether or not

96

$x \in L$, this implies $L \in \mathsf{AM} \cap \mathsf{coAM}$. (Section 5.3.4)

5.3.2 Defining $\mathcal{V}^*_{\mathsf{GK}}$

Our cheating verifier $\mathcal{V}^*_{\mathsf{GK}}$ is an extension of the one proposed by Goldreich and Krawczyk [58]. Informally, upon receiving a message, this cheating verifier uniformly chooses a new random tape consistent with the transcript seen so far, and uses this to compute his next message. The formal definition follows, using notation defined in Section 5.2.1.

Fix any black-box construction of weak zero knowledge from one-way permutations $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$. Let $\omega \in \{0,1\}^T$ be a random tape for the honest verifier $\mathcal{V}$ which is divided into next-message functions $\mathcal{V}_1, \ldots, \mathcal{V}_m$. Define

$$R_\omega^{\alpha_{[i]}} = \{\omega' \in \{0,1\}^T \mid \forall j < i, \ \mathcal{V}_j(x, \alpha_{[j]}; \omega) = \mathcal{V}_j(x, \alpha_{[j]}; \omega')\}$$

i.e. the set of random tapes that, given prover messages $\alpha_{[i]}$, produce the same verifier messages as the random tape ω. For the special case where $i = 1$, set $R_\omega^{\alpha_1} = \{0,1\}^T$ for all α_1 and all ω.

Define $\mathcal{V}_{[i]} = (\mathcal{V}_1, \ldots, \mathcal{V}_i)$ to be the circuit that outputs the concatenation of $\mathcal{V}_1, \ldots, \mathcal{V}_i$. Namely, for every $\alpha_{[i]}$ and ω, it holds that

$$\mathcal{V}_{[i]}(\alpha_{[i]}, \omega) = (\mathcal{V}_1(\alpha_1, \omega), \mathcal{V}_2(\alpha_{[2]}, \omega), \ldots, \mathcal{V}_i(\alpha_{[i]}, \omega))$$

For any $\alpha_{[i]}$, let $\mathcal{V}_{[i]}(\alpha_{[i]}, \cdot)$ denote the circuit $\mathcal{V}_{[i]}$ with the input $\alpha_{[i]}$ hard-wired (therefore it takes only input ω.

Definition 5.3.1 *The cheating verifier* $\mathcal{V}^*_{\mathsf{GK}} = (\mathcal{V}^*_{\mathsf{GK},1}, \ldots, \mathcal{V}^*_{\mathsf{GK},m})$ *is defined using the* Sam^π_m

oracle and a look-up table that associates server queries $\alpha_{[i]}$ *with* Sam^π_m *oracle responses* (ω, z).

We write $\mathcal{V}^*_{\mathsf{GK}}$ *with the understanding that the input* x *is hardwired into the verifier and the*

verifier is allowed oracle access to the permutation π *and* Sam^π_m. *Additionally, we write* $\mathcal{V}_i(\alpha_{[i]}, \cdot)$

to indicate the circuit $\mathcal{V}_i(x, \alpha_{[i]}; \omega)$ *outputting the verifier's messages* $\beta_{[i]}$ *on input* ω *with the*

values x *and* $\alpha_{[i]}$ *fixed.*

- $\mathcal{V}^*_{\mathsf{GK},1}(\alpha_1)$: *invoke* $\mathsf{Sam}^\pi_m(1, \mathcal{V}_1(\alpha_1, \cdot), \bot, \bot)$ *and let* (ω_1, β_1) *be the response. (Here, the* $\bot$

  inputs are placeholders and can be replaced by anything.) Store $(\alpha_1, \omega_1, \beta_1)$ *in the look-up*

  table and output β_1.

- $\mathcal{V}^*_{\mathsf{GK},i}(\alpha_{[i]})$ *for* $i > 1$: *let* $\alpha_{[i]} = (\alpha_{[i-1]}, \alpha_i)$. *Look up the value* $(\alpha_{[i-1]}, \omega_{i-1}, \beta_{[i-1]})$ *stored*

  during a previous query. Query

$$\mathsf{Sam}^\pi_m(i, \ \mathcal{V}_{[i]}(\alpha_{[i]}, \cdot), \ \mathcal{V}_{[i-1]}(\alpha_{[i-1]}, \cdot), \ \beta_{[i-1]})$$

  and let $(\omega_i, \beta_{[i]})$ *be the response. Store* $(\alpha_{[i]}, \omega_i, \beta_{[i]})$ *in the look-up table and output* β_i.

Observe that querying Sam^π_m in the manner described above for the case $i > 1$

returns an ω_i that is distributed uniformly in $R^{\alpha_{[i]}}_{\omega_{i-1}}$.

Recall that we assume the simulator never repeats queries and only makes refine-

ment queries. Therefore, $\mathcal{V}^*_{\mathsf{GK}}$ never tries to store inconsistent entries in the table, and

$\mathcal{V}^*_{\mathsf{GK}}$ never queries its table for entries that do not exist. Therefore, $\mathcal{V}^*_{\mathsf{GK}}$'s queries to

Sam^π_m always satisfy the restrictions laid out in Section 5.2.5. Observe that the output

of $\langle \mathcal{P}^\pi, \mathcal{V}^*_{\mathsf{GK}} \rangle(x)$ is distributed identically to the honest $\langle \mathcal{P}^\pi, \mathcal{V}^\pi \rangle(x)$. We note that $\mathcal{V}^*_{\mathsf{GK}}$ is

not efficient as there may be no way to sample from $R_\omega^{\alpha[i]}$ efficiently. However, we show in Section 5.3.4 how to simulate $\mathcal{V}_{\mathsf{GK}}^*$ using an Arthur-Merlin proof.

To complete the description of $\mathcal{V}_{\mathsf{GK}}^*$ we also need to construct a one-way permutation that remains one-way in the presence of a $\mathcal{V}_{\mathsf{GK}}^*$-oracle. To accomplish this, we refer to a result of Haitner et al. [68], which ruled out fully black-box constructions of $o(n/\log n)$-round statistically hiding commitment schemes form one-way permutations (where n is the security parameter). Building on and generalizing the works of [49, 117, 120], they demonstrated that by choosing π from $\Pi_{k,n}$ for appropriate k, π remains one-way even in the presence of a Sam_m^π-oracle.

Formally, the following lemma follows directly from their results.

Lemma 5.3.2 (implicit in [68]) *Suppose T, k satisfy $T^{3m+2} < 2^{k/8}$. Then, for any oracle machine A running in time T, it holds that:*

$$\Pr_{\pi \leftarrow \Pi_{k,n}, \, y \leftarrow \{0,1\}^n}[A^{\pi, \mathcal{V}_{\mathsf{GK}}^*}(y) = \pi^{-1}(y)] \leq 1/T$$

Proof   This follows from [68, Theorem 5.1], which established the above statement where $\mathcal{V}_{\mathsf{GK}}^*$ is replaced by Sam_m^π. From our definition of $\mathcal{V}_{\mathsf{GK}}^*$, it is clear that one call to $\mathcal{V}_{\mathsf{GK}}^*$ can be implemented using one call to Sam_m^π Furthermore, as noted above, since we assume $\mathcal{S}$ only makes unique refinement queries, all of the queries that $\mathcal{V}_{\mathsf{GK}}^*$ asks of Sam_m^π satisfy the restrictions in the definition of Sam_m^π. ∎

5.3.3 Deciding L Using $\mathcal{V}_{\mathsf{GK}}^*$

We show that any language L admitting a fully black-box constructions of a weak computational zero-knowledge proof from one-way permutations can be decided in $\mathsf{BPP}^{\pi,\mathcal{V}_{\mathsf{GK}}^*}$. Specifically, the following lemma shows that $\mathcal{S}^{\pi,\mathcal{V}_{\mathsf{GK}}^*}(x)$ generates an accepting transcript with high probability if and only if $x \in L$.

Lemma 5.3.3 *Given any fully black-box construction from one-way permutations of a constant-round weak zero knowledge proof $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$ for a language L, and any n, k satisfying $T^{3m+2} < 2^{k/16}$, where $2m = O(1)$ is the round complexity of the proof system and $T = poly(n)$ is the strict polynomial bound on the running times of $\mathcal{V}, \mathcal{S}, M$, the following hold:*

1. *If $x \in L$, then $\Pr_{\pi \leftarrow \Pi_{k,n}, \mathcal{S}, \mathcal{V}_{\mathsf{GK}}^*}[\mathcal{S}^{\pi,\mathcal{V}_{\mathsf{GK}}^*}$ generates accepting transcript$] \geq 2/3$.*

2. *If $x \notin L$, then $\Pr_{\pi \leftarrow \Pi_{k,n}, \mathcal{S}, \mathcal{V}_{\mathsf{GK}}^*}[\mathcal{S}^{\pi,\mathcal{V}_{\mathsf{GK}}^*}$ generates accepting transcript$] < 1/3$.*

Proof

Yes instances: We use the zero-knowledge property of the proof system to prove that for all $x \in L$:

$$\Pr[\mathcal{S}^{\pi,\mathcal{V}_{\mathsf{GK}}^*}(x) \text{ outputs an accepting transcript}] \geq 2/3 \tag{3.1}$$

The proof proceeds by contradiction, showing that if $\mathcal{S}$ fails to output an accepting transcript with sufficiently high probability then, by the weak zero-knowledge property of $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$, M can invert a random permutation $\pi \in \Pi_{k,n}$.

As was noted before, the distributions $\langle \mathcal{P}^\pi, \mathcal{V}_{\mathsf{GK}}^* \rangle(x) = \langle \mathcal{P}^\pi, \mathcal{V}^\pi \rangle(x)$. Therefore, by the completeness of the proof system, for $x \in L$, the transcript $\langle \mathcal{P}^\pi, \mathcal{V}_{\mathsf{GK}}^* \rangle(x)$ is accepted by the honest verifier with probability $1 - \mathsf{negl}(n)$. More formally, $\Pr[\mathcal{V}_m^\pi(x, \langle \mathcal{P}^\pi, \mathcal{V}_{\mathsf{GK}}^* \rangle(x)) = $

$1] \geq 1 - \mathsf{negl}(n)$.

For the sake of contradiction, assume that $\mathcal{S}^{\pi,\mathcal{V}_{\mathsf{GK}}^*}(x)$ outputs an accepting transcript with probability less than $2/3$. That is, $\Pr[\mathcal{V}_m^\pi(x, \mathcal{S}^{\pi,\mathcal{V}_{\mathsf{GK}}^*}(x)) = 1] < 2/3$. Then we can use the honest verifier $\mathcal{V}$ to distinguish between the prover and simulator output, since

$$| \Pr[\mathcal{V}_m^\pi(x, \langle \mathcal{P}^\pi, \mathcal{V}_{\mathsf{GK}}^* \rangle) = 1] - \Pr[\mathcal{V}_m^\pi(x, \mathcal{S}^{\pi,\mathcal{V}_{\mathsf{GK}}^*}(x)) = 1]| > 1/3 - \mathsf{negl}(n).$$

Therefore, by the weak black-box zero-knowledge property of $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$, there exists an oracle machine $M^{\pi,\mathcal{V}_{\mathsf{GK}}^*,\mathcal{V}}$ running in time T that can break the one-wayness of π with probability at least $1/T$. We can remove oracle access to $\mathcal{V}$ by having M simulate $\mathcal{V}$ internally, making at most T oracle calls to π for each call to $\mathcal{V}$. Thus, we get a machine $M^{\pi,\mathcal{V}_{\mathsf{GK}}^*}$ running in time T^2 such that

$$\Pr_{\pi \leftarrow \Pi_{k,n},\ y \leftarrow U_n} [M^{\pi,\mathcal{V}_{\mathsf{GK}}^*}(y) = \pi^{-1}(y)] \geq 1/T > 1/T^2.$$

This yields a contradiction to Lemma 5.3.2, and Equation (3.1) follows.

No instances: Here, we use statistical soundness (following [72, 78, 58]) to argue that for all $x \notin L$:

$$\Pr[\mathcal{S}^{\pi,\mathcal{V}_{\mathsf{GK}}^*}(x) \text{ outputs an accepting transcript}] < 1/3 \tag{3.2}$$

The proof proceeds by contradiction, showing that if $\mathcal{S}$ outputs an accepting transcript with high probability, then there exists a (computationally unbounded) cheating prover $\mathcal{P}_{\mathsf{GK}}^*$ that breaks the statistical soundness of the proof system. Let T, the running time of $\mathcal{S}$, be the bound on the total number of $\mathcal{V}_{\mathsf{GK}}^*$ queries made by $\mathcal{S}$, and let $m = O(1)$

be the round complexity of the zero knowledge proof system. Starting from $\mathcal{V}^*_{\mathsf{GK}}$, we define a new (inefficient) prover strategy $\mathcal{P}^*_{\mathsf{GK}}$ which interacts with an external verifier $\mathcal{V}$ as follows:

1. Choose queries to forward to $\mathcal{V}$: On input x, $\mathcal{P}^*_{\mathsf{GK}}$ picks a random subset of query indices $U = \{j_1, j_2, \ldots, j_m\} \subset [T]$ of size m. The set U represents the queries that $\mathcal{P}^*_{\mathsf{GK}}$ will forward to the verifier $\mathcal{V}$.

2. Simulate $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(x)$: Internally simulate $\mathcal{S}^{\pi, \mathcal{V}^*_{\mathsf{GK}}}(x)$ step by step. We handle the j'th oracle query, q_j, that $\mathcal{S}$ makes to $\mathcal{V}^*_{\mathsf{GK}}$ as follows. Let $q_j = \alpha_{[i]}$ for some $i \leq m$.

   - If $j \notin U$: Simulate $\mathcal{V}^*_{\mathsf{GK}}$ internally to answer q_j. More formally, look up the value $(\alpha_{[i-1]}, \omega)$ stored during a previous $\mathcal{V}^*_{\mathsf{GK}}$ query. (Note that since $\mathcal{S}$ only makes refinement queries, $\mathcal{S}$ must have made such a query.) Choose $\omega' \leftarrow R^{\alpha_{[i]}}_\omega$ uniformly at random ($\mathcal{P}^*_{\mathsf{GK}}$ can do this since he is computationally unbounded), store $(\alpha_{[i]}, \omega')$ and output $\mathcal{V}_i(x, \alpha_{[i]}, \omega')$.

   - If $j \in U$: If $q_j = \alpha_{[i]}$ and $i > 1$, forward α_i to the external $\mathcal{V}$. Upon receiving β_i in response, look up the stored value $(\alpha_{[i-1]}, \omega)$ and uniformly sample a random string $\omega'' \leftarrow \{\omega' \in R^{\alpha_{[i]}}_\omega \wedge \mathcal{V}_i(x, \alpha_{[i]}, \omega') = \beta_i\}$. Store $(\alpha_{[i]}, \omega'')$ and output β_i.

Note that as long as $\mathcal{S}$ outputs an accepting transcript with noticeable probability when interacting with $\mathcal{V}^*_{\mathsf{GK}}$ on $x \notin L$ then this cheating prover $\mathcal{P}^*_{\mathsf{GK}}$ has a noticeable probability of outputting an accepting transcript when interacting with the honest verifier $\mathcal{V}$. This happens if $\mathcal{P}^*_{\mathsf{GK}}$ chooses U to include exactly the messages that are used by $\mathcal{S}$ in his output transcript. $\mathcal{P}^*_{\mathsf{GK}}$ succeeds in choosing the correct queries with probability at least $1/T^{O(m)}$.

Thus, if $\mathcal{S}$ outputs an accepting transcript with probability $\geq 1/3$, then $\mathcal{P}_{\mathsf{GK}}^*$ outputs an accepting transcript with probability at least $1/3 \cdot 1/T^{O(m)}$ which is non-negligible when $m = O(1)$. This is a contradiction of the fact that the proof has negligible soundness error, thus Equation (3.2) follows.

∎

5.3.4   Applying Theorem 5.2.7 To Remove $\mathcal{V}_{\mathsf{GK}}^*$

We can now combine Lemma 5.3.3 and Theorem 5.2.7 to prove our main theorem.

Theorem 5.3.4 (Main Theorem) *Suppose there is a black-box construction from a one-way permutation of a constant-round weak zero knowledge proof $(\mathcal{P}, \mathcal{V}, \mathcal{S}, M)$ for a language L, where $\mathcal{S}$ is t-adaptive. Then both L and $\overline{L}$ are in $\mathsf{AM}[t]$ with honest prover complexity $\mathsf{BPP}^{\mathsf{NP}}$.*

Proof   From Lemma 5.3.3 we already know that $\mathcal{S}^{\pi, \mathcal{V}_{\mathsf{GK}}^*}$ decides L. We will construct an oracle algorithm A based on $\mathcal{S}$, such that A^{Sam} decides L and furthermore the adaptivity of A is the same as the adaptivity of $\mathcal{S}$.

Sampling π Efficiently: By Lemma 5.3.2, we know that for π to be one-way in the presence of $\mathcal{V}_{\mathsf{GK}}^*$, it is sufficient to choose $\pi \leftarrow \Pi_{k,n}$ with $k = 9(3m + 2) \log T = O(\log n)$. Such a permutation can be sampled in polynomial time by sampling a uniform permutation on $k = O(\log n)$ bits. Let $A_1^{\mathcal{V}_{\mathsf{GK}}^*}$ be identical to $\mathcal{S}^{\pi, \mathcal{V}_{\mathsf{GK}}^*}$, except A_1 first samples π by itself and then runs $\mathcal{S}^{\pi, \mathcal{V}_{\mathsf{GK}}^*}$.

From Definition 5.3.1, it holds that oracle access to $\mathcal{V}_{\mathsf{GK}}^*$ can be implemented using oracle access to Sam and an additional look-up table to associate previous queries with previous oracle responses. Therefore, we can construct a polynomial time oracle machine

A such that A^{Sam} behaves identically to $A_1^{\mathcal{V}_{\mathsf{GK}}^*}$ using the Sam oracle to simulate the $\mathcal{V}_{\mathsf{GK}}^*$ oracle for A_1. Furthermore, the adaptivity of A is identical to the adaptivity of A_1, whose adaptivity in turn is the same as that of $\mathcal{S}$.

Since $\mathcal{S}$ has adaptivity t, this implies that $L \in \mathsf{BPP}^{\mathsf{Sam}[t]}$. We can therefore apply Theorem 5.2.7 to conclude the proof. ∎

Chapter 6

Augmented Black-Box Constructions

6.1   Introduction

In the previous two chapters we have studied the limitations of black-box constructions. Such limitations are usually seen as evidence that any construction bypassing them will require new techniques. However, as mentioned in the introduction, there are several known techniques that are not black-box and black-box separations say nothing about what is feasible using these techniques. Thus, the class of black-box techniques fails to capture all "known techniques". In this chapter we propose a novel framework to partially address this shortcoming.

Specifically, we extend the model of black-box constructions to capture the most common non-black-box technique; that of using zero-knowledge proofs relative to a base primitive. We propose a model of *augmented black-box* constructions to capture this powerful class of constructions. This model consists of an oracle $\mathcal{O}$ guaranteeing the existence of some base primitive and a pair of oracles $(\mathcal{P}, \mathcal{V})$ that allow zero-knowledge proofs relative to $\mathcal{O}$. We note that such proofs are not in general possible without such oracles as, even though the existence of one-way functions implies zero-knowledge proofs for all of NP, it does not imply the existence of zero-knowledge proofs for NP$^{\mathcal{O}}$. (See [76] for further discussion.) With these oracles, a construction using zero-knowledge proofs relative to $\mathcal{O}$ can be cast as a black-box construction using oracle access to $\mathcal{O}$ and $(\mathcal{P}, \mathcal{V})$. This

model allows us to reason about the existence of such augmented black-box constructions allowing us to come much closer to truly describing what is possible using known techniques.

Our contributions. In addition to putting forth the notion of augmented black-box constructions we also show several technical results. To validate our framework, we show that the Naor-Yung/Sahai [93, 114] (shielding) construction of CCA-secure public-key encryption from CPA-secure public-key encryption falls within our framework. (Such a construction is ruled out, in a black-box sense, by the result of Gertner et al. [53].) We note that several other existing non-black-box constructions also fall within our framework, including those of [39, 12, 44]. This demonstrates that our framework meaningfully encompasses constructions that lie outside the standard black-box model.

On the negative side, we present the first impossibility result for augmented black-box constructions. Generalizing the work of Impagliazzo and Rudich [76], we rule out augmented (fully) black-box constructions of key agreement protocols with perfect completeness from one-way functions. (We leave the case of protocols without perfect completeness as an open problem.) Though it may seem "intuitively obvious" to the reader that zero-knowledge proofs cannot help in the setting of key-agreement, the challenge — as in all black-box impossibility proofs — is to prove this intuition. (In fact, under our initial modeling of a random zero knowledge proof system there *was* a construction of key agreement from one-way functions. See Section 6.3.3 for details.)

Chapter outline. To motivate the need to extend the black-box model, we begin this chapter with a brief overview of known non-black-box techniques in Section 6.2. Then,

in Section 6.3 we formally define and instantiate the notion of augmented black-box constructions, and in Section 6.4 we show that our framework encompasses the Naor-Yung/Sahai paradigm for building CCA-secure public-key encryption from CPA-secure schemes. Our main technical result is in the section that follows. Specifically, we initiate the study of augmented black-box separations by ruling out augmented black-box constructions of (perfect completeness) key agreement from one-way functions in Section 6.5.

6.2   Known Non-Black-Box Techniques

In this section, we briefly review the known non-black-box techniques that have appeared in the literature. In this summary we provide only the high-level overview of the different techniques used and refer the interested reader to the referenced works for details.

Using Zero-Knowledge Proofs: The oldest and still most commonly used non-black-box technique is that of using zero-knowledge proofs to guarantee that a primitive is used correctly. In a little more detail, this technique works as follows: To construct primitive Q (e.g. CCA-secure encryption) from a primitive P (e.g. trapdoor permutation) the construction may use P as a black-box and additionally it may give zero-knowledge proofs relative to the primitive P. More formally, the construction gives zero-knowledge proofs for some language in NP^P. Usually such proofs are used to ensure that the adversary has used the primitive P in some prescribed way. For example, in the case of constructions of CCA-secure encryption, zero-knowledge proofs are used to guarantee that any

ciphertext generated by the adversary is well-formed. What makes such constructions non-black-box is this use of zero-knowledge proofs relative to primitive P. To generate such proofs, the construction must know the circuit implementing primitive P in order to be able to reduce the statement "I used P correctly" to the corresponding statement in some NP-complete language, after which standard constructions of zero-knowledge proofs for NP [61] can be used.

This technique originated in the work of Goldreich et al. [60] who used it to enforce honest behavior in protocols for secure computation. Since then this technique has seen a significant amount of use. The most well known examples of this approach are the constructions of CCA-secure encryption from general assumptions. Specifically, all known constructions of CCA-secure encryption from trapdoor permutations [93, 37, 114, 88] are of this type. Also in the realm of public-key encryption, Pass et al. [101] use this technique to construct non-malleable encryption from CPA-secure encryption. Building on this work, Cramer et al. [33] use it to construct bounded CCA-secure encryption from CPA-secure encryption. Additional examples of constructions using this technique include Feige et al. [39] who use it to construct secure identification schemes, Bellare and Goldwasser [12] who use it to construct digital signature schemes, Fischlin [44] who uses it to construct round-optimal blind signature schemes and Boldyreva et al. [19] who use it to construct non-malleable hash-functions.

Using Secure Computation: Another non-black-box technique introduced by Beaver [11] is to execute a protocol for secure computation on the circuit for the underlying primitive P. This is done to securely evaluate the underlying primitive in such a way that both parties learn the correct output but neither party learns anything other than this. For

example, Beaver [11] uses non-black-box access to a pseudorandom generator G to do oblivious transfer extension. In this construction the two parties jointly evaluate $G(s)$ for a random seed s by running the Yao garbled circuit protocol [122] on the circuit for G.

A closely related technique is that of *computationally private randomized encodings* introduced by Applebaum et al. [2]. Roughly, a randomized encoding of a function f, is a randomized function $\hat{f}(x, r)$ such that (1) given $\hat{f}(x, r)$, $f(x)$ can be efficiently recovered and (2) given $f(x)$, it is possible to efficiently sample from $\hat{f}(x, r)$ for a random r. Such randomized encodings preserve many of the security properties of the function f so $\hat{f}$ can often be used in place of f. Applebaum et al. show that low-depth randomized encodings can be constructed for a primitive P by using the Yao garbled circuit technique [122] on the circuit for P. Thus, a construction of a low-depth primitive Q from a standard (polynomial-depth) version of the same primitive, P, can proceed as follows. First use the Yao garbled circuit to compute a low-depth randomized encoding $\hat{f}$ for the function f implementing P (given as a circuit). Then we can instantiate Q by evaluating $\hat{f}(x, r)$ for a random r. Since $\hat{f}$ has low depth this results in a low-depth implementation of P. Since this construction needs the circuit of the base primitive P (to construct the randomized encoding) it is not black-box.

Non-Black-Box Simulation: A closely related technique is that of non-black-box simulation introduced in the context of zero-knowledge arguments by Barak [5]. The non-black-box access here refers to the way that the simulator accesses the cheating verifier while generating the simulated transcript. This means that the simulator is given the code of the cheating verifier instead of just treating it as an oracle. In a breakthrough result Barak [5] showed that, in the context of zero-knowledge arguments (zero-knowledge

protocols where soundness is only required to hold against a polynomial time cheating prover), non-black-box access to the verifier can be used to get a construction achieving a number of properties that are known to be impossible for protocols using only black-box simulation.

We wish to point out that the notion of non-black-box access used here is somewhat orthogonal to the black-box constructions discussed in the remainder of this thesis. Here the black-box refers to the simulator's access to the cheating verifier, whereas everywhere else in this dissertation black-box refers to the access to the underlying primitive and the adversary breaking the security of the construction. In particular, it may be possible to give a black-box construction of a zero-knowledge protocol with non-black-box simulation. (We note that Barak's construction is not black-box due to its use of witness indistinguishable proofs relative to a collision resistant hash function.) However, we include this technique in this list due to its historical significance as the first non-black-box technique shown to bypass proven limitations of black-box techniques.

Since this original result a number of other works have used this technique to give non-black-box constructions. We list some of these works here. In a series of works Lindell, Pass and Rosen [87, 102, 99] showed constructions of bounded-concurrent two-party and multi-party secure computation. In a somewhat different direction, Barak [6] gave a protocol for coin-tossing in a constant number of rounds. This result was later used by Katz et al. [80] to give round efficient protocols for multi-party computation. Additionally, Pass and Rosen [103] used this technique to construct non-malleable commitments and zero-knowledge.

6.3 Augmented Black-Box Constructions

In this section we formally define our notion of *augmented black-box constructions*. Recall that our goal here is to model constructions that use an oracle $\mathcal{O}$ for some primitive as a black box, while also (possibly) using zero-knowledge proofs of NP statements relative to $\mathcal{O}$. To enable such proofs we introduce an additional pair of oracles $(\mathcal{P}, \mathcal{V})$ implementing a "prover" and a "verifier", respectively. We find it easiest to model $(\mathcal{P}, \mathcal{V})$ as a *witness-indistinguishable* (WI) proof system [38], and to prove our impossibility results relative to oracles achieving this notion. In Section 6.3.2, however, we show that any WI proof system can be used to construct non-interactive zero-knowledge (NIZK) proofs in the common random string model, assuming the existence of one-way functions. Thus, our model also suffices to rule out constructions using such zero-knowledge proofs.

Fix an oracle $\mathcal{O} : \{0,1\}^* \to \{0,1\}^*$. For a language L, we say $L \in \mathsf{NP}^{\mathcal{O}}$ if there exists a polynomial-time oracle machine M running in time polynomial in its first input such that $x \in L$ if and only if there exists a witness w for which $M^{\mathcal{O}}(x, w)$ accepts. (We assume a valid witness w satisfies $|w| = |x|$ without loss of generality.) For any $L \in \mathsf{NP}^{\mathcal{O}}$, we let R_L denote an NP-relation associated with L, and we let $L_n \stackrel{\text{def}}{=} L \cap \{0,1\}^n$ and $R_n \stackrel{\text{def}}{=} \{(x, w) \mid (x, w) \in R_L \text{ and } x \in L_n\}$.

We now define what it means for a pair of oracles $(\mathcal{P}, \mathcal{V})$ to be a witness-indistinguishable proof system relative to a base oracle $\mathcal{O}$. It is convenient to view the (infinite) oracles $(\mathcal{P}, \mathcal{V})$ as a sequence of oracles $\{(\mathcal{P}_n, \mathcal{V}_n)\}_{n \in \mathbb{N}}$, one for each input length. In the following all adversaries are stateful by default.

Definition 6.3.1 *Fix an oracle $\mathcal{O}$, a language $L \in \mathsf{NP}^{\mathcal{O}}$, and an* NP *relation R_L for L. An oracle*

$\mathcal{WI} = (\mathcal{P}, \mathcal{V})$ *is a* proof system *for R_L if the following hold:*

- **Perfect completeness:** *For any $n \in \mathbb{N}$, $(x, w) \in R_n$, and $r \in \{0,1\}^n$, it holds that*

  $\mathcal{V}_n(x, \mathcal{P}_n(x, w, r)) = 1.$

- **Perfect soundness:** *For any $n \in \mathbb{N}$, any $x \notin L$ and any π, it holds that $\mathcal{V}_n(x, \pi) = 0$.*

$\mathcal{WI}$ *is* witness indistinguishable (WI) *if additionally:*

- **Witness indistinguishability:** *For every polynomial-time adversary $\mathcal{A}$, it holds that*

  $|\Pr[\mathsf{ExptWI}_{\mathcal{A}}(n) = 1] - 1/2|$ *is negligible in n, where $\mathsf{ExptWI}_{\mathcal{A}}(n)$ is defined as follows:*

$(x, w_0, w_1) \leftarrow \mathcal{A}^{\mathcal{O}, \mathcal{WI}}(1^n); b \leftarrow \{0,1\};$		*if* $(x, w_0), (x, w_1) \in R_n$
$r \leftarrow \{0,1\}^n; \pi \leftarrow \mathcal{P}_n(x, w_b, r)$	:	*output 1 iff* $b' = b$
$b' = \mathcal{A}^{\mathcal{O}, \mathcal{WI}}(1^n, \pi)$		*else, output a random bit*

When the relation R_L is irrelevant for the discussion at hand, or is clear from the context, we may abuse terminology and call $\mathcal{WI}$ a WI proof system for L. We say that $\mathcal{WI}$ is a WI proof system for $\mathsf{NP}^{\mathcal{O}}$ if it is a WI proof system for the $\mathsf{NP}^{\mathcal{O}}$-complete language CIRCUIT-SAT$^{\mathcal{O}}$ (the set of satisfiable circuits with $\mathcal{O}$ gates) under the natural relation R_L.

We now define our notion of black-box constructions using a base oracle $\mathcal{O}$ and a WI oracle $\mathcal{WI}$ for $\mathsf{NP}^{\mathcal{O}}$. The definitions and terminology are adapted from the corresponding definitions in Chapter 3.

Definition 6.3.2 (Augmented fully black-box construction) *There is an* augmented fully black-box construction *of primitive Q from primitive P if there exist probabilistic polynomial-time oracle machines G and S such that:*

- *For any oracles $\mathcal{O}, \mathcal{WI}$ such that $\mathcal{O}$ implements P, and $\mathcal{WI}$ is a proof system for $\mathsf{NP}^{\mathcal{O}}$, the algorithm $G^{\mathcal{O},\mathcal{WI}}$ implements Q.*

- *For any $\mathcal{O}, \mathcal{WI}$ such that $\mathcal{WI}$ is a proof system for $\mathsf{NP}^{\mathcal{O}}$ and any (possibly inefficient) adversary $\mathcal{A}^{\mathcal{O},\mathcal{WI}}$ that breaks the Q-security of $G^{\mathcal{O},\mathcal{WI}}$, the adversary $S^{\mathcal{A},\mathcal{O},\mathcal{WI}}$ breaks the P-security of $\mathcal{O}$ or the witness indistinguishability of $\mathcal{WI}$.*

Additionally, we can define a notion of augmented semi black-box constructions as follows.

Definition 6.3.3 (Augmented semi black-box construction) *There is an* augmented semi black-box construction *of primitive Q from primitive P if there exists a probabilistic polynomial-time oracle machine G such that:*

- *For any oracles $\mathcal{O}, \mathcal{WI}$ such that $\mathcal{O}$ implements P, and $\mathcal{WI}$ is a proof system for $\mathsf{NP}^{\mathcal{O}}$, the algorithm $G^{\mathcal{O},\mathcal{WI}}$ implements Q.*

- *For any $\mathcal{O}, \mathcal{WI}$ such that $\mathcal{WI}$ is a proof system for $\mathsf{NP}^{\mathcal{O}}$ and any probabilistic polynomial-time adversary $\mathcal{A}^{\mathcal{O},\mathcal{WI}}$ that breaks the Q-security of $G^{\mathcal{O},\mathcal{WI}}$, there is a probabilistic polynomial-time S such that $S^{\mathcal{O},\mathcal{WI}}$ breaks the P-security of $\mathcal{O}$ or the witness indistinguishability of $\mathcal{WI}$.*

We remark that our notion of augmented black-box constructions is not transitive: i.e., if there is an augmented black-box construction of Q from P, and an augmented black-box construction of R from Q, this does not imply that there is an augmented black-box construction of R from P. (On the other hand, if either of the given constructions is black-box, that does imply an augmented black-box construction of R from P.)

The reason is that $\mathcal{WI}$ enables proofs for $\mathsf{NP}^\mathcal{O}$ but not $\mathsf{NP}^{\mathcal{O},\mathcal{WI}}$. While it is true that Definition 6.3.1 can be meaningfully changed to allow for proofs of $\mathsf{NP}^{\mathcal{O},\mathcal{WI}}$, doing so introduces technical issues (due to circularity) and we were unable to prove our separation results with respect to such a definition. We leave this as an interesting open question.

6.3.1   Instantiating a WI Proof System

For arbitrary $\mathcal{O}$, we now show how to instantiate a WI proof system for $\mathsf{NP}^\mathcal{O}$. We begin by describing a distribution over oracles such that an oracle sampled according to this distribution is a proof system for $\mathsf{NP}^\mathcal{O}$ and is witness indistinguishable with overwhelming probability (Lemma 6.3.7). We then show that this implies that measure 1 of the oracles under this distribution constitute a WI proof system for $\mathsf{NP}^\mathcal{O}$ (Lemma 6.3.9). Throughout this section, we take L to be $\textsc{circuit-sat}^\mathcal{O}$.

We again view the (infinite) oracle $\mathcal{WI}$ as a sequence of oracles $\{\mathcal{WI}_n = (\mathcal{P}_n, \mathcal{V}_n)\}_{n \in \mathbb{N}}$, one for each input length. Consider the distribution over $\mathcal{WI}$ where, for each n, the distribution over $\mathcal{WI}_n$ is defined as follows:

Prover oracle: $\mathcal{P}_n$ is a random function $\mathcal{P}_n : \{0,1\}^{3n} \to \{0,1\}^{7n}$ whose inputs are parsed as tuples of the form $(x, w, r) \in \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n$. Note that $\mathcal{P}_n$ is defined for all such tuples (x, w, r) of the appropriate length, and not only for those satisfying $(x, w) \in R_L$ (i.e., $\mathcal{P}_n$ does not check whether $(x, w) \in R_L$).

Verifier oracle: The verifier oracle is a function $\mathcal{V}_n : \{0,1\}^{8n} \to \{0,1\}$, whose inputs are

114

parsed as pairs of the form $(x, \pi) \in \{0, 1\}^n \times \{0, 1\}^{7n}$. The function is defined as:

$$
\mathcal{V}_n(x, \pi) = \begin{cases} 1 & \text{if } \exists(w, r) \text{ s.t. } \pi = \mathcal{P}_n(x, w, r) \ \wedge \ (x, w) \in R_L \\ \\ 0 & \text{otherwise} \end{cases}
$$

Note that $\mathcal{WI}$ sampled as above is always a proof system. It remains to show that witness indistinguishability holds with overwhelming probability. We begin by proving that, for oracles distributed as above, it is essentially impossible to "spoof" a proof. That is, for n large enough, the only way to generate a proof π such that $\mathcal{V}_n(x, \pi) = 1$ is by querying $\mathcal{P}_n$ on input $(x, w, \star)$ for some w such that $(x, w) \in R_L$. This property of the $\mathcal{WI}$ oracle will also be useful later.

More formally, for an oracle Turing machine $M^{\mathcal{O}, \mathcal{WI}}$, let Spoof_n be the event that M makes a query $\mathcal{V}_n(x, \pi)$ that returns 1, yet π was not output by a previous query $\mathcal{P}_n(x, w, \star)$ with $(x, w) \in R_L$. We prove the following bound on the probability of Spoof_n.

Lemma 6.3.4 *For any oracle $\mathcal{O}$, any oracle Turing machine $M^{\mathcal{O}, \mathcal{WI}}$ making at most q $\mathcal{V}$-queries, and any n,*

$$
\Pr[\mathsf{Spoof}_n] \leq q \cdot 2^{-4n}
$$

where the probability is taken over the choice of $\mathcal{WI}$ according to the distribution above.

Proof. We drop the subscript n for ease of presentation. Since $\mathcal{P}$ is chosen independently of $\mathcal{O}$, queries to $\mathcal{O}$ give no information about the range of $\mathcal{P}$. We assume, without loss of generality, that M never makes a query $\mathcal{V}(x, \pi)$ if π was the output of a previous query $\mathcal{P}(x, w, r) = \pi$ as such queries can be answered without querying $\mathcal{V}$.

Let $Q_0 = (x_0, \pi_0), \ldots, Q_{q-1} = (x_{q-1}, \pi_{q-1})$ be the queries that M makes to $\mathcal{V}$. For $i \in \{0, \ldots, q-1\}$, let WIN_i be the event that Q_i is the *first* query asked by M such that $\mathcal{V}(x_i, \pi_i) = 1$. We now bound the probability of WIN_i using the following thought experiment.

For $i \in \{0, \ldots, q-1\}$ define GAME_i as the following game with adversary M:

$\underline{\mathsf{GAME}_i}$: Run $M(1^n)$, asking any $\mathcal{P}$ queries made by M to the oracle. Answer M's $\mathcal{V}$ queries as follows. For $j < i$ answer M's query Q_j with 0 without querying $\mathcal{V}$. Ask Q_i to the oracle $\mathcal{V}$ and return the answer to M. Finally, answer all remaining $\mathcal{V}$ queries with 0.

Let WIN_i' be the event that M makes a query $\mathcal{V}(x, \pi) = 1$ to the oracle $\mathcal{V}$ in GAME_i. The following two claims use the above experiments to prove the lemma. Both of these claims hold for any oracle $\mathcal{O}$ and the probability is taken over the choice of $\mathcal{P}$.

Claim 6.3.5 $\Pr[WIN_i' \le 2^{-4n}]$

Proof. Consider the situation right before M makes the $\mathcal{V}$ query Q_i. Note that the image of $\mathcal{P}$ has at most 2^{3n} points. By the definition of GAME_i, no queries have been made to $\mathcal{V}$, so the 2^{3n} points in the range of $\mathcal{P}$ are distributed uniformly in the space $\{0, 1\}^{7n}$. Thus for any string $\pi \in \{0, 1\}^{7n}$, the probability that π is in the range of $\mathcal{P}$ is at most $\frac{2^{3n}}{2^{7n}} = 2^{-4n}$. This clearly bounds the probability that $\mathcal{V}(x_i, \pi_i) = 1$, proving the claim. ∎

Claim 6.3.6 *For any* $i \in \{0, \ldots, q-1\}$, $\Pr[WIN_i] = \Pr[WIN_i']$

Proof. By definition, WIN_i only occurs if Q_i is the first query asked by M such that $\mathcal{V}(x_i, \pi_i) = 1$. Thus, for WIN_i to happen, for all previous queries Q_j, we have that $\mathcal{V}(x_j, \pi_j) = 0$. However, this means that M's view in the real experiment is identical

to its view in GAME_i up to this point implying that Q_i is the same in the real game and in GAME_i. Thus, the probability that this query returns 1 is the same here and in GAME_i proving the claim. ∎

By the above, we have that

$$\Pr\left[\mathsf{Spoof}_n\right] \le \Pr\left[\bigvee_{i\in\{0,\ldots,q-1\}} \mathsf{WIN}_i\right] \le \sum_{i=0}^{q-1}\Pr[\mathsf{WIN}_i'] \le q\cdot 2^{-4n}.$$

This completes the proof of Lemma 6.3.4. ∎

We now use Lemma 6.3.4 to bound the advantage of any specific polynomial-time machine $\mathcal{A}$ in distinguishing between proofs generated by two different witnesses, for a random instance of the oracle $\mathcal{WI}$. Then, using standard techniques [76] based on the Borel-Cantelli lemma and discussed in Section 3.2.4, we show that it is possible to switch the order of quantifiers and fix a specific oracle such that any polynomial-time $\mathcal{A}$ has only a negligible distinguishing advantage. In fact, this property will hold for measure 1 of the oracles $\mathcal{WI}$.

Lemma 6.3.7 *For any oracle $\mathcal{O}$, every probabilistic polynomial-time oracle machine $\mathcal{A}$, and n large enough:*

$$\left|\Pr\left[\mathsf{ExptWI}_{\mathcal{A}}(n) = 1\right] - \frac{1}{2}\right| \le 2^{-n/2},$$

where $\mathsf{ExptWI}_{\mathcal{A}}(n)$ is as in Definition 6.3.1, and the above probability is also taken over the choice of $\mathcal{WI}$.

Proof. Consider some value of n and fix the values of $\mathcal{WI}$ other than $\mathcal{WI}_n$. Assume without loss of generality that $\mathcal{A}(1^n)$ outputs values (x, w_0, w_1) with $(x, w_0), (x, w_1) \in R_n$.

117

Then $\mathcal{A}$ is given a proof π and has to identify whether w_0 or w_1 was used to generate it.

We observe that for all $k \neq n$ the output of any query to $\mathcal{P}_k$ and $\mathcal{V}_k$ is independent of the bit b. Therefore, from this point on, we focus on queries to $\mathcal{P}_n$ and $\mathcal{V}_n$. Let q be the total number of oracle queries made by $\mathcal{A}$.

We may assume that $\mathcal{A}$ does not query $\mathcal{V}_n$ since it can simulate this oracle by itself to within statistical difference at most 2^{-n} (for n large enough). Indeed, there are three types of queries to $\mathcal{V}_n$:

- The query $\mathcal{V}_n(x, \pi)$. In this case, the output is 1.

- Queries of the form $\mathcal{V}_n(x, \pi')$, where π' was output by a previous query $\mathcal{P}_n(x, w, \star)$ with $(x, w) \in R_n$. Once again, in this case the output is 1. Note that $\mathcal{A}$ can check in polynomial time whether $(x, w) \in R_n$.

- All other queries to $\mathcal{V}_n$. In this case, Lemma 6.3.4 shows that the output of all these queries is 0 except with probability at most $q \cdot 2^{-4n}$, which is bounded by 2^{-n} for n sufficiently large.

We now show that for any $\mathcal{A}$ making at most q queries to $\mathcal{P}_n$, $\mathcal{A}$'s advantage is small. We assume, without loss of generality, that $\mathcal{A}$ never repeats any query it makes to $\mathcal{P}$. Formally, we prove the following claim:

Claim 6.3.8 *For any $n \in \mathbb{N}$, for any adversary $\mathcal{A}$ making at most q oracle queries*

$$\left| \Pr_{\mathcal{P}_n, r \leftarrow \{0,1\}^n} [\mathcal{A}^{\mathcal{P}_n}(\pi) = 1 \mid b = 0] - \Pr_{\mathcal{P}_n, r \leftarrow \{0,1\}^n} [\mathcal{A}^{\mathcal{P}_n}(\pi) = 1 \mid b = 1] \right| \leq 1/2 + q \cdot 2^{-n},$$

where $\pi = \mathcal{P}_n(x, w_b, r)$.

Proof. Consider the following experiment. First choose $b \leftarrow \{0,1\}, r \leftarrow \{0,1\}^n, \pi \leftarrow \{0,1\}^{7n}$ and give π to $\mathcal{A}$. Now, every time $\mathcal{A}$ makes a query $\mathcal{P}(x', w', r')$, if $(x', w', r') = (x, w_b, r)$ return π. Otherwise, return a random string $\pi' \leftarrow \{0,1\}^{7n}$.

The success probability of $\mathcal{A}$ in this experiment is the same as in the original game with the oracle $\mathcal{P}$. Clearly, $\mathcal{A}$ succeeds if he queries $\mathcal{P}(x, w_b, r)$. However, since π is completely independent of r, the probability that $\mathcal{A}$ succeeds by making a query with $r' = r$ is at most $q \cdot 2^{-n}$. However, if he does not make such a query $\mathcal{A}$'s view is completely independent of the bit b and so he can not distinguish between the two distributions with probability better than $1/2$. ∎

Given the above claim and the fact that $\mathcal{A}$ simulates $\mathcal{V}_n$ to within statistical distance 2^{-n}, we get that $\mathcal{A}$ can not distinguish which witness was used with probability better than $(q + 1) \cdot 2^{-n}$ which is bounded by $2^{-n/2}$ for n sufficiently large. The lemma follows.

∎

Lemma 6.3.9 *Fix an oracle $\mathcal{O}$. For measure 1 of the oracles $\mathcal{WI}$ under the distribution defined above, $\mathcal{WI}$ is a witness-indistinguishable proof system for L.*

Proof. Completeness and soundness always hold, and so we must only prove witness indistinguishability. To do so we apply a standard argument using the Borel-Cantelli lemma for reversing the order of quantifiers in Lemma 6.3.7.

Fix $\mathcal{O}$. For any $n \in \mathbb{N}$ and any probabilistic polynomial-time $\mathcal{A}$, denote by $E_{n,\mathcal{A}}$ the event in which $\mathcal{WI}$ is chosen such that

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

Lemma 6.3.7 and an averaging argument imply that for any $\mathcal{A}$ and sufficiently large n the probability of $E_{n,\mathcal{A}}$ is at most $1/n^2$. Then $\sum_n \Pr[E_{n,\mathcal{A}}]$ is finite, and so the Borel-Cantelli lemma implies that the probability over choice of $\mathcal{WI}$ that event $E_{n,\mathcal{A}}$ occurs for infinitely many values of n is zero. Thus, for large enough n and measure 1 of the oracles under the distribution in question we have

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

This holds for any specific $\mathcal{A}$, and therefore by removing a set of measure 0 for each of the (countably many) machines $\mathcal{A}$ we obtain that for measure 1 of the oracles $\mathcal{WI}$ it holds that for *all* probabilistic polynomial-time $\mathcal{A}$ the quantity $\left| \Pr\left[\mathsf{ExptWI}_{\mathcal{A}}(n) = 1\right] - \frac{1}{2} \right|$ is negligible. ■

Before concluding this section we prove a technical result regarding oracles $\mathcal{WI}$ sampled according to the distribution described earlier. We show that if f is one-way relative to $\mathcal{O}$, then for measure 1 of the oracles $\mathcal{WI}$ under the distribution defined above, f remains one-way relative to $(\mathcal{O}, \mathcal{WI})$. We note that this proof can be extended to any other security property of the oracle $\mathcal{O}$. For a discussion of security properties of an oracle see [74].

Lemma 6.3.10 *Let f be a polynomial-time oracle machine such that $f^{\mathcal{O}}$ is one-way relative to $\mathcal{O}$. Then for measure 1 of the oracles $\mathcal{WI}$ under the distribution defined above, $f^{\mathcal{O}}$ is one-way relative to $(\mathcal{O}, \mathcal{WI})$.*

Proof. It suffices to show that for any PPT $\mathcal{A}$ the probability that $\mathcal{A}^{\mathcal{O}, \mathcal{WI}}$ inverts $f^{\mathcal{O}}$ is

negligible, where the probability is also taken over choice of $\mathcal{WI}$. We can then proceed as in Lemma 6.3.9 to obtain the stated result.

Assume toward a contradiction that there exists an algorithm $\mathcal{A}$ and a polynomial $p(n) \geq n$ such that the running time of $\mathcal{A}$ is bounded by $p(n)$ and, for infinitely many n, it holds that $A^{\mathcal{O},\mathcal{WI}}$ inverts $f^{\mathcal{O}}$ with probability at least $1/p(n)$ when $\mathcal{WI}$ is chosen at random. We show how to construct a PPT algorithm $\hat{\mathcal{A}}$ such that $\hat{\mathcal{A}}^{\mathcal{O}}$ inverts $f^{\mathcal{O}}$ with inverse-polynomial probability for infinitely many values of n, a contradiction.

$\hat{\mathcal{A}}(1^n, y)$ runs $\mathcal{A}(1^n, y)$, simulating the $\mathcal{WI}$ oracle for $\mathcal{A}$ as follows. Let $k^* = \log p(n)$. Algorithm $\hat{\mathcal{A}}$ samples $\mathcal{WI}_k = (\mathcal{P}_k, \mathcal{V}_k)$ according to the prescribed distribution for all $k \leq k^*$, and these are used to (perfectly) simulate $\{\mathcal{WI}_k\}_{k \leq k^*}$ to $\mathcal{A}$. Thus, we now only need to deal with the queries of $\mathcal{A}$ to $\mathcal{WI}_k$ for $k > k^*$. When $\mathcal{A}$ queries $\mathcal{P}_k(x, w, r)$, then $\hat{\mathcal{A}}$ returns a random $\pi \in \{0,1\}^{7k}$ as the result. When $\mathcal{A}$ queries $\mathcal{V}_k(x, \pi)$ then $\hat{\mathcal{A}}$ first checks to see whether there was any prior query $\mathcal{P}_k(x, w, \star) = \pi$ with $(x, w) \in R_L$. If not, then $\hat{\mathcal{A}}$ returns 0 in response to this $\mathcal{V}_k$-query. Otherwise, $\hat{\mathcal{A}}$ returns 1.

Note that $\hat{\mathcal{A}}$'s simulation of the $\mathcal{V}$ oracle is perfect unless Spoof occurs. Thus, since $\mathcal{A}$ asks at most $p(n)$ oracle queries, by Lemma 6.3.4, $\hat{\mathcal{A}}$'s simulation of $\mathcal{A}$ degrades the latter's probability of inversion by at most $p(n) \cdot 2^{-4k^*} = \frac{p(n)}{(p(n))^4} \leq \frac{1}{2p(n)}$. This implies that $\hat{\mathcal{A}}^{\mathcal{O}}$ inverts $f^{\mathcal{O}}$ with probability at least $1/2p(n)$ for infinitely many values of n, a contradiction. ∎

6.3.2 Zero-Knowledge Proofs

We define a notion of zero knowledge, and then discuss appropriate conditions under which zero-knowledge (ZK) proofs can be constructed from WI proofs. In our

context, zero knowledge is most easily expressed in terms of non-interactive zero knowledge in the common random string model. Note that we only require zero-knowledge to hold against uniform adversaries, whereas the standard definition requires it to hold even for non-uniform adversaries.

Definition 6.3.11 *Fix an oracle $\mathcal{O}$ and a language $L \in \mathsf{NP}^{\mathcal{O}}$. An oracle* $\mathbf{ZK} = (\mathcal{P}, \mathcal{V})$ *is a* proof system in the common random string model *for L with relation R_L if there is a polynomial ℓ such that the following hold:*

- **Perfect completeness:** *For all $n \in \mathbb{N}$, all $(x, w) \in R_n$, all $\mathsf{crs} \in \{0,1\}^{\ell(n)}$, and all $r \in \{0,1\}^n$, we have $\mathcal{V}(\mathsf{crs}, x, \mathcal{P}(\mathsf{crs}, x, w, r)) = 1$.*

- **Statistical soundness:** *With all but negligible probability over choice of $\mathsf{crs} \in \{0,1\}^{\ell(n)}$, there do not exist $x \notin L_n$ and π such that $\mathcal{V}(\mathsf{crs}, x, \pi) = 1$.*

$\mathbf{ZK}$ *is a* non-interactive zero-knowledge (NIZK) proof system *if additionally:*

- **Black-box (adaptive) zero knowledge:** *There exists a* PPT simulator $\mathcal{S} \stackrel{\mathrm{def}}{=} (\mathcal{S}_1, \mathcal{S}_2)$ *such that for all probabilistic polynomial-time $\mathcal{A}$ the following is negligible:*

$$
\left| \Pr \left[
\begin{array}{l}
\mathsf{crs} \leftarrow \{0,1\}^{\ell(n)}; \\
(x, w) \leftarrow \mathcal{A}^{\mathcal{O}, \mathbf{ZK}}(\mathsf{crs}); \\
r \leftarrow \{0,1\}^n; \\
\pi \leftarrow \mathcal{P}(\mathsf{crs}, x, w, r)
\end{array}
: \mathcal{A}^{\mathcal{O}, \mathbf{ZK}}(\pi) = 1 \wedge (x, w) \in R_n \right]
\right.
$$
$$
\left. - \Pr \left[
\begin{array}{l}
(\mathsf{crs}, s) \leftarrow \mathcal{S}_1^{\mathcal{O}, \mathbf{ZK}}(1^n); \\
(x, w) \leftarrow \mathcal{A}^{\mathcal{O}, \mathbf{ZK}}(\mathsf{crs}); \\
\pi' \leftarrow \mathcal{S}_2^{\mathcal{A}, \mathcal{O}, \mathbf{ZK}}(s, x)
\end{array}
: \mathcal{A}^{\mathcal{O}, \mathbf{ZK}}(\pi') = 1 \wedge (x, w) \in R_n \right]
\right| .
$$

122

Constructing NIZK proofs from WI proofs. Fix an oracle $\mathcal{O}$, and let $\mathcal{WI} = (\mathcal{P}, \mathcal{V})$ be a WI proof system for $L =$ CIRCUIT-SAT$^{\mathcal{O}}$. We show that if a one-way function $f^{\mathcal{O}}$ exists relative to $\mathcal{O}, \mathcal{WI}$, then we can construct an NIZK proof system for NP$^{\mathcal{O}}$.

Assume $f^{\mathcal{O}}$ is one-way relative to $\mathcal{O}, \mathcal{WI}$. Using f, we can construct, in a black-box fashion, a pseudorandom generator $G^{\mathcal{O}} : \{0,1\}^n \to \{0,1\}^{2n}$ (see [73]). Define the following language $L' \in$ NP$^{\mathcal{O}}$:

$$L' \overset{\text{def}}{=} \left\{ (x, \mathsf{crs}) \text{ s.t. } \exists w \in \{0,1\}^n \text{ for which } (x, w) \in R_L \text{ or } \mathsf{crs} = G^{\mathcal{O}}(w) \right\}.$$

A zero-knowledge proof that $x \in L$ can then be constructed [40] by giving a witness-indistinguishable proof that $(x, \mathsf{crs}) \in L'$. In more detail, given a WI proof system $(\mathcal{P}, \mathcal{V})$ for L, consider the following proof system $(\mathcal{P}_{\mathbf{ZK}}, \mathcal{V}_{\mathbf{ZK}})$ for L:

Prover $\mathcal{P}_{\mathbf{ZK}}$: Given crs, x, w with $\mathsf{crs} \in \{0,1\}^{2n}$ and $(x, w) \in R_n$, set $x' = (x, \mathsf{crs})$ and note that $(x', w) \in L'$. Use a Levin reduction [86] to the NP$^{\mathcal{O}}$-complete language L to obtain $(\hat{x}, \hat{w}) \in L$. Choose $r \leftarrow \{0,1\}^{|\hat{x}|}$ and return the proof $\pi = \mathcal{P}(\hat{x}, \hat{w}, r)$.

Verifier $\mathcal{V}_{\mathbf{ZK}}$: Given crs, x, π, set $x' = (x, \mathsf{crs})$ and use a Levin reduction to the NP$^{\mathcal{O}}$-complete language L to obtain $\hat{x}$. Then output $\mathcal{V}(\hat{x}, \pi)$.

Theorem 6.3.12 *If $(\mathcal{P}, \mathcal{V})$ is a WI proof system for L and $G^{\mathcal{O}} : \{0,1\}^n \to \{0,1\}^{2n}$ is a pseudorandom generator relative to $\mathcal{O}, \mathcal{WI}$, then $(\mathcal{P}_{\mathbf{ZK}}, \mathcal{V}_{\mathbf{ZK}})$ is an NIZK proof system for L.*

Proof. Completeness is immediate, and statistical soundness of $(\mathcal{P}_{\mathbf{ZK}}, \mathcal{V}_{\mathbf{ZK}})$ follows from the perfect soundness of $(\mathcal{P}, \mathcal{V})$ and the fact that a uniform $\mathsf{crs} \in \{0,1\}^{2n}$ is in the range of G with only negligible probability.

A simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ is given as follows. $\mathcal{S}_1(1^n)$ chooses $w \leftarrow \{0,1\}^n$ computes crs $= G^{\mathcal{O}}(w)$, and then outputs (crs, w). Given x, simulator $\mathcal{S}_2$ sets $x' = (x, \text{crs})$, applies a Levin reduction to (x', w) to obtain $(\hat{x}, \hat{w}) \in L$, chooses $r \leftarrow \{0,1\}^{|\hat{x}|}$, and outputs $\pi = \mathcal{P}(\hat{x}, \hat{w}, r)$.

The fact that $\mathcal{S}$ provides a good simulation follows from pseudorandomness of G relative to $\mathcal{O}, \mathcal{WI}$, and witness indistinguishability of $\mathcal{WI}$. In a little more detail, in the real proof the distinguisher D gets (crs, π) where crs $\leftarrow \{0,1\}^{2n}$ and $\pi = \mathcal{P}(\hat{x}, \hat{w}, r)$ for $r \leftarrow \{0,1\}^{|\hat{x}|}$. In the simulated proof, D gets (crs$'$, π') where $crs' = G(w')$ for $w' \leftarrow \{0,1\}^n$ and $\pi' = \mathcal{P}(\hat{x}, \hat{w}', r)$ for $r \leftarrow \{0,1\}^n$. We show that any polynomial time distinguisher D cannot distinguish between these two distributions by a hybrid argument. Let hybrid H_0 be the real proof (crs, π). In hybrid H_1, we replace the crs in the real proof with a simulated crs$' = G(w')$. If D can distinguish between H_0 and H_1 then it breaks the pseudorandomness of G. In hybrid H_2 we replace the proof $\pi = \mathcal{P}(\hat{x}, \hat{w}, r)$ with the simulated proof $\pi' = \mathcal{P}(\hat{x}, \hat{w}', r)$. If D can distinguish between H_1 and H_2 then it breaks the witness indistinguishability of $\mathcal{WI}$. Since H_2 is distributed exactly as the simulated proof, this proves the theorem. ∎

6.3.3 On the Definition of the $\mathcal{WI}$ Oracle

We considered many possibilities for the definition of the $\mathcal{WI}$ oracle above. At first glance, it may be tempting to define the prove oracle not to depend on w as this would guarantee that a proof can not reveal any information about the witness. However, such a definition is too powerful as it allows for the following key agreement protocol when the base oracle, $\mathcal{O}$, is a one-way permutation. A picks random strings $x \leftarrow \{0,1\}^n$,

$r \leftarrow \{0,1\}^n$ and B picks $x' \leftarrow \{0,1\}^n$. They privately query $y = \mathcal{O}(x)$ and $y' = \mathcal{O}(x')$ and send the values y, y', r to each other. Now, both A and B query $\mathcal{P}$ for a proof of the following statement $s = (\exists w$ such that $y = \mathcal{O}(w) \vee y' = \mathcal{O}(w))$ and receive proofs π_A and π_B. Note, that this statement has exactly two witnesses x, x' and each of A and B knows one, but E can not learn either by the one-wayness of $\mathcal{O}$. If $\mathcal{P}$ is independent of the witness then $\pi_A = \pi_B$ and A and B have a secret which E can not learn. Note that it is also necessary that $\mathcal{P}$ take additional randomness r as an input as otherwise the proofs would not be witness indistinguishable.

6.4 An Augmented Black-Box Construction

Here we show that the Naor-Yung/Sahai construction of CCA-secure public-key encryption from CPA-secure public-key encryption can be cast as an augmented fully black-box construction. This result is not surprising; the point is to demonstrate that our framework does, indeed, capture constructions that go beyond the usual black-box ones. In particular, the construction is *shielding* in the terminology of [53], something ruled out in that same work in a black-box sense.

Let $\mathcal{O} = (G, E, D)$ be a public-key encryption scheme (with perfect correctness), and let $\mathcal{WI} = (\mathcal{P}, \mathcal{V})$ be a WI proof system for $\mathsf{NP}^{\mathcal{O}}$. Assume $\mathcal{O}$ is CPA-secure relative to $\mathcal{O}, \mathcal{WI}$. As noted in Section 6.3.2, we can use $\mathcal{WI}$ to construct an NIZK proof system $(\mathcal{P}_{\mathbf{ZK}}, \mathcal{V}_{\mathbf{ZK}})$ for $\mathsf{NP}^{\mathcal{O}}$. (Existence of CPA-secure encryption implies existence of a one-way function.) Moreover, we can use the results of Sahai [114] to transform $(\mathcal{P}_{\mathbf{ZK}}, \mathcal{V}_{\mathbf{ZK}})$ into a *simulation-sound* NIZK proof system $\mathsf{ssZK} = (\mathcal{P}_{\mathsf{ssZK}}, \mathcal{V}_{\mathsf{ssZK}})$ for $\mathsf{NP}^{\mathcal{O}}$. (We remark that for $\mathcal{WI}$ sampled according to the distribution described in Section 6.3.1, the NIZK

proof system $(\mathcal{P}_{\mathbf{ZK}}, \mathcal{V}_{\mathbf{ZK}})$ would already satisfy simulation soundness with overwhelming probability. However, here we want a construction starting from *any* WI proof system.) For notational convenience, we will treat ssZK as an NIZK proof system for the specific language

$$L \overset{\mathrm{def}}{=} \{(c_1, c_2, pk_1, pk_2) \mid \exists m, r_1, r_2 : c_1 = E^{\mathcal{O}}_{pk_1}(m; r_1) \wedge c_2 = E^{\mathcal{O}}_{pk_2}(m; r_2)\}.$$

We now describe the construction of a CCA-secure encryption scheme:

KeyGen $\mathcal{G}^{\mathcal{O}, \mathsf{ssZK}}$: Compute $(pk_1, sk_1) \leftarrow G(1^n)$ and $(pk_2, sk_2) \leftarrow G(1^n)$. Then choose $\mathsf{crs} \leftarrow \{0, 1\}^{\ell(n)}$ and set $PK = (pk_1, pk_2, \mathsf{crs})$ and $SK = (sk_1, sk_2)$.

Encryption $\mathcal{E}^{\mathcal{O}, \mathsf{ssZK}}$: To encrypt plaintext m, choose $r_1, r_2, r \leftarrow \{0, 1\}^n$ and then compute the ciphertexts $c_1 = E_{pk_1}(m; r_1)$ and $c_2 = E_{pk_2}(m; r_2)$. Set $x = (c_1, c_2, pk_1, pk_2)$ and $w = (m, r_1, r_2)$ and generate an NIZK proof $\pi = \mathcal{P}_{\mathsf{ssZK}}(\mathsf{crs}, x, w, r)$. Output (c_1, c_2, π).

Decryption $\mathcal{D}^{\mathcal{O}, \mathsf{ssZK}}$: To decrypt (c_1, c_2, π), set $x = (c_1, c_2, pk_1, pk_2)$ and check that $\mathcal{V}_{\mathsf{ssZK}}(\mathsf{crs}, x, \pi) = 1$. If not, output $\bot$. Otherwise, output $m = D_{sk_1}(c_1)$.

The following theorem follows from [114, Theorem 4.1]. We note that even though [114] proves this theorem for non-uniform zero-knowledge proofs, uniform zero-knowledge suffices since we consider a uniform notion of CCA-security.

Theorem 6.4.1 *For any $\mathcal{O}$ implementing an encryption scheme (with perfect correctness) that is CPA-secure relative to $\mathcal{O}, \mathcal{WI}$, the above construction is CCA-secure relative to $\mathcal{O}, \mathcal{WI}$. Thus, the above is an augmented fully black-box construction of a CCA-secure encryption scheme from CPA-secure encryption.*

6.5 An Impossibility Result for Key Agreement

In this section, we rule out augmented black-box constructions of key agreement with perfect completeness from one-way functions. (We conjecture that the result extends to the case of imperfect completeness, but we were unable to prove this.) For the remainder of this section, we only consider 1-bit key-agreement protocols with perfect completeness.

Say (A, B) is a pair of polynomial-time oracle algorithms that is an augmented black-box construction of key agreement from one-way functions. Then:

- For any $\mathcal{O}, \mathcal{WI}$ such that $\mathcal{WI}$ is a proof system for $\text{NP}^{\mathcal{O}}$ and all n, following an execution between $A^{\mathcal{O},\mathcal{WI}}(1^n)$ and $B^{\mathcal{O},\mathcal{WI}}(1^n)$ both parties agree on a common bit $k \in \{0, 1\}$.

- Given (A, B) and E, define the advantage of E by the following experiment:

  1. $A^{\mathcal{O},\mathcal{WI}}(1^n)$ and $B^{\mathcal{O},\mathcal{WI}}(1^n)$ interact, resulting in a shared key k and a transcript T.

  2. E is given T, and outputs a bit k'.

  The advantage of E is $|\Pr[k' = k] - 1/2|$.

  For any $\mathcal{O}$ and $\mathcal{WI}$ such that $\mathcal{O}$ is one-way relative to $(\mathcal{O}, \mathcal{WI})$ and $\mathcal{WI}$ is a WI proof system for $\text{NP}^{\mathcal{O}}$, every unbounded algorithm E making at most polynomially many queries to $\mathcal{O}$ and $\mathcal{WI}$ has negligible advantage.

To prove that no augmented (fully) black-box construction of key agreement from one-way functions exists, we instantiate the oracle $\mathcal{O}$ with a random oracle and choose

127

$\mathcal{WI}$ as described in Section 6.3.1. That is, $\mathcal{O} = \{\mathcal{O}_n\}_{n \in \mathbb{N}}$ where for each $n \in \mathbb{N}$, $\mathcal{O}_n$ is chosen uniformly at random from the space of all functions from $\{0,1\}^n \to \{0,1\}^n$. A random oracle is one-way [76], and Lemma 6.3.10 shows that it remains one-way in the presence of $\mathcal{WI}$ chosen from the specified distribution. Moreover, by Lemma 6.3.9 we have that $\mathcal{WI}$ is a WI proof system for $\mathsf{NP}^{\mathcal{O}}$. We note that even though these lemmas are stated with respect to polynomial time adversaries, since our proofs relativize, they also hold for computationally unbounded adversaries making at most polynomially many oracle queries.

Now consider a construction $(A^{\mathcal{O},\mathcal{WI}}, B^{\mathcal{O},\mathcal{WI}})$ of key-agreement relative to these oracles. if (A, B) is an augmented black-box construction of key-agreement from one-way functions, then for any unbounded algorithm E making at most polynomially many oracle queries that has non-negligible advantage, there should exist a polynomial time machine $S^{E,\mathcal{O},\mathcal{WI}}$ that inverts $\mathcal{O}$ or breaks the witness indistinguishability of $\mathcal{WI}$. However, since S makes at most polynomially many queries to $\mathcal{O},\mathcal{WI}$, such an S does not exist. Therefore, every unbounded algorithm E making at most polynomially many queries to $\mathcal{O}$ and $\mathcal{WI}$ should have negligible advantage. However, we show an explicit E for which this is not the case, thus proving that no augmented (fully) black-box construction of key agreement from one-way functions exists. As described, E is not polynomial time. However, E can be made efficient if $\mathsf{P} = \mathsf{NP}$; thus any augmented semi-black-box construction of key agreement from one-way functions would imply $\mathsf{P} \neq \mathsf{NP}$.

6.5.1 Breaking Key Agreement Relative to a Random Oracle

In this section we provide a warmup for our main proof by ruling out (standard) black-box constructions of key agreement from one-way functions. This proof may also be of independent interest for pedagogical purposes as a simplified version of the proofs in [76, 10]. Note, however, that we prove a weaker result: we only rule out constructions of key-agreement protocols with perfect completeness based on one-way functions whereas [76, 10] rule out constructions with arbitrary completeness even based on one-way permutations.

Let (A, B) be a construction of key agreement from one-way functions. Let q_A (resp., q_B) be a polynomial upper bound on the number of queries made by A (resp., B). Consider an attacker E defined as follows. E, given a transcript trans of an execution of (A, B) in the presence of a random oracle $\mathcal{O}$, maintains a set $Q(E)$ of query/answer pairs for $\mathcal{O}$, and a multiset of candidate keys K, both initialized to $\emptyset$. Then E runs $2q_B + 1$ iterations of the following attack:

- *Simulation phase:* E finds a view of A consistent with the given transcript and with $Q(E)$. This view contains the randomness r_A used by A, as well as a set of oracle queries/answers $\hat{Q}(A)$ made by A. The set $\hat{Q}(A)$ is chosen to be consistent with any queries/answers in $Q(E)$, but it need not be consistent with the true oracle $\mathcal{O}$.

  Let k denote the key computed by A in the view. Then E adds k to K.

- *Update phase:* E makes all queries in $\hat{Q}(A) \setminus Q(E)$ to the true oracle $\mathcal{O}$, and adds the resulting query/answer pairs to $Q(E)$.

Following the above, E has a multiset K of $2q_B + 1$ possible keys. E outputs the majority value in K.

In each iteration, E makes at most q_A queries to $\mathcal{O}$. Thus, E makes $O(q_A \cdot q_B)$ queries overall. We claim that E outputs the key computed by A and B with probability 1. Toward this, we first prove the following:

Claim 6.5.1 *Let k denote the actual key computed by A and B in an execution of the protocol. Then in each iteration of the attack, either E adds k to K, or E adds to $Q(E)$ one of the queries made by B in the real execution.*

Proof. Let $Q(B)$ denote the queries made by B in the real execution of the protocol. In a given iteration, there are two possibilities. If $\hat{Q}(A) \cap Q(B) \not\subseteq Q(E)$, then we are done since E makes all queries in $\hat{Q}(A) \setminus Q(E)$ to the true oracle $\mathcal{O}$. If, on the other hand, $\hat{Q}(A) \cap Q(B) \subseteq Q(E)$ then there is an oracle $\tilde{\mathcal{O}}$ that is consistent with the sampled view of A and the view of the real B. That is, there is an execution of the protocol with an oracle $\tilde{\mathcal{O}}$ that yields the observed transcript trans, a view for B identical to the view of the real B, and a view for A identical to the view generated by E in the current iteration. Perfect completeness implies that the key k computed by A in this case must match the (actual) key computed by B. ∎

Since B makes at most q_B queries, it follows that there are at most q_B iterations in which E adds an incorrect key to K, and so at least $q_B + 1$ iterations in which E adds the correct key to K. Since E outputs the key that occurs most often, E always outputs the correct key.

6.5.2 Breaking Key Agreement Relative to $\mathcal{O}, \mathcal{WI}$

Here we prove our main result:

Theorem 6.5.2 *There is no augmented fully black-box construction of key agreement with perfect completeness from one-way functions.*

The overall structure of the attack is the same as in the previous section, but there are some key differences. Our attack again proceeds by having E repeatedly find a view of A consistent with a transcript trans and the oracle queries $Q(E)$ that E has made thus far. Let $Q(A)$ and $Q(B)$ denote the queries of A and B, respectively, in the actual execution of the protocol, and let $\hat{Q}(A)$ denote the queries of A in the view found by E in some iteration. In the previous section we argued that as long as $\hat{Q}(A) \cap Q(B) \subseteq Q(E)$, the key found by E in the given iteration matches the key computed by the real B. This was because, under that condition, there must exist an oracle $\tilde{\mathcal{O}}$ that is consistent with an execution of the protocol in which party A makes queries $\hat{Q}(A)$, party B makes queries $Q(B)$, and the resulting transcript is trans. Here, however, this need not be the case. For example, consider a real execution of the protocol in which B makes a query $\mathcal{V}(x, \pi)$ that returns 1, yet B does not make any corresponding query $\mathcal{P}(x, w, \star) = \pi$ with $(x, w) \in R_L$. If E samples a view of A in which $x \notin L$, then there are no oracles $\tilde{O}, \widetilde{\mathcal{WI}}$ consistent with the sampled view of A and the real view of B, but neither does E necessarily learn any new queries in $Q(B)$.

We deal with the above by modifying the attack and changing the proof. First, we modify the attack by having E sample *extended* views of A, which include a view of A along with additional oracle queries used for "book-keeping". Second, rather than

131

showing that, in every iteration, E either learns the correct key or a query in $Q(B)$, we show that, in every iteration, E either learns the correct key or a query in $Q(AB) \stackrel{\text{def}}{=} Q(A) \cup Q(B)$.

An additional subtlety arises due to the possibility that Spoof_i occurs (cf. Lemma 6.3.4) for some i. In our attack we handle this by guaranteeing that $\mathsf{Spoof} = \cup_i \mathsf{Spoof}_i$ occurs with sufficiently small probability, and showing that the attack is successful whenever Spoof does not occur. (Our proof can be significantly simplified if we make the assumption that $A(1^n)$ and $B(1^n)$ only query their oracles on inputs of length n, however we wish to avoid this assumption.)

Preliminaries: We view $Q(A), Q(B)$, and $Q(E)$ interchangeably as sets of queries and sets of query/answer pairs. We write, e.g., $[\mathcal{P}(x, w, r) = \pi] \in Q(A)$ to denote that A made the query $\mathcal{P}(x, w, r)$ and received the answer π. As usual, we let L denote the set of satisfiable circuits with $\mathcal{O}$-gates.

We assume any key-agreement construction (A, B) has the following normal form: Before a party queries $\mathcal{P}(x, w, r)$, that party also asks all $\mathcal{O}$-queries necessary to check whether $(x, w) \in R_L$; after receiving the result $\pi = \mathcal{P}(x, w, r)$, that party also asks $\mathcal{V}(x, \pi)$. Any key-agreement protocol can be modified to satisfy this condition with only a polynomial blow-up in the number of queries. We let $q = q(n) \geq n$ denote a polynomial upper bound on the combined running time of A and B (and so in particular a bound on the number of queries they make).

Without loss of generality, assume that for any (circuit) $x \in \{0,1\}^n$ and $w \in \{0,1\}^n$, computation of x on input w queries $\mathcal{O}$ at most n times, each time on input of length at most n. In other words, deciding whether $(x, w) \in \textsc{circuit-sat}^{\mathcal{O}}$ depends only on the

values of $\mathcal{O}$ on inputs of length at most n.

Extended views of A**:** In our attack, E will repeatedly sample *extended* views of A which include A's view along with some additional oracle queries/answers. We denote an extended view by $(r_A, \mathcal{O}', \mathcal{WI}')$, where r_A are the random coins of A and $\mathcal{O}', \mathcal{WI}'$ is a set of query/answer pairs that includes all those made by A (using coins r_A and the given transcript). E samples only *consistent* extended views, which we define now.

Definition 6.5.3 *Let* $Q = (\mathcal{O}', \mathcal{WI}' = (\mathcal{P}', \mathcal{V}'))$ *be a set of queries/answers. We say it is* consistent *if*

1. *For every query* $[\mathcal{P}'(x, w, r) = \pi] \in \mathcal{WI}'$*, oracle* $\mathcal{O}'$ *contains queries/answers sufficient to determine whether* $(x, w) \in R_L$*. Moreover, if* $(x, w) \in R_L$ *then* $[\mathcal{V}'(x, \pi) = 1] \in \mathcal{WI}'$*, while if* $(x, w) \notin R_L$ *then* $[\mathcal{V}'(x, \pi) = 0] \in \mathcal{WI}'$*.*

2. *For every query* $[\mathcal{V}'(x, \pi) = 1] \in \mathcal{WI}'$*, there exist* w, r *such that* $\mathcal{O}'$ *contains queries/answers for which* $(x, w) \in R_L$ *and* $[\mathcal{P}'(x, w, r) = \pi] \in \mathcal{WI}'$*.*

Let trans *be a transcript of an execution between* $A(1^n)$ *and* $B(1^n)$*, and let* $Q(E)$ *be a set of queries/answers. We say the extended view* $(r_A, \mathcal{O}', \mathcal{WI}')$ *is* consistent with trans *and* $Q(E)$ *if* $\mathcal{O}', \mathcal{WI}'$ *is consistent, and also:*

1. *Every query in* $Q(E)$ *is in* $\mathcal{O}', \mathcal{WI}'$*, and is answered the same way.*

2. $A^{\mathcal{O}', \mathcal{WI}'}(1^n; r_A)$*, when fed with incoming messages as in* trans*, would generate outgoing messages consistent with* trans*. Furthermore, all oracle queries/answers made/received by* A *in such an execution are in* $\mathcal{O}', \mathcal{WI}'$*.*

The attack. Let $t = 4 \log q$. First, in a pre-processing step, E queries $\mathcal{O}(x)$ for all x with $|x| \leq t$; queries $\mathcal{P}(x, w, r)$ for all x, w, r with $|x| = |w| = |r| \leq t$; and queries $V(x, \pi)$ for all x, π with $|x| = |\pi|/7 \leq t$. Denote these queries/answers by $Q^*(E)$. The rest of the attack is similar to that of the previous section. E, given a transcript trans of an execution of (A, B), initializes $Q(E) = Q^*(E)$ and $K = \emptyset$, and then runs $2q + 1$ iterations of the following:

- *Simulation phase:* E finds an extended view $(r_A, \mathcal{O}', \mathcal{WI}')$ consistent with trans and $Q(E)$, with $\mathcal{O}', \mathcal{WI}'$ of size at most $|Q(E)| + q$. (If no such extended view exists, E aborts.) Let k be the key computed by A in this view. E adds k to K.

- *Update phase:* E makes all queries in $(\mathcal{O}' \cup \mathcal{WI}') \setminus Q(E)$ to the true oracles $\mathcal{O}, \mathcal{WI}$. For any queries $[\mathcal{P}'(x, w, r) = \pi]$ just made, E also makes any $\mathcal{O}$ queries needed to determine whether $(x, w) \in R_L$, as well as the query $V(x, \pi)$. All the resulting query/answer pairs are added to $Q(E)$.

Following the above, E has a multiset K of $2q + 1$ possible keys. E outputs the majority value in K.

Analysis. In pre-processing, E makes polynomially many queries. In each iteration of the attack, E makes at most $q + q(q + 1) \leq 3q^2$ queries: there are at most q queries in $(\mathcal{O}' \cup \mathcal{WI}') \setminus Q(E)$, and for each such query of the form $[\mathcal{P}'(x, w, r) = \pi]$ we have $|x| \leq q$ and so at most q queries are needed to check whether $(x, w) \in R_L$ and one additional query for $V(x, \pi)$. Thus, E makes at most $3q^2 \cdot (2q + 1)$ queries after the pre-processing, which is bounded by $7q^3$ for $q > 3$.

For any i, define Spoof_i (cf. Lemma 6.3.4) to be the event that there is a query $[\mathcal{V}_i(x, \pi) = 1] \in Q(A) \cup Q(B)$, yet there is no query

$$[\mathcal{P}_i(x, w, \star) = \pi] \in Q(A) \cup Q(B) \cup Q^*(E)$$

with $(x, w) \in R_L$. Let $\mathsf{Spoof} = \bigvee_i \mathsf{Spoof}_i$. We claim that Spoof occurs with probability at most $1/8$. Indeed, by construction Spoof_i cannot occur for $i \leq t$, and for $i > t$, by Lemma 6.3.4 $\Pr[\mathsf{Spoof}_i] \leq q \cdot 2^{-4i}$. Thus, by a union bound, $\Pr[\bigvee_{i>t} \mathsf{Spoof}_i] \leq q^{-15} \cdot \sum_{i=1}^{\infty} (2^{-4})^i \leq 1/8$ for $q \geq 2$.

Define Spoof' to be the event that, at some point during the attack, E queries $\mathcal{V}(x, \pi) = 1$ to the real oracle, but there was no previous query $[\mathcal{P}_i(x, w, \star) = \pi]$ made by A, B, or E with $(x, w) \in R_L$. By construction, this can only possibly occur if $|x| > 4 \log q$. Since E makes at most $7q^3$ queries after the pre-processing stage, however, Spoof' occurs with probability at most $1/8$.

In the rest of the analysis, we show that as long as neither Spoof nor Spoof' occur, E outputs the key computed by A and B. This suffices, since then E finds the shared key with probability at least $3/4$ overall. Then, as in the previous section, the following lemma will prove Theorem 6.5.2:

Lemma 6.5.4 *Let k denote the actual key computed by A and B in an execution of the protocol, and assume neither Spoof nor Spoof' occur. Then E does not abort, and in each iteration of the attack either E adds k to K, or E adds to $Q(E)$ one of the queries made by A or B in the real execution.*

Proof. Let $Q(AB) \stackrel{\text{def}}{=} Q(A) \cup Q(B)$ denote the queries/answers made/received by A or B

in the real execution. We first show that E never aborts. Say $Q(E)$ is consistent at the beginning of some iteration; this is true by construction in the first iteration. Since Spoof did not occur, a consistent, extended view is given by letting $(\mathcal{O}', \mathcal{WI}') = Q(E) \cup Q(AB)$, which is of size at most $|Q(E)| + q$. Moreover, regardless of what consistent, extended view is actually sampled by E, the new set $Q(E)$ defined at the end of the iteration is consistent unless Spoof' occurs.

In the remainder of the proof we assume that neither Spoof nor Spoof' occur. We now prove the rest of the lemma. Let $(r_A, \mathcal{O}', \mathcal{WI}')$ be the consistent, extended view chosen by E in some iteration. We define three events, and show:

- If one of the events occurs, then, in the update phase of that iteration, E adds to $Q(E)$ some query in $Q(AB)$.

- If none of the events occur then there are oracles $\tilde{\mathcal{O}}, \widetilde{\mathcal{WI}}$ that match (i.e., are not inconsistent with) the extended view of A and the real view of B. (Thus, by perfect completeness, E adds the correct key to K in that iteration.)

Before defining the events, we introduce some terminology. Given some set of queries Q, we say Q *fixes* $x \in L$ if either (1) there exists a w and $\mathcal{O}$-queries in Q such that $(x, w) \in R_L$, or (2) there is a query $[\mathcal{V}(x, \star) = 1] \in Q$. We say Q *fixes* $x \notin L$ if for all w there are $\mathcal{O}$-queries in Q such that, regardless of how any of the $\mathcal{O}$-queries not in Q are answered, it holds that $(x, w) \notin R_L$. We define Q *fixes* $(x, w) \in R_L$ and Q *fixes* $(x, w) \notin R_L$ in the obvious way.

We now define the events of interest:

E_1: $\mathcal{O}', \mathcal{WI}'$ disagrees with $Q(AB)$ on the answer to some $\mathcal{O}$-, $\mathcal{P}$-, or $\mathcal{V}$-query.

136

E_2: There exists an x such that $Q(AB)$ fixes $x \in L$ but $\mathcal{O}', \mathcal{WI}'$ fixes $x \notin L$, or vice versa.

E_3: A $\mathcal{V}'$-query returning 0 in $\mathcal{WI}'$ is "inconsistent" with the $\mathcal{O}, \mathcal{P}$ queries in $Q(AB)$, or vice versa. Formally, one of the following occurs:

- There is a query $[\mathcal{V}'(x, \pi) = 0] \in \mathcal{WI}'$, but $[\mathcal{P}(x, w, \star) = \pi] \in Q(AB)$ and $Q(AB)$ fixes $(x, w) \in R_L$.

- There is a query $[\mathcal{P}'(x, w, \star) = \pi] \in \mathcal{WI}'$ and $\mathcal{O}'$ fixes $(x, w) \in R_L$, but $[\mathcal{V}(x, \pi) = 0] \in Q(AB)$.

Claim 6.5.5 *If any of E_1, E_2, or E_3 occur in the simulation phase of some iteration, then E learns a new query in $Q(AB)$ in the update phase of that iteration.*

Proof. If E_1 occurs, the claim is immediate. ($Q(E)$ contains the answers of the true oracles, and so can never disagree with $Q(AB)$. So any disagreement between $\mathcal{O}', \mathcal{WI}'$ and $Q(AB)$ must be due to some query in $\mathcal{O}', \mathcal{WI}'$ outside of $Q(E)$.) If E_2 occurs there are several sub-cases to consider:

1. Say $Q(AB)$ fixes $x \in L$, but $\mathcal{O}', \mathcal{WI}'$ fixes $x \notin L$. The second event implies that for all w oracle $\mathcal{O}'$ fixes $(x, w) \notin R_L$. There are two ways the first event can occur:

   - There exists a w such that $Q(AB)$ fixes $(x, w) \in R_L$. In this case there must be an $\mathcal{O}$-query in $Q(AB)$ that is answered inconsistently with some query in $\mathcal{O}'$, and event E_1 has occurred.

   - There is a query $[\mathcal{V}(x, \pi) = 1] \in Q(AB)$ (for some π). Since Spoof has not occurred, this means that for some w, r there is a query $[\mathcal{P}(x, w, r) = \pi]$ in $Q(AB)$

137

or $Q^*(E)$. Say $[\mathcal{P}(x, w, r) = \pi] \in Q(AB)$. Then by our normal-form assumption, $Q(AB)$ fixes $(x, w) \in R_L$; this, in turn, implies an $\mathcal{O}$-query in $Q(AB)$ inconsistent with $\mathcal{O}'$ (which, recall, fixed $x \notin L$), and so E_1 has occurred.

On the other hand, say $[\mathcal{P}(x, w, r) = \pi] \in Q^*(E)$. Then, by construction of $Q^*(E)$, the query $[\mathcal{V}(x, \pi) = 1]$ is also in $Q^*(E)$, and $Q^*(E)$ fixes $(x, w) \in R_L$. But since any queries in $\mathcal{O}'$ must agree with the corresponding $\mathcal{O}$-queries in $Q^*(E)$, this cannot happen.

2. Say $\mathcal{O}', \mathcal{WI}'$ fixes $x \in L$, but $Q(AB)$ fixes $x \notin L$. The second event implies that for all w we have that $Q(AB)$ fixes $(x, w) \notin R_L$. There are two ways the first event can occur:

   - There exists a w for which $\mathcal{O}'$ fixes $(x, w) \in R_L$. In this case there is an $\mathcal{O}$-query in $Q(AB)$ that is answered inconsistently with some query in $\mathcal{O}'$, and event E_1 has occurred.

   - There is a query $[\mathcal{V}(x, \pi) = 1] \in \mathcal{WI}'$ for some π. By definition of consistency, there exists w such that $\mathcal{O}'$ fixes $(x, w) \in R_L$. Then there must be an $\mathcal{O}$-query in $Q(AB)$ that is answered inconsistently with $\mathcal{O}'$, and so E_1 has occurred.

Finally, we turn to E_3. Here there are two sub-cases:

1. Say $[\mathcal{V}'(x, \pi) = 0] \in \mathcal{WI}'$, but $[\mathcal{P}(x, w, \star) = \pi] \in Q(AB)$ and furthermore $Q(AB)$ fixes $(x, w) \in R_L$. Because of our normal-form assumption, $[\mathcal{V}(x, \pi) = 1] \in Q(AB)$. Thus there is a $\mathcal{V}$-query in $Q(AB)$ that is answered inconsistently with $\mathcal{WI}'$ and so E_1 has occurred.

2. Say $[\mathcal{P}'(x, w, \star) = \pi] \in \mathcal{WI}'$ and $\mathcal{O}'$ fixes $(x, w) \in R_L$, but we have $[\mathcal{V}(x, \pi) = 0] \in Q(AB)$. By definition of consistency, $[\mathcal{V}(x, \pi) = 1] \in \mathcal{WI}'$. Thus there is a $\mathcal{V}$-query in $Q(AB)$ that is answered inconsistently with $\mathcal{WI}'$, and so E_1 has occurred.

This concludes the proof of Claim 6.5.5. ∎

To complete the proof of the lemma, we show that if none of E_1, E_2, or E_3 occur, there exist oracles $\tilde{\mathcal{O}}, \widetilde{\mathcal{WI}}$ (in the support of the distribution from Section 6.3.1) that match (i.e., do not disagree with) $\mathcal{O}', \mathcal{WI}'$, and $Q(AB)$. This means there is an execution of the protocol with oracles $\tilde{\mathcal{O}}, \widetilde{\mathcal{WI}}$ that yields a view for B identical to the view of the real B, and a view for A identical to the view of A in the extended view sampled by E. Perfect completeness implies that the key k computed by A in that case must match the (actual) key computed by B, as we needed to show.

We construct $\tilde{\mathcal{O}}, \widetilde{\mathcal{WI}}$ as follows. First, answer all queries in $\mathcal{O}', \mathcal{WI}'$, and $Q(AB)$ as answered by those oracles; if E_1 does not occur, this is well-defined as there is no conflict. Answer all other queries in $\tilde{\mathcal{O}}$ arbitrarily. Note that if $\mathcal{O}', \mathcal{WI}', Q(AB)$ fixes $x \in L$ then so does $\tilde{\mathcal{O}}$, and similarly if $\mathcal{O}', \mathcal{WI}', Q(AB)$ fixes $x \notin L$. Note also that with $\tilde{\mathcal{O}}$ fixed, so are $\tilde{L}$ and $\tilde{R}_L$.

For $\tilde{\mathcal{P}}$, proceed as follows. Recall that all $\tilde{\mathcal{P}}_i$ queries for $i \le t = 4\log q$ were made by E during pre-processing and so are already fixed. Any other unassigned query $\tilde{\mathcal{P}}(x, w, r)$ with $|x| > t$ is defined as follows:

- If $(x, w) \notin \tilde{R}_L$, the query is answered arbitrarily.

- If $(x, w) \in \tilde{R}_L$, let $\pi^* \in \{0, 1\}^{7|x|}$ be such that $\mathcal{V}(x, \pi^*)$ is not in $\mathcal{WI}'$ or $Q(AB)$. (There must exist such a π^*, by the bound on the number of queries in these sets.)

Set $\tilde{\mathcal{P}}(x, w, r) = \pi^*$.

With the $\tilde{\mathcal{O}}$ and $\tilde{\mathcal{P}}$ queries fixed, oracle $\tilde{\mathcal{V}}$ is set as in Section 6.3.1.

We show that $\tilde{\mathcal{O}}, \widetilde{\mathcal{WI}}$ match (i.e., do not disagree with) $\mathcal{O}', \mathcal{WI}'$, and $Q(AB)$. By construction, the only possible conflict can be between $\tilde{\mathcal{V}}$ and some $\mathcal{V}$-query in $\mathcal{WI}'$ or $Q(AB)$. No such conflict is possible:

1. Say $[\mathcal{V}(x, \pi) = 1] \in \mathcal{WI}'$ for some x, π. Then by definition of consistency, there exist w, r such that $\mathcal{O}'$ fixes $(x, w) \in R_L$, and $[\mathcal{P}(x, w, r) = \pi] \in \mathcal{WI}'$. But then $(x, w) \in \tilde{R}_L$ and $\tilde{\mathcal{P}}(x, w, r) = \pi$, and so $\tilde{\mathcal{V}}(x, \pi) = 1$.

2. Say $[\mathcal{V}(x, \pi) = 1] \in Q(AB)$ for some x, π. Since Spoof does not occur, there exist w, r such that $\mathcal{O}' \cup Q(AB)$ fixes $(x, w) \in R_L$, and $[\mathcal{P}(x, w, r) = \pi] \in \mathcal{WI}' \cup Q(AB)$. But then $(x, w) \in \tilde{R}_L$ and $\tilde{\mathcal{P}}(x, w, r) = \pi$, and so $\tilde{\mathcal{V}}(x, \pi) = 1$.

3. Say $[\mathcal{V}(x, \pi) = 0] \in \mathcal{WI}' \cup Q(AB)$ for some x, π. If $x \notin \tilde{L}$ then $\tilde{\mathcal{V}}(x, \pi) = 0$ also. If $x \in \tilde{L}$, there is an inconsistency only if there is some w such that $\tilde{\mathcal{P}}(x, w, \star) = \pi$ and $(x, w) \in \tilde{R}_L$. Note that $\tilde{\mathcal{P}}(x, w, \star) = \pi$ can only occur if $[\mathcal{P}(x, w, \star) = \pi] \in \mathcal{WI}' \cup Q(AB)$, but in that case (since $[\mathcal{V}(x, \pi) = 0] \in \mathcal{WI}' \cup Q(AB)$ and E_3 did not occur) either $\mathcal{O}'$ or $Q(AB)$ fix $(x, w) \notin R_L$, and hence $(x, w) \notin \tilde{R}_L$ either.

This completes the proof of Lemma 6.5.4. ∎

Chapter 7

Conclusions

In this dissertation we have studied the limitations of cryptographic construc-
tions. We began, in Chapters 4 and 5, with the restricted class of black-box construc-
tions demonstrating new black-box separations between several widely used crypto-
graphic primitives. Then, in Chapter 6, we proposed the model of *augmented black-box*
constructions to capture a richer class of techniques. Using this new model we were
able to demonstrate limitations on the power of a commonly used class of non-black-
box constructions, those using zero-knowledge proofs. We view this as a significant step
towards understanding the relationships between cryptographic primitives and, more
importantly, for truly capturing what is possible using "known techniques".

We believe that the study of relationships among cryptographic primitives is an im-
portant one. Separation results can make clear fundamental differences between primi-
tives. They can also save a lot of wasted effort by guiding researchers away from hopeless
approaches. Additionally, such results may aid in finding new constructions by pinpoint-
ing exact properties that a construction can have in order to bypass them. For these rea-
sons, we believe that our augmented black-box model will find further applications in the
study of cryptographic primitives and the search for new cryptographic constructions.

Many open questions remain following our work, both in the traditional setting of
black-box separations and in the setting of augmented black-box separations. Some such
questions include:

Black-Box Separations. Perhaps the most interesting open question in the study of black-box constructions is the black-box complexity of CCA-secure encryption. Specifically, it is not known if there exists a black-box construction of CCA-secure encryption from CPA-secure encryption or even from trapdoor permutations. Interestingly, for the case of trapdoor permutations there exist non-black-box constructions [93, 37, 114, 88]. Whereas, for the case of constructions from CPA-secure encryption, there exists a partial black-box separation [53]. However, these results fail to resolve the question regarding the existence of a black-box construction. Since CCA-secure encryption has been accepted as the standard security notion for public-key encryption, it is an interesting and important open question to understand the black-box complexity of this primitive.

Another interesting problem in the area of black-box separations is to investigate the power of fully-homomorphic encryption. Fully-homomorphic encryption (FHE) is a very powerful new primitive that has only recently been realized [51] and has already proven very useful in many constructions and applications. We believe that an interesting question is to explore the limits of what can, and what cannot, be accomplished using this powerful new primitive. As a starting point, we propose looking at limitations on the power of black-box constructions using fully-homomorphic encryption.

Non-Black-Box Separations. Even more open problems remain in the study of non-black-box constructions and separations. The augmented black-box suggested in Chapter 6 gives a way to study such constructions and we suggest several directions for improving our results and model. The first such open question is to close the gap between our result and the corresponding black-box separations [76, 10]. Specifically, it would be interesting to prove that our augmented black-box separation holds also for the case of

constructions of key agreement with imperfect completeness and for constructions start-ing from one-way permutations. Going beyond key agreement, it would be interesting to study what other primitives can be separated under augmented black-box separations. Such separations will help us better understand the power of zero-knowledge proofs in cryptographic constructions. For some preliminary results in this direction see [26].

The augmented black-box model comes short of capturing all known techniques. Thus, an important line of work is to extend this model to capture additional construc-tions. Some potential problems in this direction are as follows. Currently, augmented black-box constructions do not allow one to give proofs of proofs. More formally, the Prover oracle can not prove membership in a language defined relative to itself. It would be interesting to find an alternative model in which such proofs are allowed but our sep-aration result still holds. Additionally, the augmented black-box model fails to capture many known non-black-box techniques (e.g. [11, 2, 5]). It would be very interesting to devise a model capturing some of these constructions. Specifically, we believe that capturing the non-black-box simulation technique of [5] would be of particular interest.

Bibliography

[1] William Aiello and Johan Håstad. Statistical zero-knowledge languages can be recognized in two rounds. *Journal of Computer and System Sciences*, 42(3):327–345, 1991.

[2] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Computationally private randomizing polynomials and their applications. *Computational Complexity*, 15(2):115–162, 2006.

[3] László Babai. Trading group theory for randomness. In *17th Annual ACM Symposium on Theory of Computing*, pages 421–429. ACM Press, May 1985.

[4] Theodore P. Baker, John Gill, and Robert Solovay. Relativizatons of the P =? NP question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

[5] Boaz Barak. How to go beyond the black-box simulation barrier. In *42nd Annual Symposium on Foundations of Computer Science*, pages 106–115. IEEE Computer Society Press, October 2001.

[6] Boaz Barak. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In *43rd Annual Symposium on Foundations of Computer Science*, pages 345–355. IEEE Computer Society Press, November 2002.

[7] Boaz Barak and Yehuda Lindell. Strict polynomial-time in simulation and extraction. In *34th Annual ACM Symposium on Theory of Computing*, pages 484–493. ACM Press, May 2002.

[8] Boaz Barak, Yehuda Lindell, and Salil P. Vadhan. Lower bounds for non-black-box zero knowledge. *Journal of Computer and System Sciences*, 72(2):321–391, 2006.

[9] Boaz Barak and Mohammad Mahmoody-Ghidary. Lower bounds on signatures from symmetric primitives. In *48th Annual Symposium on Foundations of Computer Science*, pages 680–688. IEEE Computer Society Press, October 2007.

[10] Boaz Barak and Mohammad Mahmoody-Ghidary. Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 374–390. Springer, August 2009.

[11] Donald Beaver. Correlated pseudorandomness and the complexity of private computations. In *28th Annual ACM Symposium on Theory of Computing*, pages 479–488. ACM Press, May 1996.

[12] Mihir Bellare and Shafi Goldwasser. New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 194–211. Springer, August 1990.

[13] Mihir Bellare, Markus Jakobsson, and Moti Yung. Round-optimal zero-knowledge arguments based on any one-way function. In Walter Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 280–305. Springer, May 1997.

[14] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. Perfect zero-knowledge in constant rounds. In *22nd Annual ACM Symposium on Theory of Computing*, pages 482–493. ACM Press, May 1990.

[15] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73. ACM Press, November 1993.

[16] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE Symposium on Security and Privacy*, pages 321–334. IEEE Computer Society Press, May 2007.

[17] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proc. ICM*, 1986.

[18] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4):850–864, 1984.

[19] Alexandra Boldyreva, David Cash, Marc Fischlin, and Bogdan Warinschi. Foundations of non-malleable hash and one-way functions. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 524–541. Springer, December 2009.

[20] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer, May 2005.

[21] Dan Boneh and Matthew K. Franklin. Identity based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.

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

[23] Dan Boneh and Ramarathnam Venkatesan. Breaking RSA may not be equivalent to factoring. In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 59–71. Springer, May / June 1998.

[24] Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 535–554. Springer, February 2007.

[25] Zvika Brakerski, Shafi Goldwasser, Guy N. Rothblum, and Vinod Vaikuntanathan. Weak verifiable random functions. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 558–576. Springer, March 2009.

[26] Zvika Brakerski, Jonathan Katz, Gil Segev, and Arkady Yerukhimovich. Limits on the power of zero-knowledge proofs in cryptographic constructions. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 559–578. Springer, March 2011.

[27] Gilles Brassard, Claude Crépeau, and Moti Yung. Everything in NP can be argued in perfect zero-knowledge in a bounded number of rounds (extended abstract). In Jean-Jacques Quisquater and Joos Vandewalle, editors, *Advances in Cryptology – EUROCRYPT'89*, volume 434 of *Lecture Notes in Computer Science*, pages 192–195. Springer, April 1990.

[28] Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. Separation results on the "one-more" computational problems. In Tal Malkin, editor, *Topics in Cryptology – CT-RSA 2008*, volume 4964 of *Lecture Notes in Computer Science*, pages 71–87. Springer, April 2008.

[29] Daniel R. L. Brown. Irreducibility to the one-more evaluation problems: More may be less. Cryptology ePrint Archive, Report 2007/435, 2007. http://eprint.iacr.org/.

[30] Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. *Journal of Cryptology*, 20(3):265–294, July 2007.

[31] Yan-Cheng Chang, Chun-Yuan Hsiao, and Chi-Jen Lu. The impossibility of basing one-way permutations on central cryptographic primitives. *Journal of Cryptology*, 19(1):97–114, January 2006.

[32] Jean-Sébastien Coron. Optimal security proofs for PSS and other signature schemes. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 272–287. Springer, April / May 2002.

[33] Ronald Cramer, Goichiro Hanaoka, Dennis Hofheinz, Hideki Imai, Eike Kiltz, Rafael Pass, Abhi Shelat, and Vinod Vaikuntanathan. Bounded CCA2-secure encryption. In Kaoru Kurosawa, editor, *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 502–518. Springer, December 2007.

[34] Dana Dachman-Soled, Yehuda Lindell, Mohammad Mahmoody, and Tal Malkin. On the black-box complexity of optimally-fair coin tossing. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 450–467. Springer, March 2011.

[35] Yevgeniy Dodis, Roberto Oliveira, and Krzysztof Pietrzak. On the generic insecurity of the full domain hash. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 449–466. Springer, August 2005.

[36] Yevgeniy Dodis and Leonid Reyzin. On the power of claw-free permutations. In Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano, editors, *SCN 02: 3rd International Conference on Security in Communication Networks*, volume 2576 of *Lecture Notes in Computer Science*, pages 55–73. Springer, September 2002.

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

[38] U. Feige, D. Lapidot, and A. Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st Annual Symposium on Foundations of Computer Science*, pages 308–317. IEEE Computer Society Press, October 1990.

[39] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988.

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

[41] Uriel Feige and Adi Shamir. Zero knowledge proofs of knowledge in two rounds. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 526–544. Springer, August 1990.

[42] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 480–491. Springer, August 1994.

[43] Marc Fischlin. On the impossibility of constructing non-interactive statistically-secret protocols from any trapdoor one-way function. In Bart Preneel, editor, *Topics in Cryptology – CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 79–95. Springer, February 2002.

[44] Marc Fischlin. Round-optimal composable blind signatures in the common reference string model. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 60–77. Springer, August 2006.

[45] Marc Fischlin and Dominique Schröder. On the impossibility of three-move blind signature schemes. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 197–215. Springer, May 2010.

[46] Lance Fortnow. The complexity of perfect zero-knowledge (extended abstract). In Alfred Aho, editor, *19th Annual ACM Symposium on Theory of Computing*, pages 204–209. ACM Press, May 1987.

[47] Sanjam Garg, Raghav Bhaskar, and Satyanarayana V. Lokam. Improved bounds on security reductions for discrete log based signatures. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 93–107. Springer, August 2008.

[48] Rosario Gennaro, Yael Gertner, and Jonathan Katz. Lower bounds on the efficiency of encryption and digital signature schemes. In *35th Annual ACM Symposium on Theory of Computing*, pages 417–425. ACM Press, June 2003.

[49] Rosario Gennaro, Yael Gertner, Jonathan Katz, and Luca Trevisan. Bounds on the efficiency of generic cryptographic constructions. *SIAM Journal on Computing*, 35(1):217–246, 2005.

[50] Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st Annual Symposium on Foundations of Computer Science*, pages 305–313. IEEE Computer Society Press, November 2000.

[51] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 169–178. ACM Press, May / June 2009.

[52] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st Annual Symposium on Foundations of Computer Science*, pages 325–335. IEEE Computer Society Press, November 2000.

[53] Yael Gertner, Tal Malkin, and Steven Myers. Towards a separation of semantic and CCA security for public key encryption. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 434–455. Springer, February 2007.

[54] Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *42nd Annual Symposium on Foundations of Computer Science*, pages 126–135. IEEE Computer Society Press, October 2001.

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

[56] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33:792–807, 1986.

[57] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, 1996.

[58] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.

[59] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *21st Annual ACM Symposium on Theory of Computing*, pages 25–32. ACM Press, May 1989.

[60] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 171–185. Springer, August 1987.

[61] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991.

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

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

[64] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *18th Annual ACM Symposium on Theory of Computing*, pages 59–68. ACM Press, May 1986.

[65] S. Dov Gordon, Hoeteck Wee, David Xiao, and Arkady Yerukhimovich. On the round complexity of zero-knowledge proofs based on one-way permutations. In Michel Abdalla and Paulo S. L. M. Barreto, editors, *Progress in Cryptology - LATINCRYPT 2010: 1st International Conference on Cryptology and Information Security in Latin America*, volume 6212 of *Lecture Notes in Computer Science*, pages 189–204. Springer, August 2010.

[66] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06: 13th Conference on Computer and Communications Security*, pages 89–98. ACM Press, October / November 2006. Available as Cryptology ePrint Archive Report 2006/309.

[67] Satoshi Hada and Toshiaki Tanaka. On the existence of 3-round zero-knowledge protocols. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 408–423. Springer, August 1998.

[68] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In *48th Annual Symposium on Foundations of Computer Science*, pages 669–679. IEEE Computer Society Press, October 2007.

[69] Iftach Haitner, Jonathan J. Hoch, and Gil Segev. A linear lower bound on the communication complexity of single-server private information retrieval. In Ran Canetti, editor, *TCC 2008: 5th Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 445–464. Springer, March 2008.

[70] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 202–219. Springer, March 2009.

[71] Iftach Haitner, Mohammad Mahmoody, and David Xiao. A new sampling protocol and applications to basing cryptographic primitives on the hardness of NP. In *IEEE Conference on Computational Complexity*, pages 76–87. IEEE Computer Society, 2010.

[72] Iftach Haitner, Omer Reingold, Salil P. Vadhan, and Hoeteck Wee. Inaccessible entropy. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 611–620. ACM Press, May / June 2009.

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

[74] Dennis Hofheinz. Possibility and impossibility results for selective decommitments. *Journal of Cryptology*, 24(3):470–516, July 2011.

[75] Chun-Yuan Hsiao and Leonid Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 92–105. Springer, August 2004.

[76] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st Annual ACM Symposium on Theory of Computing*, pages 44–61. ACM Press, May 1989.

[77] Jeff Kahn, Michael E. Saks, and Clifford D. Smyth. A dual version of reimer's inequality and a proof of rudich's conjecture. In *IEEE Conference on Computational Complexity*, pages 98–103, 2000.

[78] Jonathan Katz. Which languages have 4-round zero-knowledge proofs? In Ran Canetti, editor, *TCC 2008: 5th Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 73–88. Springer, March 2008.

[79] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.

[80] Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Round efficiency of multi-party computation with a dishonest majority. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 578–595. Springer, May 2003.

[81] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 146–162. Springer, April 2008.

[82] Jonathan Katz, Dominique Schröder, and Arkady Yerukhimovich. Impossibility of blind signatures from one-way permutations. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 615–629. Springer, March 2011.

[83] Jonathan Katz and Arkady Yerukhimovich. On black-box constructions of predicate encryption from trapdoor permutations. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 197–213. Springer, December 2009.

[84] Eike Kiltz, Payman Mohassel, and Adam O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 673–692. Springer, May 2010.

[85] Jeong Han Kim, Daniel R. Simon, and Prasad Tetali. Limits on the efficiency of one-way permutation-based hash functions. In *40th Annual Symposium on Foundations of Computer Science*, pages 535–542. IEEE Computer Society Press, October 1999.

[86] Leonid A. Levin. Universal sequential search problems. *Problems Inform. Transmission*, 9(3):265–266, 1973.

[87] Yehuda Lindell. Bounded-concurrent secure two-party computation without setup assumptions. In *35th Annual ACM Symposium on Theory of Computing*, pages 683–692. ACM Press, June 2003.

[88] Yehuda Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 241–254. Springer, May 2003.

[89] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. In *31st Annual Symposium on Foundations of Computer Science*, pages 2–10. IEEE Computer Society Press, October 1990.

[90] Takahiro Matsuda and Kanta Matsuura. On black-box separations among injective one-way functions. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 597–614. Springer, March 2011.

[91] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991.

[92] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *21st Annual ACM Symposium on Theory of Computing*, pages 33–43. ACM Press, May 1989.

[93] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd Annual ACM Symposium on Theory of Computing*. ACM Press, May 1990.

[94] R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *ISTCS '93*, pages 3–17, 1993.

[95] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based encryption with non-monotonic access structures. In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM CCS 07: 14th Conference on Computer and Communications Security*, pages 195–203. ACM Press, October 2007.

[96] Pascal Paillier and Damien Vergnaud. Discrete-log-based signatures may not be equivalent to discrete log. In Bimal K. Roy, editor, *Advances in Cryptology – ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 1–20. Springer, December 2005.

[97] Pascal Paillier and Jorge L. Villar. Trading one-wayness against chosen-ciphertext security in factoring-based encryption. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 252–266. Springer, December 2006.

[98] Periklis A. Papakonstantinou. Constructions, lower bounds, and new directions in cryptography and computational complexity. CS Ph.D Thesis, University of Toronto, 2010. `http://itcs.tsinghua.edu.cn/~papakons/pdfs/phd_thesis.pdf`.

[99] Rafael Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. In László Babai, editor, *36th Annual ACM Symposium on Theory of Computing*, pages 232–241. ACM Press, June 2004.

[100] Rafael Pass. Parallel repetition of zero-knowledge proofs and the possibility of basing cryptography on NP-hardness. In *IEEE Conference on Computational Complexity*, pages 96–110. IEEE Computer Society, 2006.

[101] 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*, volume 4117 of *Lecture Notes in Computer Science*, pages 271–289. Springer, August 2006.

[102] Rafael Pass and Alon Rosen. Bounded-concurrent secure two-party computation in a constant number of rounds. In *44th Annual Symposium on Foundations of Computer Science*, pages 404–415. IEEE Computer Society Press, October 2003.

[103] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *46th Annual Symposium on Foundations of Computer Science*, pages 563–572. IEEE Computer Society Press, October 2005.

[104] Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Towards non-black-box lower bounds in cryptography. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 579–596. Springer, March 2011.

[105] Rafael Pass and Muthuramakrishnan Venkitasubramaniam. Private coins versus public coins in zero-knowledge proof systems. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 588–605. Springer, February 2010.

[106] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 403–418. Springer, March 2009.

[107] Aduri Pavan, Alan L. Selman, Samik Sengupta, and N. V. Vinodchandran. Polylogarithmic-round interactive proofs for coNP collapse the exponential hierarchy. *Theoretical Computer Science*, 385(1-3):167–178, 2007.

[108] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer, February 2004.

[109] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd Annual ACM Symposium on Theory of Computing*, pages 387–394. ACM Press, May 1990.

[110] Alon Rosen. A note on constant-round zero-knowledge proofs for NP. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 191–202. Springer, February 2004.

[111] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 419–436. Springer, March 2009.

[112] Steven Rudich. Limits on the provable consequences of one-way functions. CS Ph.D Thesis, Berkeley, 1988. http://www.cs.cmu.edu/~rudich/papers/thesis.ps.

[113] Steven Rudich. The use of interaction in public cryptosystems (extended abstract). In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 242–251. Springer, August 1992.

[114] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science*, pages 543–553. IEEE Computer Society Press, October 1999.

[115] Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, May 2005.

[116] Adi Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology – CRYPTO'84*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, August 1985.

[117] 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*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345. Springer, May / June 1998.

[118] Dominique Unruh. Random oracles and auxiliary input. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 205–223. Springer, August 2007.

[119] Yevgeniy Vahlis. Two is a crowd? a black-box separation of one-wayness and security under correlated inputs. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 165–182. Springer, February 2010.

[120] Hoeteck Wee. One-way permutations, interactive hashing and statistically hiding commitments. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 419–433. Springer, February 2007.

[121] Andrew C. Yao. Theory and applications of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91. IEEE Computer Society Press, November 1982.

[122] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science*, pages 162–167. IEEE Computer Society Press, October 1986.