Unconditional Relationships within Zero Knowledge

Citation	Ong, Shien Jin. 2007. Unconditional Relationships within Zero Knowledge. Doctoral dissertation, Harvard University.
Citable link	http://nrs.harvard.edu/urn-3:HUL.InstRepos:5128473
Terms of Use	This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA

Dissertation Advisor: Professor Salil P. Vadhan **Shien Jin Ong**

Unconditional Relationships within Zero Knowledge

Abstract

Zero-knowledge protocols enable one party, called the *prover*, to convince another party, called the *verifier*, the validity of a mathematical statement such that the verifier learns nothing other than the fact that the proven statement is true. The different ways of formulating the terms "convince" and "learns nothing" give rise to four classes of languages having zero-knowledge protocols, which are: statistical zero-knowledge proof systems, computational zero-knowledge proof systems, statistical zero-knowledge argument systems, and computational zero-knowledge argument systems.

We establish complexity-theoretic characterizations of these four zero-knowledge complexity classes, of which our characterizations for argument systems are novel. Using these characterizations, we show that for languages in NP, the following hold.

- Instance-dependent commitment schemes are necessary and sufficient for zero-knowledge protocols. Instance-dependent commitment schemes for a given language are commitment schemes that can depend on the instance of the language, and where the hiding and binding properties are required to hold only on the YES and NO instances of the language, respectively.
- Computational zero knowledge and computational soundness (a property held by argument systems) are symmetric properties. Namely, we show that the class of languages in NP $\cap$ co-NP having zero-knowledge arguments is closed under complement, and that a language in NP has a statistical zero-knowledge *argument* system if and only if its complement has a *computational* zero-knowledge proof system.
- Any zero-knowledge argument system that is only guaranteed to be secure against the *honest verifier* that follows the prescribed protocol can be transformed into one that is secure against *malicious verifiers* that can deviate from the protocol. In addition, our transformation gives us zero-knowledge argument systems with desirable properties like public coins, perfect completeness, a black-box simulator, and an efficient prover.

The novelty of our results above is that they are *unconditional*, meaning that they do not rely on any unproven complexity assumptions such as the existence of one-way functions. Moreover, in establishing our complexity-theoretic characterizations, we give the first construction of statistical zero-knowledge argument systems for all of NP based on any one-way function.

CONTENTS

PREVIOUSLY PUBLISHED WORK

Most of the research results in this dissertation have appeared in conference proceedings. Large portions of Chapter 3 are based on a joint with Minh-Huyen Nguyen and Salil Vadhan, which appeared as:

> "Statistical Zero-Knowledge Arguments for NP from Any One-Way Function," in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006)*, pages 3–14, IEEE Computer Society, 2006. *Invited to SIAM Journal on Computing Special Issue on FOCS 2006.*

Section 3.6 in Chapter 3 is joint work with Iftach Haitner, Omer Reingold, and Salil Vadhan, but has not been published.

Chapter 4 is based on a joint work with Salil Vadhan, and appeared as:

> "Zero Knowledge and Soundness are Symmetric," in *Advances in Cryptology – EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 187–209, Lecture Notes in Computer Science 4515, Springer, 2007. *Winner of the Best Paper Award.*

In addition, parts of Chapters 1 and 2 are also based on the above *EUROCRYPT 2007* paper.

ACKNOWLEDGEMENTS

I am grateful to:

Salil Vadhan, my research advisor, mentor, and friend—much that I learned about research, I learnt from Salil;

Michael Sipser, my MIT undergraduate advisor, who through his lucidly written book, *Introduction to the Theory of Computation* [Sip], and excellent teaching, sparked my interest in theoretical computer science;

Silvio Micali, the master of "cryptographic" intuition, who illustrated the beauty of cryptography through his engaging lectures;

my research collaborators: Boaz Barak, Yevgeniy Dodis, Daniele Micciancio, Minh-Huyen Nguyen, David Parkes, Manoj Prabhakaran, Alon Rosen, Amit Sahai, and Salil Vadhan;

members of my dissertation examining committee: Harry Lewis, Michael Rabin, Salil Vadhan, and Leslie Valiant;

fellow theoretical computer science graduate students and postdoctoral fellows at Harvard: Kai-Min Chung, Eleni Drinea, Vitaly Feldman, Dan Gutfreund, Alexander Healy, Shaili Jain, Adam Kirsch, Loizos Michael, Minh-Huyen Nguyen, Alon Rosen, and Emanuele Viola;

Susan Wieczorek, my school's amazingly dedicated graduate program administrator, who is an oracle for all obscure and not-so-obscure matters of graduate school bureaucracy;

my caring parents, Ong Chong Wee and Ooi Poh Yean, who instilled in me the love of knowledge;

my girlfriend, Liu Lian, for her support through this lengthy dissertation writing process.

To my parents who taught me the meaning of life,

and

to my dearest girlfriend, Lian, who gave me meaning to life.

1

INTRODUCTION

Suppose you solved a famous mathematical problem—like the P versus NP problem, one of the seven Millennium Prize Problems proposed by the Clay Mathematics Institute.[1] Will you be able to convince someone of the proof without letting the person steal your idea and claim entitlement to your proof?

The answer, surprisingly, is yes; ***zero-knowledge protocols***, introduced by Goldwasser, Micali, and Rackoff [GMR1], provide a way for one party, called the ***prover***, to *convince* another party, called the ***verifier***, the validity of a *statement* such that the verifier *learns nothing* other than the fact that the proven statement is true. In particular, even after interacting with the prover, the verifier does not gain the ability to reprove the statement to other parties!

In this dissertation, we study zero-knowledge protocols in a complexity-theoretic framework; in other words, we investigate the classes of *languages*[2] having zero-knowledge protocols. We study zero-knowledge protocols because in addition to being fascinating mathematical objects of study, zero-knowledge protocols have vast applicability in cryptography and complexity theory; for their vast applicability, we refer the reader to a comprehensive survey on zero knowledge by Goldreich [Gol3]. Furthermore, the study of zero knowledge has led to many developments in seemingly unrelated areas of theoretical computer science like the field of inapproximability. For instance, it can be shown that no polynomial-time

[1] The P versus NP problem, arguably the most important unsolved problem in computer science, asks whether the class P of languages that can be solved efficiently equals the class NP of languages that can be verified efficiently. It is widely believed that P $\neq$ NP; indeed most results in cryptography and complexity theory would be either trivial or be not applicable if P = NP.

[2] A ***language*** is used to describe a collection of valid statements that relate to a certain algorithmic task. Refer to Section 1.1 for an informal description, and to Section 2.2.1 for a formal description.

algorithm can always find a *vertex cover*[3] of a given graph that is at most 1.37 times larger than the minimum vertex cover, unless P = NP [DS]. On the other hand, a simple greedy algorithm will always find a vertex cover that is at most twice the size of the optimal. The way these inapproximability results, like the one for vertex cover, come about was through the study of *probabilistically checkable proofs* (cf., [FGL$^+$, AS, ALM$^+$]), which in turn was inspired by zero-knowledge protocols (cf., [BOGKW, FRS]). In conclusion, zero knowledge has—and we predict will continue to—serve as a bridge between complexity theory, cryptography, and other areas of theoretical computer science.

Chapter organization. An overview of zero-knowledge protocols is provided in the next section. In Section 1.2, we highlight the main contributions of this dissertation. In Section 1.3, we state our work in perspective with previous results. We conclude this chapter with Section 1.4, detailing our agenda for the remaining chapters.

1.1 An Overview of Zero Knowledge

In this section, we give an overview of zero knowledge protocols; a more detailed and formal discussion is provided in Section 2.3. The material presented in this section is inspired by Vadhan's thesis [Vad1] and survey [Vad2], and by Sipser's book [Sip].

Languages

In theoretical computer science, a ***language*** is used to describe a collection of valid statements that relate to a certain algorithmic task. For example, the task of determining whether two graphs are *isomorphic*[4] can be formulated by the following language:

$$\textsc{Graph Isomorphism} = \{(G, H) : G \text{ and } H \text{ are isomorphic}\} \ .$$

Thus, we can think of a language L as defining the following algorithmic task: given a string x, determine if $x \in L$ or $x \notin L$.

NP proof systems

Recall the brief discussion we had earlier about the P versus NP problem. The class P is the set of languages L such that the algorithmic task of determining if $x \in L$ can be done ***efficiently***: taking a polynomial number of computation steps. It is not known if the language Graph Isomorphism is in P; the naive method of trying all possible permutations of the nodes in the graph G and then comparing it to H is very inefficient. Nevertheless, if we are given a permutation that reorders the nodes of G so that it matches H, then we can

[3] A ***vertex cover*** of a graph is a subset of nodes where every edge of that graph touches one of those nodes. Finding the minimum-size vertex cover in a given graph is NP-hard [Kar].

[4] Graphs G and H are ***isomorphic*** if the nodes of G may be permuted so that it is identical to H.

easily verify that G and H are indeed isomorphic graphs. We classify languages possessing this *efficient verifiability* property as the class NP.

We can therefore think of NP languages as having two entities: a ***prover*** that finds a valid proof for a given statement, and an efficient ***verifier*** that checks the proof. While the verifier must be efficient, the prover is allowed to be *computationally unbounded* since finding a valid proof might be tedious. More precisely, a language L is in NP if there exists a prover P and an efficient verifier V satisfying the following two conditions.

- *Completeness*: for every $x \in L$ (valid statement), the proof provided by prover P makes verifier V accept.
- *Soundness*: for every $x \notin L$ (invalid statement), no proof, even ones concocted by cheating provers, will convince verifier V to accept.

We call (P, V) satisfying the above conditions an ***NP proof system*** for the language L.

Zero-knowledge protocols

Putting this in context of Graph Isomorphism, the NP proof system that we hinted to above is a prover P that sends a permutation π of the nodes in G so that it matches H, and a verifier V that accepts only if $\pi(G) = H$.[5] The permutation π provided by P, however, reveals a lot of knowledge: it gives the verifier V the ability to prove to some other party that G and H are isomorphic graphs since V now knows a valid permutation. This limitation is inherent in all traditional mathematical proofs—where a *static proof* is written down to be checked—since knowing that static proof would allow one to reprove it to some other party. Therefore, in order to get around this bottleneck, Goldwasser, Micali, and Rackoff [GMR1], in their original treatise on zero knowledge, provided two additional features to the prover and the verifier. First, these two entities are allowed to be ***interactive***: the prover and the verifier exchange messages in multiple rounds, after which the verifier decides to accept or reject. Second, these two entities are allowed to be ***probabilistic***: the verifier can send random *challenges* to the prover, and the prover can respond with random *answers*. Hence, in this setting, the verifier will be convinced of a correct proof only with a high degree of confidence, but not with absolute certainty. If either of these two features—interactive or probabilistic—is missing, it is impossible to achieve zero knowledge for nontrivial languages [GO]. To accommodate these two features, the completeness and soundness conditions for zero-knowledge protocols are made probabilistic as follows.

- *Completeness*: for every $x \in L$ (valid statement), prover P *interacting* with verifier V will make V accept with at least 99% probability.

[5] $\pi(G)$ denotes the graph obtained by reordering the nodes of graph G according to permutation π.

- *Soundness*: for every $x \notin L$ (invalid statement), no cheating prover P^* can make V accept with probability greater than 1%. In other words, V will reject with at least 99% probability, no matter what strategy the prover pursues.

The above two conditions by themselves do not yet capture the essence of a zero knowledge, which is the magical property of the verifier *learning nothing* from its interaction with the prover, other than the fact that the statement proven is true. This guarantee of learning nothing is formalized in [GMR1] by requiring the existence of an efficient algorithm, called a ***simulator***, whose output is indistinguishable from the verifier's ***view of the interaction*** with the prover, where the verifier's view consists of all messages exchanged between the prover and verifier and the verifier's random coins. (Unlike the verifier, the simulator does not have access to the prover.) Intuitively, the verifier learns nothing because it could run the simulator instead of interacting with the prover. Thus, (P, V) is said to be a ***zero-knowledge protocol*** if it satisfies the completeness and soundness conditions above, and an additional condition below.

- *Zero Knowledge*: for every $x \in L$ (valid statement), and every efficient (cheating) verifier V^*, there exists an efficient simulator S whose output is indistinguishable from the V^*'s view of the interaction with the prover P.

The zero-knowledge condition is required to hold only when $x \in L$ because the prover only provides proofs for valid statements. In addition, we have required that even verifiers deviating from the prescribed protocol will learn nothing. This requirement is generally what is needed in cryptography because we cannot assume parties to act as prescribed. There is a weaker notion of zero knowledge, called ***honest-verifier zero knowledge***, which guarantees that only an honest verifier that follows the prescribed protocol learns nothing; we explore this notion in Section 2.3.

Flavors of zero-knowledge protocols

Zero-knowledge protocols come in several flavors, depending on how one formulates the two security conditions: (i) the zero-knowledge condition, which says that the verifier learns nothing other than the fact the assertion being proven is true, and (ii) the soundness conditions, which says that the prover cannot convince the verifier of a false assertion. We call these security conditions because they these two conditions need to hold even against cheating parties; the former against cheating verifiers, and the latter against cheating provers. In contrast, the completeness condition is not considered a security condition because it only refers to honest provers and verifiers that follow the prescribed protocol.

In ***statistical zero knowledge***, the zero-knowledge condition holds regardless of the computational resources the verifier invests into trying to learn something from the interaction (except with small probability).[6] In ***computational zero knowledge***, we only require that efficient, probabilistic polynomial-time verifiers learn nothing from the interaction.[7] Similarly, for soundness, we have ***statistical soundness***, giving rise to ***proof systems***, where even a computationally unbounded prover cannot convince the verifier of a false statement, and ***computational soundness***, giving rise to ***argument systems*** [BCC], where we only require that an efficient, probabilistic polynomial-time prover cannot convince the verifier of a false statement. Using a prefix of S or C to indicate whether the zero knowledge is statistical or computational and a suffix of P or A to indicate whether we have a proof system or argument system, we obtain four complexity classes corresponding to the different types of zero-knowledge protocols: SZKP, CZKP, SZKA, and CZKA.

We return to our question of whether the NP-language Graph Isomorphism has a zero-knowledge protocol. If we make complexity assumptions, then it turns out that *every* language in NP has zero-knowledge protocols where at least one of the security conditions is computational: namely, NP $\subseteq$ CZKP [GMW2] and NP $\subseteq$ SZKA [BCC].[8] (This also implies that NP $\subseteq$ CZKA since CZKA has the weakest securities properties, and hence contains all the other zero-knowledge complexity classes.) In other words, assuming widely believed but unproven complexity assumptions, every traditional mathematical proof can be argued convincingly in a way that does not leak knowledge.

Our goal in this dissertation is to conduct a study on these four zero-knowledge complexity classes—SZKP, CZKP, SZKA, and CZKA—without relying on unproven complexity assumptions. For example, if a language L is in CZKA, what can be said of its *complement* $\overline{L} = \{x : x \notin L\}$? This will be answered in Section 1.2, where we also present our main contributions, but before going there, we highlight the close relationship between zero-knowledge protocols and a cryptographic primitive called *commitment schemes*.

Commitment schemes

A ***commitment scheme*** is a two-stage protocol between a sender and a receiver. In the first stage, called the ***commit stage***, the sender *commits* to a private message m. In the second stage, called the ***reveal stage***, the sender reveals m and *proves* that it was the message to which she committed in the first stage. We require two properties of commitment schemes. The ***hiding*** property says that an adversarial receiver learns nothing about m in the commit stage. The ***binding*** property says that after the commit stage, an adversarial

[6] There is a third flavor of zero knowledge which is the strongest of all: ***perfect zero knowledge***, where the verifier cannot learn anything even with negligible probability. In this dissertation, we do *not* study the distinction between perfect zero knowledge and statistical zero knowledge. See Section 2.3.4 for a more detailed discussion.

[7] More precisely, in statistical zero knowledge, we require that the verifier's view of the interaction can be efficiently simulated up to negligible statistical distance, whereas in computational zero knowledge, we only require that the simulation be computationally indistinguishable from the verifier's view.

[8] While *statistical* zero-knowledge *proofs* are most desirable to have, it is unlikely that every language in NP has them [For, AH, BHZ].

sender is bound to a particular value of m: namely, she cannot successfully open the commitment to two different bits in the reveal stage. Like the zero knowledge and soundness conditions, the hiding and binding properties each come in two flavors: (i) ***statistical***, where the property holds regardless of the adversary's computational resources, and (ii) ***computational***, where the property holds only for efficient, probabilistic polynomial-time adversaries.

Earlier we mentioned that if we make complexity assumptions, then every language in NP has a zero-knowledge protocol. This result was first proven by Goldreich, Micali, and Wigderson [GMW2], who constructed computational zero-knowledge proof systems for every language in NP. Their zero-knowledge protocol uses commitment schemes—specifically, of the computationally-hiding and statistically-binding flavor—and their usage of commitment schemes is precisely the reason why their protocol requires complexity assumptions; commitment schemes are not known to exist unconditionally, and indeed the existence of commitment schemes implies the existence of one-way functions [IL].[9]

Nevertheless, as observed by Itoh, Ohta, and Shizuya [IOS], zero-knowledge protocols based on commitment schemes actually do not require the hiding and binding properties of commitments to hold at the same time. In the Goldreich, Micali & Wigderson protocol [GMW2], and many other zero-knowledge protocols, the hiding and binding properties of the commitment schemes used translate to the zero knowledge and soundness conditions, respectively. Hence, the hiding property is only required when $x \in L$, and the binding property is only required when $x \notin L$.

Based on this observation, Itoh, Ohta, and Shizuya defined ***instance-dependent commitment schemes*** to be analogues of commitments schemes that are tailored specifically to a given language.[10] More precisely, the sender and receiver of an instance-dependent commitment scheme receive an instance x of a language L as auxiliary input, and the scheme is required to be hiding when $x \in L$ and be binding when $x \notin L$. Thus, instance-dependent commitment schemes are a relaxation of standard commitment schemes, since we do not require that the hiding and binding properties hold at the same time. This relaxation, however, is still useful in constructing zero-knowledge protocols for $L \in$ NP, due to the following two observations: (i) the hiding and binding properties of the commitment schemes used in some zero-knowledge protocols translate to the zero knowledge and soundness conditions, respectively, and (ii) the zero knowledge and soundness conditions are only required when $x \in L$ and $x \notin L$, respectively. Therefore, we conclude this section

[9] ***One-way functions*** are functions that are easy to compute but are computationally infeasible to invert. Assuming that factoring large integers is computationally infeasible, the multiplication function $f(x, y) = x \cdot y$ is a one-way function. Although many researchers believe that one-way functions exist, a proof of their existence who give a proof that P $\neq$ NP, resolving arguably the most important unsolved problem in computer science. A formal definition of one-way functions is given in Section 2.4.1.

[10] There were various terms used to describe *instance-dependent commitment schemes*. Itoh, Ohta, and Shizuya [IOS] called these *language-dependent cryptographic primitives*, Micciancio and Vadhan [MV] called these *problem-dependent commitment schemes*, and the present usage traces to Vadhan [Vad3].

by stressing that having an instance-dependent commitment scheme for a language $L \in \mathrm{NP}$ suffices to *unconditionally* construct a zero-knowledge protocol for L.

1.2 Contributions of this Dissertation

In this section, we highlight several new research results contained in this dissertation.

1.2.1 Necessity of instance-dependent commitments

At the end of the previous section, we saw that an instance-dependent commitment for a language $L \in \mathrm{NP}$ suffices to construct a zero-knowledge protocol for L, based on techniques from [GMW2, IOS]. In this dissertation, we show that the converse holds: an instance-dependent commitments for a language $L \in \mathrm{NP}$ is necessary to obtain a zero-knowledge protocol for L.

THEOREM 1.2.1

1. The SZKP case: a language $L \in \mathrm{NP}$ has a statistical zero-knowledge proof system if and only if L has an instance-dependent commitment scheme that is statistically hiding when $x \in L$ and statistically binding when $x \notin L$.

2. The CZKP case: a language $L \in \mathrm{NP}$ has a computational zero-knowledge proof system if and only if L has an instance-dependent commitment scheme that is computationally hiding when $x \in L$ and statistically binding when $x \notin L$.

3. The SZKA case: a language $L \in \mathrm{NP}$ has a statistical zero-knowledge argument system if and only if L has an instance-dependent commitment scheme that is statistically hiding when $x \in L$ and computationally binding when $x \notin L$.

4. The CZKA case: a language $L \in \mathrm{NP}$ has a computational zero-knowledge argument system if and only if L has an instance-dependent commitment scheme that is computationally hiding when $x \in L$ and computationally binding when $x \notin L$.

This theorem will be proven in Chapter 4; specifically, it will follow from Theorems 4.1.1, 4.1.2, 4.1.4, and 4.1.5 in Section 4.1. This theorem can be viewed as demonstrating the centrality of commitment schemes in zero-knowledge protocols of NP. To paraphrase Damgård [Dam2, p. 19], many researchers intuitively believe that commitment schemes are fundamental to the construction of zero-knowledge protocols. Hence, we made this intuition—held by many researchers—precise.

Prior to our work, Vadhan [Vad3] constructed instance-dependent commitments, albeit with an inefficient (i.e., exponential time) sender, for languages with zero-knowledge *proofs*. For these same languages, Nguyen and Vadhan [NV] constructed instance-dependent commitments, whose sender and receiver algorithms are efficient (i.e., polynomial time), but paid a price of obtaining commitments with a weaker and nonstandard *1-out-of-2* binding property. Instance-dependent commitments for a restricted class of zero-knowledge proofs, namely *3-round public-coin* zero-knowledge proofs, were implicit in the works of Damgård [Dam2, Dam3]. Indeed, Kapron, Malka, and Srinivasan [KMS] used Damgård's techniques to show that 3-round public-coin zero-knowledge proofs where the verifier just sends a random bit—called *V-bit protocols*—exactly characterize *noninteractive* instance-dependent commitments.[11] Finally, we note that Ostrovsky and Wigderson [OW] showed that zero-knowledge proofs for *hard-on-average* languages imply one-way functions, and hence standard commitment schemes [Nao, HILL]. Consequently, our results are the first to achieve instance-dependent commitments with standard properties—such as an efficient sender and a standard binding property—from general zero-knowledge complexity classes.

1.2.2 Symmetry between zero knowledge and soundness

Recall that the two security conditions for zero-knowledge protocols are the zero-knowledge condition and the soundness condition. These two security conditions have different traits; zero knowledge is a *secrecy* condition, whereas soundness is more like an *unforgeability* condition. Nevertheless, in a remarkable paper, Okamoto [Oka] established a symmetry between *statistical zero knowledge* and *statistical soundness* by proving that a language $L \in$ SZKP if and only if its complement $\overline{L} \in$ SZKP.[12] We view this as a symmetry result because the statistical zero knowledge property for L—which holds when $x \in L$ or equivalently, when $x \notin \overline{L}$—translates to a statistical soundness property for $\overline{L}$, and vice-versa. Thus, by showing that SZKP is closed under complement, Okamoto established a symmetry between zero knowledge and soundness, in the case when both security conditions are statistical.

We ask whether an analogous theorem holds when the security conditions are *computational*, namely when considering computational zero-knowledge arguments. If we make complexity assumptions, then the answer is yes, because all languages in NP $\cap$ co-NP and their complements have computational zero-knowledge arguments [GMW2, BCC]. In this dissertation, we establish an *unconditional* symmetry between computational zero knowledge and computational soundness.

[11] ***Noninteractive commitments*** are commitments where the sender commits to a message in the commit stage by sending a *single* message to the receiver; hence, the receiver does not send any message, both in the commit and reveal stages.

[12] Okamoto's result was actually for the class of languages having *honest-verifier* statistical zero-knowledge proofs, but Goldreich, Sahai, and Vadhan [GSV1] showed that this is the same as the class of languages having general, malicious-verifier statistical zero-knowledge proofs.

THEOREM 1.2.2
(Symmetry Theorem.)

- CZKA versus co-CZKA: a language $L \in \mathrm{NP} \cap \text{co-NP}$ has a computational zero-knowledge argument system if and only if its complement $\overline{L}$ has a computational zero-knowledge argument system.
- SZKA versus CZKP: a language $L \in \mathrm{NP}$ has a statistical zero-knowledge argument system if and only if its complement $\overline{L}$ has a computational zero-knowledge proof system.

The above Symmetry Theorem will be proven in Section 4.3 of Chapter 4.

Remark. On page 5, we asked if a language L is in CZKA, what can be said of its *complement* $\overline{L} = \{x : x \notin L\}$? The Symmetry Theorem answers this affirmatively, by placing $\overline{L} \in \mathrm{CZKA}$ if $L \in \mathrm{NP} \cap \text{co-NP}$.

1.2.3 Honest-verifier equals malicious-verifier zero-knowledge arguments

We show that for every language $L \in \mathrm{NP}$, any zero-knowledge argument system for L that is only guaranteed to be secure against the ***honest verifier*** that follows the prescribed protocol can be *unconditionally* transformed into one that is secure against ***malicious verifiers*** that can deviate from the protocol. In addition, our transformation gives us zero-knowledge argument systems with desirable properties like public coins, perfect completeness, a black-box simulator, and an efficient prover.[13] Previously, such results were only known under unproven complexity assumptions [GMW2, BCC], or were known unconditionally for the case of zero-knowledge *proof* systems [Oka, GSV1, Vad3, NV].

This result will be established in Chapter 4; specifically, it will be stated in Theorems 4.1.1, 4.1.2, 4.1.4, and 4.1.5 in Section 4.1.

1.2.4 New characterizations of zero-knowledge argument systems

All our unconditional results highlighted in the previous subsections are obtained by new characterizations of the classes of languages in NP having zero-knowledge argument systems. These characterizations are a generalization of the "SZK/OWF Characterization Theorem" of Vadhan [Vad3], which states that any language L having a computational zero-knowledge *proof* system can be described as having a statistical zero-knowledge proof plus a set of instances $I \subseteq L$ from which we can construct a one-way function. To characterize zero-knowledge *argument* systems, we will also allow some additional instances in the complement set $\overline{L}$ from which we can construct a one-way function. We honor the pioneering work

[13] The properties—efficient prover, perfect completeness, public coins, and black-box simulation—are defined in Section 2.3

of Vadhan that provides this useful method for characterizing zero-knowledge complexity classes by calling it the *Vadhan condition.*

DEFINITION 1.2.3

A language L satisfies the ***Vadhan condition*** if there exists a set of instances I such that:

- the *promise problem*[14] $(L \setminus I, \overline{L} \setminus I)$ is in SZKP, and
- there exists a polynomial-time computable function $f_x \colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}$, with $n(\cdot)$ and $m(\cdot)$ being polynomials and instance x given as an auxiliary input, which acts like a one-way function on every $x \in I$. That is, for every nonuniform probabilistic polynomial-time adversary A, and for every constant $c > 0$, we have

  $$\Pr_{y \leftarrow \{0,1\}^{n(|x|)}} \left[A(f_x(y)) \in f_x^{-1}(f_x(y)) \right] \leq \frac{1}{|x|^c} \ ,$$

  for every sufficiently long $x \in I$.

We call I the set of ***OWF instances***, $I \cap L$ the set of ***OWF YES instances***, and $I \cap \overline{L}$ the set of ***OWF NO instances***.

We use the Vadhan condition to characterize the four zero-knowledge complexity classes as follows.

THEOREM 1.2.4

1. The SZKP case (trivial): a language $L \in \mathrm{NP}$ has a statistical zero-knowledge proof system if and only if L satisfies the Vadhan condition without OWF instances, namely $I = \emptyset$.

2. The CZKP case ([Vad3]): a language $L \in \mathrm{NP}$ has a computational zero-knowledge proof system if and only if L satisfies the Vadhan condition without OWF NO instances, namely $I \cap \overline{L} = \emptyset$.

3. The SZKA case (*new*): a language $L \in \mathrm{NP}$ has a statistical zero-knowledge argument system if and only if L satisfies the Vadhan condition without OWF YES instances, namely $I \cap L = \emptyset$.

4. The CZKA case (*new*): a language $L \in \mathrm{NP}$ has a computational zero-knowledge argument system if and only if L satisfies the Vadhan condition.

[14] A ***promise problem*** Π consists of a pair $(\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$ of disjoint sets of strings, corresponding to YES and NO instances of Π, respectively [ESY]. All of the complexity classes that we consider—for instance, SZKP, CZKP, SZKA, and CZKA—generalize to promise problems in a natural way: completeness and zero knowledge are required for YES instances, and soundness is required for NO instances. A language L is a special case of a promise problem, by taking $\Pi = (L, \overline{L})$.

This theorem will be proven in Chapter 4; specifically, it is a succinct version of Theorems 4.1.1, 4.1.2, 4.1.4, and 4.1.5 in Section 4.1.

1.2.5 Statistical zero-knowledge arguments for NP from one-way functions

We give the first construction of statistical zero-knowledge arguments for NP based on any one-way function, as stated in the following theorem.

THEOREM 1.2.5
If one-way functions exist, then every language in NP has a statistical zero-knowledge argument system.

This theorem will be established in Section 3.5 of Chapter 3. Although this theorem is a conditional result, it is used to establish the SZKA and CZKA cases of Theorem 1.2.4, which gives *unconditional* complexity-theoretic characterizations of zero-knowledge argument systems.

Previous constructions of statistical zero-knowledge arguments for NP, starting from the construction of Brassard, Chaum, and Crépeau [BCC], require stronger complexity assumptions. Our result can be viewed as settling the complexity of statistical zero-knowledge arguments for NP because the existence of one-way functions is essentially the minimal complexity assumption needed [Ost].[15]

1.3 Motivation and Other Related Works

The research presented in this dissertation was inspired by the seminal work of Vadhan [Vad3] on the unconditional study of *computational zero-knowledge proofs*, and by the work of Nguyen and Vadhan [NV] who show that zero-knowledge *proofs* can be unconditionally converted into ones with *efficient provers*. Vadhan's work is the first unconditional study done on an entire zero-knowledge complexity class where at least one of the security conditions is computational (in his case, the zero knowledge condition is computational). Prior to Vadhan, the unconditional works in zero knowledge centered around *statistical zero-knowledge proofs*, where both the zero knowledge and soundness conditions are statistical; examples of these works include [BP, DDPY1, DC, Oka, DOY, DDPY2, SV, GSV1, GV, GSV2, Vad1, MV]. In this dissertation, we extend work of Vadhan [Vad3] and Nguyen and Vadhan [NV] to unconditionally study the more general classes of languages having zero-knowledge *arguments*, where the soundness condition is computational.

Vadhan's work, in turn, was motivated by the work of Ostrovsky and Wigderson [OW], "who gave the first hint that it might be possible to prove unconditional results about zero

[15] The result of Ostrovsky [Ost] is stated only for proof systems, but it also holds for argument systems.

knowledge" [Vad3, p. 1161]. They showed that if *hard-on-average*[16] languages have computational zero-knowledge proofs, then one-way functions exists. With one-way functions, we know that all languages in NP have computational zero-knowledge proofs [GMW2, Nao, HILL]. Thus, if we can show that NP $\not\subseteq$ BPP implies NP has hard-on-average languages, then we can conclude—without assuming any unproven complexity assumptions—that every language in NP has computational zero-knowledge proofs. The reason for this is that we can do the following case analysis: (i) if NP $\subseteq$ BPP, then every language L in NP is also in BPP, and hence L has a trivial zero-knowledge proof system where the verifier decides the language on its own; (ii) if NP $\not\subseteq$ BPP, then by the above hypothesis, NP has hard-on-average languages, and hence by [OW], one-way functions exist. As argued above, one-way functions imply that NP $\subseteq$ CZKP.

Unfortunately, our current knowledge of complexity theory does not exclude the possibility of both NP $\not\subseteq$ BPP and NP having *no* hard-on-average language happening simultaneously. Indeed, this possibility is often referred to as the *Heuristica world* of Impagliazzo [Imp].

Our research aims to overcome this gap in our understanding of zero knowledge even if we live in the Heuristica world. To achieve this objective, we follow Vadhan [Vad3] and characterize zero-knowledge protocols on a language-by-language basis instead of analyzing consequences of zero knowledge for an entire complexity class like NP. In other words, we study implications of a zero-knowledge protocol for a language L that are directly related to the language L itself. To get a more concrete taste of what this means, refer to the statements in Theorems 1.2.1 and 1.2.4.

Other unconditional works in cryptography. Our unconditional results should also be contrasted with some other works that do not rely on complexity assumptions, but work in less standard models of cryptography. For example, the works of Maurer [Mau], Cachin and Maurer [CM], and Ding and Rabin [DR] offer unconditionally secure cryptographic protocols against *memory-bounded* adversaries. In contrast, we only limit our adversaries in terms of its running time (to be probabilistic polynomial time), which is more standard in cryptography. In a different setting, Pass and Shelat [PS] constructed *noninteractive* zero-knowledge proof systems[17] in the *secret parameter setup model* for NP, where in this model the prover and the verifier is assumed to be able to obtain correlated private information.

[16] A language L is ***hard on average*** if it is hard to decide whether a random instance is in L or out of L. For a precise definition, refer to [Vad3, Def. 7.2].

[17] ***Noninteractive zero-knowledge proofs***, introduced by Blum et al. [BDMP], are zero-knowledge proofs that consists only of a single message from the prover to the verifier. Zero knowledge is achievable in this setting because both prover and verifier is assumed to have access to a *common random string*, uniformly chosen by a trusted third party.

1.4 Structure of this Dissertation

We provide an outline for the remaining chapters in this dissertation as follows.

Chapter 2 (A Tour of Zero Knowledge) is designed to provide the necessary definitions in order to understand the results in this dissertation.

Chapter 3 (Statistically-Hiding Commitments) has two main focuses. The first is to construct a statistically-hiding and computationally-hiding commitment scheme based on any one-way function. The second is to show that every language L in SZKP has an instance-dependent commitment scheme that is statistically hiding when $x \in L$ instances and statistically binding when $x \notin L$.

Chapter 4 (Unconditional Characterizations of Zero Knowledge) provides the main unconditional results of this dissertation, which were highlighted in Section 1.2. It does so by establishing an expanded version of Theorem 1.2.4 that gives characterizations of zero-knowledge protocols in terms of the Vadhan condition.

Chapter 5 (Future Research) explores a direction for future research.

Appendix A (Deferred Proofs) presents the proofs that are absent in the main text.

2

A TOUR OF ZERO KNOWLEDGE

In this chapter, we provide the necessary definitions in order to understand our results in Chapters 3 and 4. In general, unless noted otherwise, we use standard notations from complexity theory and cryptography; Sipser's textbook [Sip] provides an excellent introduction to complexity theory, and Vadhan's survey [Vad2] provides an illuminating introduction to zero-knowledge proofs and interactive proofs. For a comprehensive reference on the topics covered in this chapter, we refer the reader to a book by Goldreich [Gol2].

Chapter organization. In Section 2.1, we illustrate the concepts of zero knowledge through the elegant example of Graph Isomorphism. After that, we review basic complexity theory and cryptographic notions in Section 2.2.

In Section 2.3, we define the various flavors and variants of zero knowledge protocols. And then, in Section 2.4, we introduce one-way functions and commitments schemes, and their instance-dependent analogues. We conclude this chapter, in Section 2.5, by showing that instance-dependent commitments for a language in NP can be used to construct zero-knowledge protocols for that language.

2.1 An Example: Graph Isomorphism

Recall the algorithmic task of determining whether two graphs are isomorphic from Section 1.1. To recap, graphs G and H are ***isomorphic*** if the nodes of G may be permuted so that it is identical to H, and the language

$$\text{Graph Isomorphism} = \{(G,H) : G \text{ and } H \text{ are isomorphic}\} \ .$$

We have seen that given an permutation π of the nodes in G, one can easily check if

$\pi(G) = H$, in which case, G and H are isomorphic graphs.[1] Furthermore, if G and H are not isomorphic, then no permutation π will give rise to $\pi(G) = H$. We classify these types of languages into the class NP, which is formally defined next.

DEFINITION 2.1.1

A language L is in NP if there exists a deterministic polynomial-time verifier V and a constant $c > 0$ satisfying the following two conditions.

- *Completeness*: for every $x \in L$ (valid statement), there exists a string w of length less than $|x|^c$ that makes the verifier accept. In other words, $\exists w$ such that $|w| \leq |x|^c$ and $V(x, w) = 1$.
- *Soundness*: for every $x \notin L$ (invalid statement), every string w of length less than $|x|^c$ will make the verifier reject. In other words, $\forall w$ such that $|w| \leq |x|^c$ it is the case that $V(x, w) = 0$.

We can think of the string w as being supplied by another entity called a prover P. The reason why we bound the length of w to a polynomial in the length of x is so that V can check the supplied proof w efficiently.

Note that if we prove that G and H are isomorphic graphs by sending over a permutation π such that $\pi(G) = H$, then the verifier would learnt a reordering of the nodes of G that makes it equal to H. Hence, is there a way to prove that G and H are isomorphic graphs in a way that the verifier learns nothing? The answer is yes, and we present the following zero-knowledge protocol for Graph Isomorphism due to Goldreich, Micali, and Wigderson [GMW2].

PROTOCOL 2.1.2

Zero-knowledge protocol for Graph Isomorphism [GMW2].

Common input: pair of graphs (G, H). (We assume that the number of nodes in G and H are equal; otherwise, these graphs are not isomorphic.)

Auxiliary input for P: a permutation π such that $\pi(G) = H$.

$P \to V$: Select a uniformly random permutation π' over the nodes of H, and send graph $J = \pi'(H)$.

$V \to P$: Send a uniformly random bit c.

$P \to V$: If $c = 0$, send permutation $\pi'' = \pi' \circ \pi$. Otherwise, if $c = 1$, send permutation $\pi'' = \pi'$.

V: *Accept* if $c = 0$ and $\pi''(G) = J$, or if $c = 1$ and $\pi''(H) = J$. Otherwise, *reject*.

[1] $\pi(G)$ denotes the graph obtained by reordering the nodes of graph G according to permutation π.

In the protocol above, it is useful to think of the verifier's bit c as a challenge for the prover to show a permutation of G, when $c = 0$, or a permutation of H, when $c = 1$, that makes it match J.

We informally argue the completeness and soundness of Protocol 2.1.2 as follows:

Completeness. If G and H are isomorphic graphs, and the auxiliary input π to P is such that $\pi(G) = H$, then it is straightforward to check that the verifier V will always accept (with probability 1). Hence, Protocol 2.1.2 is said to have ***perfect completeness***.

Soundness. On the other hand, if G and H are not isomorphic, then any (cheating) prover P^* will not be able to convince V on both challenges $c = 0$ and $c = 1$ after graph J has been sent over in the first round. This is because in order to convince V on both $c = 0$ and $c = 1$, there must be permutations of both G and H that match J, implying that G and H isomorphic—and this is a contradiction. Therefore, in the case when G and H are not isomorphic, V will reject with probability at least 1/2, giving a soundness error of 1/2. To reduce the soundness error to 2^{-k}, we can repeat the protocol, sequentially or in parallel, k times.

Intuitively, verifier in Protocol 2.1.2 *learns nothing* because all it sees are a randomly permuted graph J from either G or H, and a permutation π'' that establishes it. It could have, without the help of a prover, toss a coin to decide whether to pick graph G or H, and then randomly permute the chosen graph on its own. To make our intuition precision, as stated in Section 1.1, this zero-knowledge guarantee of *learning nothing* is formalized in [GMR1] by requiring the existence of an efficient algorithm, called a simulator, whose output is indistinguishable from the verifier's view of the interaction with the prover, where the verifier's view consists of all messages exchanged between the prover and verifier and the verifier's random coins. As a first cut, we present a simulator for the ***honest verifier*** that follows the prescribed protocol.

ALGORITHM 2.1.3

Simulator S_1 for the honest verifier V in Protocol 2.1.2.

Input: pair of graphs (G, H). (As in Protocol 2.1.2, we assume that the number of nodes in G and H are equal; otherwise, these graphs are not isomorphic.)

1. Select a uniformly random permutation π' over the nodes of H, and a uniformly random bit b.
2. If $b = 0$, output $(\pi'(G), 0, \pi')$. Otherwise, if $b = 1$, output $(\pi'(H), 1, \pi')$.

It is important to realize that the simulator is only required to work in the case when G and H are isomorphic graphs. This is because the prover only provides proofs for valid statements. With this in mind, we informally argue the *honest-verifier zero knowledge* property of Protocol 2.1.2 as follows.

Honest-verifier zero knowledge. With probability 1/2, the verifier's view in Protocol 2.1.2 is $(\pi'(H), 0, \pi' \circ \pi) = (\pi' \circ \pi(G), 0, \pi' \circ \pi)$, noting that $\pi' \circ \pi$ is a uniformly random permutation over the nodes of G (since π', a uniformly random permutation, composed with π, a fixed permutation, yields a uniformly random permutation, i.e., $\pi' \circ \pi$). With the remaining probability of 1/2, the verifier's view in Protocol 2.1.2 is $(\pi'(H), 1, \pi')$, where π' is a uniformly random permutation over the nodes of H.

With the same probabilities, the simulator S_1 outputs identical distributions to the above; namely, with probability 1/2, S_1 outputs $(\pi'(G), 0, \pi')$, and with the remaining probability 1/2, S_1 outputs $(\pi'(H), 0, \pi')$, where in both cases, π' is a uniformly random permutation. Therefore, the output of S_1 is identically distributed to verifier's view in Protocol 2.1.2.

Parties in cryptography are usually *malicious*; in our case, a malicious verifier could bias the bit c in hope of learning something, or could base the value of c on the graph J sent in the first round. To guarantee that no verifiers, even malicious ones, can gain knowledge from the prover, we present the following simulator that handles arbitrary behaviors of malicious verifiers.

ALGORITHM 2.1.4

Simulator S_2 for any malicious verifier V^* in Protocol 2.1.2.

Input: pair of graphs (G, H). (As in Protocol 2.1.2, we assume that the number of nodes in G and H are equal; otherwise, these graphs are not isomorphic.)

1. Select a uniformly random string r so that we can fix the random tape of V^* to be r.
2. Select a uniformly random permutation π' over the nodes of H, and a uniformly random bit b.
3. If $b = 0$, set graph $J = \pi'(G)$. Otherwise, if $b = 1$, set graph $J = \pi'(H)$.
4. Let $c = V^*(J; r)$.
5. If $c = b$, then output (J, c, π', r) and halt. Otherwise, if $c \neq b$, then output `fail`.

We informally argue the *zero knowledge* property (i.e., secure against malicious verifiers) of Protocol 2.1.2 as follows.

Zero knowledge (secure against malicious verifiers). We first argue that conditioning on simulator S_2 not outputting `fail`, its output is identically distributed to V^*'s view of the interaction with the prover in Protocol 2.1.2. This is because, even if graph J is set as $\pi'(G)$ by S_2, J is still a uniformly random isomorphic copy of H since G and H are isomorphic graphs. Hence, the first-round message, namely J, produced by either S_2 or P is identically distributed. The second-round message, namely the bit c, produced by either S_2 or V^* is also identically distributed conditioned on any J, since both S_2 and V^* set the value of c to be equal $V^*(J;r)$. The third round message produced by by either S_2 or V^* is also identically distributed conditioned on any J and c, since $c = b$ (we are in the case where S_2 does not output `fail`).

Next, we argue that S_2 outputs `fail` with probability at most 1/2. This is because the bits c and b are independent, and b is a uniform bit. (The only way V^* can make c be dependent on b is by correlating c with J, but since J does not reveal any information about b, the values of c and b cannot be correlated.) So the probability that $c \neq b$, which corresponds to S_2 outputting `fail`, is exactly 1/2. This probability can be made exponentially small to 2^{-k} by running the simulator k times.

It is interesting to note that parallel repetition of Protocol 2.1.2 preserves the honest-verifier zero knowledge property, but does not necessarily preserve the zero knowledge (against malicious verifiers) property. This is because in the honest verifier case, a simulator can toss coins and determine the value of c in advance, whereas in the malicious verifier case, a simulator must guess the value of c. So if the protocol is repeated k times in parallel, the simulator for a malicious verifier might get all k values of the c's correctly only with probability 2^{-k}. This is an exponentially small value and cannot be increased to a constant by a polynomial number of repetitions. The zero knowledge (against malicious verifiers) property, however, is preserved under sequential repetition, at a cost of increasing the number of rounds. See Remark 2.3.6 for a further discussion on parallel versus sequential repetitions in the context of zero-knowledge protocols.

2.2 Preliminaries

2.2.1 Basic notations

Strings. Inputs of an algorithmic task are normally modeled as finite sequences of elements that come from a finite set. To be precise, we define an ***alphabet*** to be any finite set, a ***string*** to be any finite sequence of elements from that alphabet, and a ***language*** to be any set of strings. A computation process is normally thought of as manipulations of *bits*, so in this regard, the natural alphabet is the *binary alphabet* $\{0,1\}$. The binary alphabet can be used without loss of generality since it is easy to encode strings over a non-binary alphabet as a *binary string* without expanding the length of the string too much. At times, especially when considering security parameters, we use a *unary alphabet*, and denote a *unary string* of length k as 1^k.

Random variables. A ***random variable***, formally stated, is a *measurable function* mapping a *probability space* into a *measurable space*. Since we are dealing with only finite spaces in this dissertation, we can treat a random variable X as a function from a finite set S to the nonnegative reals $\mathbb{R}_{\geq 0}$ with the property that $\sum_{S \in S} X(S) = 1$. Hence, we can think of random variable X taking on value S with probability $X(S)$. A random variable X is ***uniform*** over a finite set S if it has equal weights on every element in S, i.e., $X(S) = 1/|S|$ for all $S \in S$. Because a random variable can be thought of as representing a *probability distribution*, we sometimes use these two terms interchangeably.

We write $x \leftarrow X$ to indicate that x is selected according to X. For a subset T of S, we write $x \leftarrow T$ to mean that x is selected according to the uniform random variable over T. We adopt the convention that when the same random variable occurs several times in an expression, they refer to a single sample. For example, $\Pr[f(X) = X]$ is defined to be the probability that when $x \leftarrow X$, we have $f(x) = x$. We write U_n to denote the random variable that is uniform over $\{0,1\}^n$.

Polynomially-small and negligible functions. A function $\mu : \mathbb{N} \rightarrow [0,1]$ is ***polynomially small*** if $\mu(n) = n^{-\Omega(1)}$. A function $\varepsilon \colon \mathbb{N} \rightarrow [0,1]$ is ***negligible*** if $\varepsilon(n) = n^{-\omega(1)}$. Let $\mathrm{neg}(n)$ denote an arbitrary negligible function (i.e., when we say that $f(n) < \mathrm{neg}(n)$ we mean that *there exists* a negligible function $\varepsilon(n)$ such that for every n, $f(n) < \varepsilon(n)$). Likewise, $\mathrm{poly}(n)$ denotes any function $f(n) = n^{O(1)}$.

Probabilistic algorithms. For a probabilistic algorithm A, we write $A(x; r)$ to denote the output of A on input x and coin tosses r. In this case, $A(x)$ is a random variable representing the output of A for uniformly selected coin tosses. The term ***PPT*** refers to probabilistic algorithms (i.e., Turing machines) that run in *strict* polynomial time. A ***nonuniform PPT*** algorithm is a pair $(A, \bar{z})$, where $\bar{z} = z_1, z_2, \ldots$ is an infinite sequence of strings where $|z_n| = \mathrm{poly}(n)$, and A is a PPT algorithm that receives pairs of inputs of the form $(x, z_{|x|})$. (The string z_n is the called the ***advice string*** for A for inputs of length n.) Nonuniform PPT algorithms are equivalent to (nonuniform) families of polynomial-sized Boolean circuits.

Statistical measures. The ***statistical difference***, also known as *variation distance*, between random variables X and Y taking values in $\mathcal{U}$ is defined to be

$$\Delta(X, Y) = \max_{S \subset \mathcal{U}} \left| \Pr[X \in S] - \Pr[Y \in S] \right| \quad .$$

Random variables X and Y are ε***-close*** if $\Delta(X, Y) \leq \varepsilon$. Conversely, random variables X and Y are ε***-far*** if $\Delta(X, Y) > \varepsilon$. For basic facts about this metric, see [SV, Sec. 2.3].

Entropy. The ***entropy*** of a random variable X is

$$\mathrm{H}(X) = \mathop{\mathrm{E}}_{x \leftarrow X}\big[-\log \Pr[X = x]\big] \enspace ,$$

where here and throughout all logarithms are of base 2. This notion of entropy corresponds to *Shannon entropy* or *information entropy* in the information theory literature. Intuitively, $\mathrm{H}(X)$ measures the amount of randomness in X *on average* (in bits). For a worst-case measure of randomness, the ***min-entropy*** of X is most often used, and is defined as

$$\mathrm{H}_\infty(X) = \min_x\big[-\log \Pr[X = x]\big] \enspace .$$

In general $\mathrm{H}_\infty(X) \leq \mathrm{H}(X)$, but when X is flat (that is, uniform on its support), then $\mathrm{H}(X) = \mathrm{H}_\infty(X) = \log|\mathrm{Supp}(X)|$.

2.2.2 Indistinguishability of probability ensembles

Recall that the zero-knowledge guarantee of learning nothing is formalized in [GMR1] by requiring the existence of an efficient simulator whose output is *indistinguishable* from the verifier's view of the interaction with the prover. Now, we formalize the notion of being indistinguishable.

DEFINITION 2.2.1

A ***probability ensemble***, or just an ***ensemble***, is a set of random variables $\{A_x\}_{x\in\{0,1\}^*}$, where A_x takes values in $\{0,1\}^{p(|x|)}$ for some polynomial p. We call such an ensemble ***samplable*** if there is a probabilistic polynomial-time algorithm M such that for every x, the output $M(x)$ is distributed according to A_x.

DEFINITION 2.2.2

Ensembles $\{A_x\}_{x\in\{0,1\}^*}$ and $\{B_x\}_{x\in\{0,1\}^*}$ are ***computationally indistinguishable on*** $I \subseteq \{0,1\}^*$ if for every nonuniform PPT D, there exists a negligible function ε such that for all $x \in I$,

$$|\Pr[D(x, A_x) = 1] - \Pr[D(x, B_x) = 1]| \leq \varepsilon(|x|) \enspace .$$

Similarly, ensembles $\{A_x\}_{x\in\{0,1\}^*}$ and $\{B_x\}_{x\in\{0,1\}^*}$ are ***statistically indistinguishable on*** $I \subseteq \{0,1\}^*$ if the above is required for all functions D, instead of only nonuniform PPT ones. Equivalently, $\{A_x\}_{x\in\{0,1\}^*}$ and $\{B_x\}_{x\in\{0,1\}^*}$ are statistically indistinguishable on I if and only if A_x and B_x are $\varepsilon(|x|)$-close for some negligible function ε and all $x \in I$. We use $\approx_c$ and $\approx_s$ to denote computational and statistical indistinguishability, respectively.

2.2.3 Promise problems

In Section 1.1, we mentioned that a *language* L can be thought of as defining the following algorithmic task: given a string x, determine if $x \in L$ or $x \notin L$.

We now consider a wider class of algorithmic tasks than those represented by languages. Specifically, we allow some input strings to be excluded. This new algorithmic task is formalized by a ***promise problem*** [ESY], which is specified by two disjoint sets of strings $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$, where Π_{Y} is the set of ***YES instances*** and Π_{N} is the set of ***NO instances***. Such a promise problem is associated with the following algorithmic task: given an input string that is *promised* to lie in $\Pi_{\mathrm{Y}} \cup \Pi_{\mathrm{N}}$, decide whether it is in Π_{Y} or in Π_{N}. Note that languages are a special case of promise problems (namely, a language L over alphabet Σ corresponds to the promise problem $(L, \Sigma^* \setminus L)$). Thus working with promise problems makes our results more general. Moreover, even to prove our results just for languages, it turns out to be extremely useful to work with promise problems along the way.

The ***complement*** of a promise problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$ is the promise problem $\overline{\Pi} = (\Pi_{\mathrm{N}}, \Pi_{\mathrm{Y}})$. The ***union*** of two promise problems Π and Γ is the promise problem $\Pi \cup \Gamma = (\Pi_{\mathrm{Y}} \cup \Gamma_{\mathrm{Y}}, \Pi_{\mathrm{N}} \cap \Gamma_{\mathrm{N}})$. The ***intersection*** of two promise problems Π and Γ is the promise problem $\Pi \cap \Gamma = (\Pi_{\mathrm{Y}} \cap \Gamma_{\mathrm{Y}}, \Pi_{\mathrm{N}} \cup \Gamma_{\mathrm{N}})$. A ***class of promise problems*** is a set of promise problems. If C is a class of promise problems, then the ***complement class*** is defined as co-C $= \{\overline{\Pi} : \Pi \in \mathrm{C}\}$.

Most complexity classes, typically defined as classes of languages, extend to promise problems in a natural way, by translating conditions on inputs in the language to be conditions on YES instances, and conditions on inputs not in the language to be conditions on NO instances. For example, a promise problem Π is in BPP if there is a probabilistic polynomial-time algorithm A such that $x \in \Pi_{\mathrm{Y}} \Rightarrow \Pr[A(x) = 1] \geq 2/3$ and $x \in \Pi_{\mathrm{N}} \Rightarrow \Pr[A(x) = 0] \leq 1/3$.

A promise problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$ ***polynomial-time reduces***, or just ***reduces***, to another promise problem $\Gamma = (\Gamma_{\mathrm{Y}}, \Gamma_{\mathrm{N}})$ if there is a polynomial-time computable function f satisfying $f(x) \in \Gamma_{\mathrm{Y}}$ for every $x \in \Pi_{\mathrm{Y}}$ and $f(x) \in \Gamma_{\mathrm{N}}$ for every $x \in \Pi_{\mathrm{N}}$. That is, we work with polynomial-time mapping reductions (i.e., Karp reductions), unless otherwise specified. If Π reduces to Γ and vice versa, then Π and Γ are ***polynomial-time equivalent***. If C is a class of promise problems, then Π is ***complete for*** C (or C-***complete***) if $\Pi \in \mathrm{C}$ and every promise problem in C reduces to Π.

REMARK 2.2.3

Some theoretical computer science papers do make a distinction between BPP, the class of languages that are decidable in probabilistic polynomial time, and prBPP, the promise variant of BPP. This is mainly because their results do not apply to both classes: for example, prBPP = prP implies BPP = P, but the converse is not known.

We, however, do not make this distinction because all our results apply to promise classes as well as classes of languages. Hence, throughout this dissertation, all complexity classes are classes of promise problems. In addition, we often use the term ***problem*** Π to mean the promise problem Π.

2.3 Zero-Knowledge Protocols

Our goal in this section is to build up the necessary machinery to understand the various flavors and variants of zero-knowledge protocols. Throughout our discussions, it will be helpful to keep the example of Graph Isomorphism from Section 2.1 in mind.

2.3.1 Interactive proofs and interactive arguments

In Protocol 2.1.2 for Graph Isomorphism the prover exchange messages with the verifier in order to prove that the two graphs are isomorphic. We model this interaction between the prover and verifier in a setting of an *interactive protocol.*

DEFINITION 2.3.1

An ***interactive protocol*** (A, B) consists of two algorithms A and B that compute the *next-message function* of the (honest) parties in the protocol. Specifically, for even values of k, $A(x, a, m_1, \ldots, m_k; r_A)$ denotes the next message m_{k+1} sent by party A when the common input is x, A's auxiliary input is a, A's coin tosses are r_A, and the messages exchanged so far are $m_1, \ldots, m_k$. Analogously for party B, $B(x, b, m_1, \ldots, m_k; r_B)$, for odd values of k, denotes the next message m_{k+1} sent by party B when the common input is x, B's auxiliary input is b, B's coin tosses are r_B, and the messages exchanged so far are $m_1, \ldots, m_k$. There are two special messages, `accept` and `reject`, which immediately halt the interaction.

Party A (resp. B) is ***polynomial-time computable*** if its next-message function can be computed in polynomial time (in $|x| + |a| + |m_1| + \cdots + |m_k|$).

The number of ***rounds*** in an execution of the protocol is the *total* number of messages exchanged between A and B, not including the final `accept/reject` message.

Interactive protocol (A, B) is ***public coin*** if all of the messages sent by B are simply the output of its coin-tosses (independent of the history), except for the final `accept/reject` message which is computed as a deterministic function of the transcript. (Such protocols are also sometimes known as *Arthur-Merlin games* [BM].)

For an interactive protocol (A, B), we write $(A(a), B(b))(x)$ to denote the random process obtained by having A and B interact on common input x, (private) auxiliary inputs a and b to A and B, respectively (if any), and independent random coin tosses for A and B. We call (A, B) ***polynomially bounded*** if there is a polynomial p such that for all x, a, b, the total length of all messages exchanged in $(A(a), B(b))(x)$ is at most $p(|x|)$ with probability 1. Moreover, for any algorithm B^*, A will immediately halt and reject in $(A(a), B^*(b))(x)$

if the total length of the messages ever exceeds $p(|x|)$, and similarly for B interacting with any A^*.

View of the interaction. To denote A's ***view of the interaction***, we use random variable $\text{view}_A(A(a), B(b))(x)$ to mean $(x, a, m_1, m_2, \ldots, m_t, r_A)$, where the m_i's are all the messages exchanged and r_A is A's coin tosses. Similarly, to denote B's view of the interaction, we use random variable $\text{view}_B(A(a), B(b))(x)$ to mean $(x, b, m_1, m_2, \ldots, m_t, r_B)$, where the m_i's are all the messages exchanged and r_B is B's coin tosses. When dealing with interactive protocols (P, V) involving a prover P and a verifier V, it is common to write $\langle P, V\rangle(x)$ to denote V's view of the interaction, that is $\langle P, V\rangle(x) = \text{view}_V(P, V)(x)$.

Transcripts and outputs. A ***transcript*** of interactive protocol (A, B), denoted by the random variable $\text{transcript}(A(a), B(b))(x)$, is the messages exchanged in the protocol including the common input x, i.e., $(x, m_1, m_2, \ldots, m_t)$. Let random variables $\text{output}_A(A(a), B(b))$ and $\text{output}_B(A(a), B(b))(x)$ denote A's and B's ***private output*** after the interaction, respectively. At times, we will refer to protocols with a joint output; such an output is specified by a deterministic, polynomial-time computable function of the messages exchanged.

Statistical versus computational soundness. Recall that in Section 1.1, we discussed two flavors of soundness: ***statistical soundness***, giving rise to ***interactive proof systems***, where even a computationally unbounded prover cannot convince the verifier of a false statement, and ***computational soundness***, giving rise to ***interactive argument systems*** [BCC], where we only require that an efficient, probabilistic polynomial-time prover cannot convince the verifier of a false statement.

First, we give a formal definition of statistical soundness, which gives rise to interactive proof systems.

DEFINITION 2.3.2

An interactive protocol (P, V) is an ***interactive proof system*** for a promise problem Π if exist functions $c, s : \mathbb{N} \to [0, 1]$ such that $1 - c(n) > s(n) + 1/\operatorname{poly}(n)$ and the following conditions hold.

- *Efficiency*: (P, V) is polynomially bounded, and V is polynomial-time computable.
- *Completeness*: if $x \in \Pi_{\text{Y}}$, then V accepts in $(P, V)(x)$ with probability at least $1 - c(|x|)$,
- *Statistical soundness*: if $x \in \Pi_{\text{N}}$, then for every P^*, V accepts in $(P^*, V)(x)$ with probability at most $s(|x|)$.

We call $c(\cdot)$ the ***completeness error*** and $s(\cdot)$ the ***soundness error***. Interactive protocol (P, V) has ***negligible error*** if both c and s are negligible; it has ***perfect completeness*** if $c = 0$.

Let IP be the class of promise problems possessing interactive proof systems. An equivalent definition of IP is the class of problems possessing public-coin interactive proof systems with perfect completeness and negligible soundness error [GS, FGM$^+$].

Let MA be the class of promise problems possessing *single-round* interactive proof systems: in such protocols, the prover P sends a *single* message to V, and V decides whether to accept or reject based on the prover's message and its own random coins. Thus, we can think of MA as a generalization of NP where the verification of witnesses is probabilistic.

Remark. Protocol 2.1.2 for Graph Isomorphism is an interactive proof system with perfect completeness and soundness error 1/2.

Next, we give a formal definition of computational soundness, which gives rise to interactive argument systems.

DEFINITION 2.3.3

An interactive protocol (P, V) is an ***interactive argument system*** for Π if the soundness condition in Definition 2.3.2 holds against all nonuniform PPT P^*, instead of every, even computationally unbounded, P^*. Specifically, we require interactive argument systems to satisfy both the efficiency and the completeness conditions in Definition 2.3.2, and to satisfy a weaker, computational soundness condition below.

- *Computational soundness*: if $x \in \Pi_{\mathrm{N}}$, then for every *nonuniform PPT* P^*, V accepts in $(P^*, V)(x)$ with probability at most $s(|x|)$.

Let IA be the class of promise problems possessing interactive argument systems.

Unlike interactive proofs, the complexity-theoretic aspects of IA are not well-studied. In particular, we do not know if general interactive arguments can be made to have public coin or to have perfect completeness. The completeness and soundness error, however, can be made negligibly small by sequential repetition.

2.3.2 Efficient provers

Although we define interactive arguments without restricting the computational resource the honest prover P, it is natural to do since the cheating provers P^* are restricted to be PPT. Hence, interactive arguments are most interesting when considering problems in NP, because for these problems, we can restrict the honest prover to be PPT given a witness of membership. To formalize this idea, we define witness relations for problems in NP.

Recall that NP, informally stated, is the class of problems that can be verified in polynomial time given a valid witness. To formally define the relationship between an instance and its corresponding valid witnesses, we consider a relation W and say that W is *polynomial* *time* if deciding whether an element is in W can be done in polynomial time in the length of the first component of the input (this is typically the length of the problem instance). With this in mind, a problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$ is in NP if there exist a polynomial-time binary relation $W \subseteq \{0,1\}^* \times \{0,1\}^*$ such that the following two conditions hold:

- for every $x \in \Pi_{\mathrm{Y}}$, there exists a w with $(x, w) \in W$;

- for every $x \in \Pi_{\mathrm{Y}}$, and for every w, it is the case that $(x, w) \notin W$.

Any polynomial-time binary relation W that satisfies the above two conditions is said to be an ***NP-relation*** for the problem Π.

For MA, the probabilistic analog of NP, we generalize the relation W to allow for randomness; specifically, we expand the domain of $W \subseteq \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$. To relate it with the NP case above, we abuse notation and write $(x, w) \in W$ if $\Pr_r[(x, w, r) \in W] \geq 2/3$, and write $(x, w) \notin W$ if $\Pr_r[(x, w, r) \in W] \leq 1/3$. Then, a problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}) \in \mathrm{MA}$ if there exist a polynomial-time relation $W \subseteq \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$ such that the following two conditions hold:

- for every $x \in \Pi_{\mathrm{Y}}$, there exists a w with $(x, w) \in W$, namely $\Pr_r[(x, w, r) \in W] \geq 2/3$;

- for every $x \in \Pi_{\mathrm{Y}}$, and for every w, it is the case that $(x, w) \notin W$, namely $\Pr_r[(x, w, r) \in W] \leq 1/3$.

Any polynomial-time relation W that satisfies the above two conditions is said to be an ***MA-relation*** for the problem Π.

In an interactive protocol (P, V) for problem $\Pi \in \mathrm{NP}$ [resp., $\Pi \in \mathrm{MA}$], prover P is an ***efficient prover*** if its strategy on problem instance x is computable in polynomial time given w as auxiliary input, where $(x, w) \in W$ and W is an NP-relation [resp., MA-relation] for Π. When this is the case, protocol (P, V) is said to have an efficient prover. By allowing the relation W to be more general, we could have defined the notion of efficient prover to languages outside MA. Nevertheless, we did not do so since efficient provers can be defined, without loss of generality, only for problems in MA [BD, BLV].

Remark. Protocol 2.1.2 for Graph Isomorphism has an efficient prover.

2.3.3 Flavors and variants of zero knowledge

Recall that in Section 1.1, we discussed the two ***flavors*** of zero knowledge: ***statistical zero knowledge***, where the zero-knowledge condition holds regardless of the computational resources the verifier invests into trying to learn something from the interaction (except with negligible probability), and ***computational zero knowledge***, we only require that efficient, probabilistic polynomial-time verifiers learn nothing from the interaction.[2] For each of these flavors, there are several ***variants*** of zero knowledge, referring to how rich a class of verifier strategies are considered.

Honest-verifier zero knowledge

We start with the weakest variant of zero knowledge, where only the ***honest verifier*** that follows the prescribed strategy is guaranteed to learning nothing.[3]

DEFINITION 2.3.4

An interactive proof system (P, V) for a promise problem Π is ***statistical [resp., computational] honest-verifier zero knowledge*** if there exists a probabilistic polynomial-time *simulator* S such that the ensembles $\{\langle P, V\rangle(x)\}$ and $\{S(x)\}$ are statistically [resp., computationally] indistinguishable on the set Π_Y.

HV-SZKP and HV-CZKP denote the classes of promise problems have honest-verifier statistical and computational zero-knowledge proofs, respectively. Analogously, HV-SZKA and HV-CZKA denote the classes of promise problems have honest-verifier statistical and computational zero-knowledge arguments, respectively.

Malicious-verifier, auxiliary-input zero knowledge

While honest-verifier zero knowledge is already a nontrivial and interesting notion, cryptographic applications usually require that the zero-knowledge condition holds even if a ***malicious verifier*** deviates arbitrarily from the specified protocol. This is captured by the following definition.

DEFINITION 2.3.5

An interactive proof system (P, V) for a promise problem Π is ***statistical [resp., computational] auxiliary-input zero knowledge***[4] if for every PPT V^* and polynomial p, there exists a PPT S such that the ensembles $\{\langle P, V^*(z)\rangle(x)\}$ and $\{S(x, z)\}$ are statistically

[2] There is a third flavor of zero knowledge which is the strongest of all: ***perfect zero knowledge***, where the verifier cannot *learn anything* even with negligible probability. In this dissertation, we do *not* study the distinction between perfect zero knowledge and statistical zero knowledge. See Section 2.3.4 for a more detailed discussion.

[3] This is an instantiation of what is called an *honest-but-curious adversary* or *passive adversary* in the literature on cryptographic protocols.

[4] Our formulation of auxiliary-input zero knowledge is slightly different than, but equivalent to, the definition in the textbook [Gol2]. We allow V^* to run in polynomial time in the lengths of both its input x and its auxiliary input z, but put a polynomial bound on the length of the auxiliary input. In [Gol2, Sec 4.3.3], V^* is restricted to run in time that is polynomial in just the length of the input x, and no bound is imposed on the length of the auxiliary input z (so V^* may only be able to read a prefix of z). The purpose of allowing the auxiliary input to be longer than the running time of z is to provide additional nonuniformity to the distinguisher (beyond that which the verifier has); we do this directly by allowing the distinguisher to be nonuniform in Definition 2.2.2.

[resp., computationally] indistinguishable on the set $\{(x, z) : x \in \Pi_Y, |z| = p(|x|)\}$.

SZKP and CZKP are the classes of promise problems possessing statistical and computational auxiliary-input zero-knowledge proofs, respectively. Analogously, SZKA and CZKA are the classes of promise problems possessing statistical and computational auxiliary-input zero-knowledge arguments, respectively. To avoid cumbersome terminologies, we often drop the prefix "auxiliary input" and just use ***zero knowledge*** to mean auxiliary-input zero knowledge.

REMARK 2.3.6

The auxiliary input z in the above definition allows one to model a priori information that the verifier may possess before the interaction begins, such as from earlier steps in a larger protocol in which the zero-knowledge proof is being used or from prior executions of the same zero-knowledge proof. As a result, auxiliary-input zero knowledge is closed under sequential composition. That is, if an auxiliary-input zero-knowledge proof is repeated polynomially many times sequentially, then it remains auxiliary-input zero knowledge [GO]. Plain zero knowledge (i.e., without auxiliary inputs) is not closed under sequential composition [GK2], and thus auxiliary-input zero knowledge is the definition typically used in the literature. We caution that parallel repetition does not, in general, preserve the (auxiliary input) zero knowledge property [GK2, FS].

Black-box zero knowledge

Typically, a protocol is proven to be zero knowledge by actually exhibiting a single, universal simulator that simulates an arbitrary verifier strategy V^* by using V^* as a subroutine. That is, the simulator does not depend on or use the code of V^* (or its auxiliary input), and instead only requires *black-box access* to V^*. This type of simulation is formalized as follows.

DEFINITION 2.3.7

An interactive proof system (P, V) for a promise problem Π is ***statistical [resp., computational] black-box zero knowledge*** if there exists an oracle PPT S such that for every nonuniform PPT V^*, the ensembles $\{\langle P, V^* \rangle(x)\}$ and $\{S^{V^*(x,\cdot;\cdot)}(x)\}$ are statistically [resp., computationally] indistinguishable on the set Π_Y.

Remark. Protocol 2.1.2 for Graph Isomorphism is statistical black-box zero knowledge since its simulator S_2 in Algorithm 2.1.4 only requires black-box access to the malicious verifier V^*, and outputs transcripts that are statistically indistinguishable from V^*'s view of the interaction with the prover.

Even though the above Definition 2.3.7 does not explicitly refer to an auxiliary input, the definition encompasses auxiliary-input zero knowledge because we allow V^* to be nonuniform (and thus the auxiliary input can be hardwired in V^* as advice). The work of Barak [Bar] demonstrated that non-black-box zero-knowledge arguments can achieve properties (such as simultaneously being public coin, having a constant number of rounds, and having negligible error) that were known to be impossible for black-box zero knowledge [GK2]. Nevertheless, our results will show that, when ignoring round efficiency considerations, black-box zero knowledge is as rich as auxiliary-input zero knowledge; for example, in Chapter 4, we show that every problem in CZKA has a black-box computational zero-knowledge argument system.

2.3.4 Remarks on the definitions

Our definitions mostly follow the now-standard definitions of zero-knowledge proof and argument systems as presented in [Gol2], but we highlight the following points.

1. *Prover complexity*: interactive proofs and interactive arguments, and their zero-knowledge analogues, allow the honest prover to be computationally unbounded, unless we specify *efficient prover*. It was shown by Nguyen and Vadhan [NV] that for problems in NP (actually, also MA), any zero-knowledge *proof* system with an unbounded prover can be transformed into one with an efficient prover; we will show the same for *argument* systems.

2. *Promise problems*: as has been done numerous times before (e.g., [GK3, SV]), we extend all of the definitions to promise problems $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$ in the natural way, i.e., conditions previously required for inputs in the language (e.g., completeness and zero knowledge) are now required for all YES instances, and conditions previously required for inputs not in the language (e.g., soundness) are now required for all NO instances. Similarly, all of our complexity classes (e.g., CZKA, SZKP and BPP) are classes of promise problems. These extensions to promise problems are essential for formalizing our arguments, but all the final characterizations and results we derive about CZKA automatically hold for the corresponding class of languages, simply because languages are a special case of promise problems.

3. *Nonuniform formulation*: as has become standard, we have adopted a nonuniform formulation of zero knowledge, where the computational indistinguishability has to hold even with respect to nonuniform distinguishers and is universally quantified over all YES instances. Uniform treatments of zero knowledge are possible (see [Gol1] and [BLV, Apdx. A]), but the definitions are much more cumbersome. We do not know whether analogues of our results hold for the uniform formulation of zero knowledge, and leave that as a problem for future work.

4. *Strict polynomial-time simulators*: we restrict our attention to zero knowledge with respect to simulators that run in *strict* polynomial time; in Definitions 2.3.4, 2.3.5,

and 2.3.7, the simulator is PPT, which means that in runs in strict polynomial time. The original Goldwasser, Micali, and Rackoff definition, however, allows the running time of the simulator to be *expected polynomial time*; we say that random variable X_n is ***expected polynomial*** if $\mathrm{E}[X_n] = \mathrm{poly}(n)$. We prefer the strict polynomial time definition of zero knowledge, because as pointed out by Levin [Lev], the definition of expected polynomial time is dependent on the model of computation and does not compose well.[5]

5. *Perfect versus statistical zero knowledge*: A third flavor of zero knowledge proposed by [GMR1], which is the strongest of all, is ***perfect zero knowledge***. This is where all verifiers cannot *learn anything* even with negligible probability; that is, the output of the simulator is required to be *identically* distributed to the view of the verifier. Refer to [Gol2, Def. 4.3.1] for a formal definition of perfect zero knowledge.

   In this dissertation, we do *not* study the distinction between perfect zero knowledge and statistical zero knowledge. We note that perfect zero knowledge is a fragile property to manipulate and could depend on the model of computation. For instance, Protocol 2.1.2, the zero-knowledge protocol for Graph Isomorphism, is perfect zero knowledge with perfect completeness only if we assume that the prover is able to uniformly select a random permutation. In a model of probabilistic computation with random *bits*, selecting a uniform permutation can be achieved only by allowing for a negligible probability of failure. (This is because the total possible permutations over n elements, which is $n!$, is not a power of 2.) Nevertheless, by considering statistical zero knowledge, we are able to absorb this negligible probability of failure into the statistical difference between the output of the simulator and the view of the verifier, and hence achieve a statistical zero knowledge proof system for Graph Isomorphism with perfect completeness.

   Furthermore, if we insist on perfect zero knowledge, the malicious-verifier simulator for Protocol 2.1.2, such as the one in Algorithm 2.1.4, must be allowed to fail with small probability; no efficient simulator is known to be able to output a distribution that is identical to the verifier's view of the interaction in Protocol 2.1.2 without failing [Gol2, Sect. 4.3.1.1]. Again by considering statistical zero knowledge, we are able to absorb this small probability of failure into the statistical difference between the output of the simulator and the view of the verifier, and obtain a simulator that does not need to fail.

[5] For instance, if X_n is expected polynomial time, it is not necessarily the case that X_n^2 is expected polynomial time. A counterexample is X_n equaling 2^n with probability 2^{-n} and 1 otherwise.

2.4 Cryptographic Primitives and Instance-Dependent Analogues

In this section, we give formal definitions of *one-way functions* and *commitment schemes*, and their instance-dependent analogues.

2.4.1 One-way functions

The most basic primitive of modern cryptography is a one-way function, which are functions that are *easy to compute* but *hard to invert*.

DEFINITION 2.4.1

Let $s\colon \mathbb{N} \to \mathbb{N}$ be any function. A function $f\colon \{0,1\}^* \to \{0,1\}^*$ is a $s(n)$***-secure one-way function***, or equivalently ***has security*** $s(n)$, if f is computable in polynomial time and for every nonuniform PPT A,

$$\Pr_{y \leftarrow \{0,1\}^n}[A(1^n, f(y)) \in f^{-1}(f(y))] < 1/s(n),$$

for all sufficiently large n. Function f is a ***one-way function*** if f is $s(n)$-secure for every polynomial s.

Without loss of generality, we can consider only one-way functions that are length-preserving, that is for all $y \in \{0,1\}^*$, $|f(y)| = |y|$. This is because general one-way functions can be converted into ones that are length-preserving (cf., [Gol2, p. 39]).

One-way function f is a ***regular one-way function with preimage size*** $g(n)$ if there exists a function $g\colon \mathbb{N} \to \mathbb{N}$ such that $\forall z \in \mathrm{Supp}(f(U_n))$, $|\{y \in \{0,1\}^n : f(y) = z\}| = g(n)$.

2.4.2 Commitment schemes

Another basic primitive of modern cryptography is a ***(bit) commitment scheme***, which is a two-stage protocol between a sender and a receiver. In the first stage, called the ***commit stage***, the sender *commits* to a private bit b. In the second stage, called the ***reveal stage***, the sender reveals b and *proves* that it was the bit to which she committed in the first stage. We require two properties of commitment schemes. The ***hiding*** property says that the receiver learns nothing about b in the commit stage. The ***binding*** property says that after the commit stage, the sender is bound to a particular value of b; that is, she cannot successfully open the commitment to two different bits in the reveal stage.

DEFINITION 2.4.2
An ***commitment scheme*** is an interactive protocol $\mathsf{Com} = (S, R)$ with the following properties:

1. Scheme Com proceeds in two stages: a ***commit stage*** and a ***reveal stage***. In both stages, the ***sender*** S and the ***receiver*** R receive a security parameter 1^n as common input.
2. At the beginning of the commit stage, sender S receives a private input $b \in \{0, 1\}$, which denotes the bit that S is supposed to commit to. At the end of the commit stage, both sender S and receiver R output a ***commitment*** string c.
3. In the reveal stage, sender S sends a pair (b, d), where d is the ***decommitment*** string for bit b. Receiver R accepts or rejects based on b, d, and c.
4. The sender S and receiver R algorithms are computable in polynomial time in the security parameter n.
5. R will always accept (with probability 1) if both sender S and receiver R follow their prescribed strategy.

A commitment scheme is ***public coin*** if all messages sent by the receiver are independent random coins.

Next, we define the hiding and binding properties of commitment schemes.

DEFINITION 2.4.3
Commitment scheme $\mathsf{Com} = (S, R)$ is ***statistically [resp., computationally] hiding*** if for every [resp., nonuniform PPT] R^*, the ensembles $\{\mathrm{view}_{R^*}(S(0), R^*)(1^n)\}_{n \in \mathbb{N}}$ and $\{\mathrm{view}_{R^*}(S(1), R^*)(1^n)\}_{n \in \mathbb{N}}$ are statistically [resp., computationally] indistinguishable, where $\mathrm{view}_{R^*}(S(b), R^*)$ denotes the view of R^* in the commit stage interacting with $S(b)$.

DEFINITION 2.4.4
Commitment scheme $\mathsf{Com} = (S, R)$ is ***statistically [resp., computationally] binding*** if for every [resp., nonuniform PPT] S^*, there exists a negligible function ε such that the malicious sender S^* succeeds in the following game with probability at most $\varepsilon(n)$:

> On security parameter 1^n, S^* interacts with R in the commit stage obtaining commitment c. Then S^* outputs pairs $(0, d_0)$ and $(1, d_1)$, and *succeeds* if in the reveal stage, $R(0, d_0, c) = R(1, d_1, c) = \texttt{accept}$.

In the two next subsections—Sections 2.4.3 and 2.4.4—we consider analogues of one-way functions and commitments scheme that can depend on the problem instance.

2.4.3 Instance-dependent one-way functions

It will be useful for us to work with cryptographic primitives that may depend on an instance x of a problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$, and where the security condition will hold only if x is in some particular set $I \subseteq \{0,1\}^*$. Indeed, recall that the Vadhan condition (Definition 1.2.3) refers to such a variant of of one-way functions, as captured by Definition 2.4.6 below.

To define instance-dependent one-way functions, we will need to define what it means for a function to be *instance dependent*.

DEFINITION 2.4.5
An ***instance-dependent function*** is a family $\mathcal{F} = \{f_x \colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}\}_{x\in\{0,1\}^*}$, where $n(\cdot)$ and $m(\cdot)$ are polynomials. We call $\mathcal{F}$ ***polynomial-time computable*** if there is a deterministic polynomial-time algorithm F such that for every $x \in \{0,1\}^*$ and $y \in \{0,1\}^{n(|x|)}$, we have $F(x,y) = f_x(y)$.

To simplify notation, we often write $f_x \colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}$ to mean the instance-dependent function $\{f_x \colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}\}_{x\in\{0,1\}^*}$.

DEFINITION 2.4.6
For any set $I \subseteq \{0,1\}^*$, a polynomial-time computable instance-dependent function $f_x \colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}$ is an ***instance-dependent one-way function on*** I if for every nonuniform PPT adversary A, there exists a negligible function ε such that for every $x \in I$,

$$\Pr_{y\leftarrow\{0,1\}^{n(|x|)}}\left[A(x, f_x(y)) \in f_x^{-1}(f_x(y))\right] \le \varepsilon(|x|) \ .$$

Next we consider an instance-dependent variant of *distributionally one-way functions*, which are functions that are hard for PPT adversaries to invert in a distributional manner—that is, given y it is hard for PPT adversaries to output a random preimage $f^{-1}(y)$. The standard definition of distributionally one-way function is given by Impagliazzo and Luby [IL]; here we give the instance-dependent analogue.

DEFINITION 2.4.7
For any set $I \subseteq \{0,1\}^*$, a polynomial-time computable instance-dependent function $f_x \colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}$ is an ***instance-dependent distributionally one-way function on*** I if there exists a polynomial $p(\cdot)$ such that for every nonuniform PPT adversary A, the random variables $(U_{n(|x|)}, f_x(U_{n(|x|)}))$ and $(A(f_x(U_{n(|x|)})), f_x(U_{n(|x|)}))$ are $1/p(|x|)$-far for all sufficiently long $x \in I$.

Asking to invert in a distributional manner is a stronger requirement that just finding a preimage, therefore distributionally one-way functions might seem weaker than one-way functions. Impagliazzo and Luby [IL], however, proved that they are in fact equivalent. Like almost all reductions between cryptographic primitives, this result immediately extends to the instance-dependent analogue (using the same proof).

PROPOSITION 2.4.8

(Based on [IL, Lem. 1].) For every set $I \subseteq \{0,1\}^*$, there exists an instance-dependent one-way function on I if and only if there exists an instance-dependent distributionally one-way function on I.

2.4.4 Instance-dependent commitment schemes

Recall the standard definition of the commitment schemes from Section 2.4.2. Instance dependent analogues of commitments schemes are commitments schemes that are tailored specifically to a specific problem Π. More precisely, ***instance-dependent commitment schemes*** [BMO, IOS, MV] receive an instance x of the problem Π as auxiliary input, and are required to be hiding when $x \in \Pi_Y$ and be binding when $x \in \Pi_N$.[6] Thus, they are a relaxation of standard commitment schemes, since we do not require that the hiding and binding properties hold at the same time. Nevertheless, as observed in [IOS], this relaxation is still useful in constructing zero-knowledge protocols. The reason is that zero-knowledge protocols based on commitments (for example, the protocol of [GMW2]) typically use only the hiding property in proving zero knowledge (which is required only when x is a YES instance) and use only the binding property in proving soundness (which is required only when x is a NO instance).

We give a definition of instance-dependent commitment schemes that extends the standard (that is, non-instance dependent) definition of commitment schemes in a natural way. Note that in our definition below, the reveal stage is *noninteractive* (that is, consisting of a single message from the sender to the receiver). This because in the reveal stage, without loss of generality, we can have the sender provide the receiver the random coin tosses it used in the commit stage, and the receiver verifies consistency.

DEFINITION 2.4.9

An ***instance-dependent commitment scheme*** is a family $\{\mathsf{Com}_x\}_{x\in\{0,1\}^*}$ with the following properties:

1. Scheme Com_x proceeds in two stages: a *commit stage* and a *reveal stage*. In both stages, the *sender* and *receiver* receive instance x as common input, and hence we denote the sender and receiver as S_x and R_x, respectively, and write $\mathsf{Com}_x = (S_x, R_x)$.

[6] There were various terms used to describe *instance-dependent commitment schemes*. Itoh, Ohta, and Shizuya [IOS] called these *language-dependent cryptographic primitives*, Micciancio and Vadhan [MV] called these *problem-dependent commitment schemes*, and the present usage traces to Vadhan [Vad3].

2. At the beginning of the commit stage, sender S_x receives a private input $b \in \{0,1\}$, which denotes the bit that S is supposed to commit to. At the end of the commit stage, both sender S_x and receiver R_x output a *commitment* c.

3. In the reveal stage, sender S_x sends a pair (b, d), where d is the *decommitment* string for bit b. Receiver R_x accepts or rejects based on x, b, d, and c.

4. The sender S_x and receiver R_x algorithms are computable in polynomial time (in $|x|$), given x as auxiliary input.

5. For every $x \in \{0,1\}^*$, R_x will always accept (with probability 1) if both sender S_x and receiver R_x follow their prescribed strategy.

Instance-dependent commitment scheme $\{\mathsf{Com}_x = (S_x, R_x)\}_{x\in\{0,1\}^*}$ is *public coin* if for every $x \in \{0,1\}^*$, all messages sent by R_x are independent random coins.

To simplify notation, we write Com_x or (S_x, R_x) to denote instance-dependent commitment scheme $\{\mathsf{Com}_x = (S_x, R_x)\}_{x\in\{0,1\}^*}$.

The hiding and binding properties of standard commitments—stated in Definitions 2.4.3 and 2.4.4—extend in a natural way to their instance-dependent analogues.

DEFINITION 2.4.10

Instance-dependent commitment scheme $\mathsf{Com}_x = (S_x, R_x)$ is ***statistically [resp., computationally] hiding on*** $I \subseteq \{0,1\}^*$ if for every [resp., nonuniform PPT] R^*, the ensembles $\{\mathrm{view}_{R^*}(S_x(0), R^*)\}_{x\in I}$ and $\{\mathrm{view}_{R^*}(S_x(1), R^*)\}_{x\in I}$ are statistically [resp., computationally] indistinguishable, where random variable $\mathrm{view}_{R^*}(S_x(b), R^*)$ denotes the view of R^* in the commit stage interacting with $S_x(b)$. For a problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$, an instance-dependent commitment scheme Com_x for Π is ***statistically [resp., computationally] hiding on the YES instances*** if Com_x is statistically [resp., computationally] hiding on Π_{Y}.

DEFINITION 2.4.11

Instance-dependent commitment scheme $\mathsf{Com}_x = (S_x, R_x)$ is ***statistically [resp., computationally] binding on*** $I \subseteq \{0,1\}^*$ if for every [resp., nonuniform PPT] S^*, there exists a negligible function ε such that for all $x \in I$, the malicious sender S^* succeeds in the following game with probability at most $\varepsilon(|x|)$.

> S^* interacts with R_x in the commit stage obtaining commitment c. Then S^* outputs pairs $(0, d_0)$ and $(1, d_1)$, and *succeeds* if in the reveal stage, $R_x(0, d_0, c) = R_x(1, d_1, c) = \texttt{accept}$.

For a problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$, an instance-dependent commitment scheme Com_x for Π is ***statistically [resp., computationally] binding on the NO instances*** if Com_x is statistically [resp., computationally] binding on Π_{N}.

For concreteness, we present an instance-dependent commitment scheme for the Graph Isomorphism problem (cf., [BMO, IOS]).

ALGORITHM 2.4.12

Instance-dependent commitment scheme for Graph Isomorphism.

Problem instance: A pair of graphs (G_0, G_1) (this corresponds to pair (G, H) in the definition of Graph Isomorphism presented in Section 2.1.)

Commit stage: to commit to a bit b, sender $S_{(G_0,G_1)}$ selects a uniformly random permutation π over the nodes of G_b, and sends as commitment the graph $J = \pi(G_b)$ to the receiver.

Reveal stage: sender $S_{(G_0,G_1)}$ sends (b, π) to the receiver. Receiver $R_{(G_0,G_1)}$ *accepts* if $\pi(G_b) = J$, and *rejects* otherwise.

This instance-dependent commitment scheme for Graph Isomorphism is statistically hiding on the YES instances, and statistically binding on the NO instances. For a YES instance $(G_0, G_1) \in$ Graph Isomorphism, a commitment to b is a uniformly random permutation over the nodes of G_b. Since G_0 and G_1 are isomorphic, the commitments to 0 and 1 are identically distributed, thus yielding the statistical hiding (in fact, *perfect* hiding) property. For a NO instance $(G_0, G_1) \notin$ Graph Isomorphism, a commitment to b is also a uniformly random permutation over the nodes of G_b. In this case, however, G_0 and G_1 are not isomorphic, so distributions of the commitments to 0 and 1 are disjoint. (If not, there will exist permutations π and π' such that $\pi(G_0) = \pi'(G_1)$, contradicting the assumption that G_0 and G_1 are isomorphic graphs.) This yields the statistical binding (in fact, *perfect* binding) property.

As demonstrated by this Graph Isomorphism example, instance-dependent commitments have an edge over standard commitments because they can be both statistically hiding and statistically binding, and they can be obtained unconditionally; in Section 3.6 and Chapter 4, we prove that instance-dependent commitments can be obtained unconditionally from every problem having zero-knowledge protocols. On the other hand, standard commitment schemes do require computational assumptions, and we will see this connection next.

Constructing standard commitments based on any one-way function

Naor [Nao] constructed computationally-hiding and statistically-binding commitments from any *pseudorandom generator*, which in turn can be based on any one-way function [HILL]. The combined results of [Nao, HILL] is viewed as settling the complexity of computationally-hiding and statistically-binding commitments because the existence of one-way functions is the minimal complexity assumption needed for these commitments [IL].

PROPOSITION 2.4.13
(From [Nao, HILL].) If one-way functions exist, then there exist commitment schemes that are computationally hiding and statistically binding. Moreover, the instance-dependent commitment scheme obtained is public coin and constant round.

It turns out that the statistical binding property of Naor's scheme does not depend on the one-way security of the function. Thus, we can construct an instance-dependent commitment scheme from any instance-dependent function such that the scheme is always statistically binding, but is guaranteed to be computationally hiding only on the instances where the function is hard to invert. This is stated in the following proposition, which can be viewed as an instance-dependent formulation of Proposition 2.4.13.

PROPOSITION 2.4.14
(Follows from [Nao, HILL].) For every set $K \subseteq \{0,1\}^*$, if there is an instance-dependent one-way function on K, then problem $(K, \overline{K})$ has an instance-dependent commitment scheme that is computationally hiding on the YES instances (namely, instances in K), and statistically binding on the NO instances (namely, instances in $\overline{K}$). Moreover, the instance-dependent commitment scheme obtained is public coin and constant round.

In Chapter 3, we derive analogous results for the construction of *statistically-hiding* and computationally-binding commitments from one-way functions (see Theorem 3.0.4 and Proposition 3.5.44).

2.5 Zero-Knowledge Protocols from Instance-Dependent Commitments

In this section, we formalize what we informally concluded at the end of Section 1.1, which is that having an instance-dependent commitment scheme for a problem $\Pi \in \text{NP}$ is sufficient to *unconditionally* construct a zero-knowledge protocol for Π.

Following Itoh, Ohta, and Shizuya [IOS], we substitute standard commitments in existing zero-knowledge protocols with instance-dependent commitments. Specifically, we do this substitution in the Blum zero-knowledge protocol [Blu]. We use Blum's protocol for round efficiency; at a cost of having more rounds for comparable soundness error, we can also use the Goldreich, Micali & Wigderson protocol [GMW2].

We present a description of Blum's protocol in Protocol 2.5.1, where we explicitly substitute instance-dependent commitments for standard commitments. Before presenting Protocol 2.5.1, a few definitions are in place. A ***Hamiltonian cycle*** in a directed graph is a directed path that goes through each and every node exactly once and returns to the starting node. The problem Hamiltonian Path is the task of determining whether a given directed graph G has a Hamiltonian cycle, and this problem is NP-complete [Kar]. The ***adjacency matrix*** of a directed graph G is a matrix with rows and columns labeled by the nodes of G, with 1 or 0 in position (v_i, v_j) according to whether there is an edge from v_i to v_j or not—1 if there is an edge, and 0 if an edge is not present.

PROTOCOL 2.5.1

Protocol (P, V) for problem $\Pi \in \mathrm{NP}$ using instance-dependent commitment scheme Com_x.

Common input: instance x of the problem Π.

Auxiliary input to P: a witness w for x such that $(x, w) \in W$, where W is an NP-relation for Π.

1. P and V: Reduce the instance x to a directed graph G of the Hamiltonian Path problem.
2. P: Reduce the NP-witness w to a Hamiltonian cycle C in the graph G. Select a uniform permutation π over the nodes of G, and define the graph $G' = \pi(G)$.
3. $P \leftrightarrow V$: P commits to V the entire adjacency matrix of G' (entry by entry) using Com_x.
4. $V \to P$: Send a random bit $c \leftarrow \{0, 1\}$.
5. $P \to V$: If $c = 0$, reveal the permutation π and decommit all entries of the adjacency matrix of G'. Otherwise, if $c = 1$, reveal only the commitments to entries that correspond to the cycle C, i.e., $(\pi(i), \pi(j))$ where $(i, j) \in C$.
6. V: *Accept* if $c = 0$ and $\pi(G) = G'$, or if $c = 1$ and the revealed entries correspond to a simple cycle of length n. (Also, check that the revealed committed values are valid.) Otherwise, *reject*.

The hiding and binding properties of the commitment scheme Com_x translates to the zero knowledge and soundness properties of protocol (P, V), respectively.

PROPOSITION 2.5.2

(Based on [Blu].) If Com_x is an instance-dependent commitment scheme for problem $\Pi \in \mathrm{NP}$, then Protocol 2.5.1 is:

- statistical [resp., computational] zero knowledge if Com_x is statistically [resp., computationally] hiding on the YES instances, and
- a proof [resp., argument] system if Com_x is statistically [resp., computationally] binding on the NO instances.

An immediate corollary of the above proposition is that standard commitments are also sufficient for zero-knowledge protocols for NP, because standard commitments are stronger than instance-dependent commitments. Standard commitments, however, require complexity assumptions, whereas instance-dependent commitments sometimes do not.

COROLLARY 2.5.3

- If there exist commitment schemes that are statistically [resp., computationally] hiding and *statistically binding*, then there exist statistical [resp., computational] zero-knowledge *proof systems* for all of NP.
- If there exist commitment schemes that are statistically [resp., computationally] hiding and *computationally binding*, then there exist statistical [resp., computational] zero-knowledge *argument systems* for all of NP.

Because the main ideas behind Proposition 2.5.2 have been presented by Blum [Blu] and clarified in many other works (e.g., [Gol2, BL, BLV]), we only give a proof sketch of this proposition.

Proof Sketch of Proposition 2.5.2. It is straightforward to see that Protocol 2.5.1 has perfect completeness. We argue the zero knowledge and soundness properties as follows.

Zero knowledge. Let us analyze the verifier's view at the end of the interaction. If $c = 0$, then all it sees is a random permutation π of the graph G, which it can clearly generate on its own. Otherwise, if $c = 1$, then it just sees a random simple cycle of length n. This is because the cycle C in G has been permuted into a random cycle $\pi(C)$ in $G' = \pi(G)$. Again, it can clearly generate a random cycle on its own.

Soundness. Consider a cheating prover P^* and the case where $x \in \Pi_{\mathrm{N}}$, which translates to the graph G not having a Hamiltonian cycle. We claim that after the prover P^* has committed to the adjacency matrix of *any* graph G' (not necessarily one chosen according to the protocol), then it cannot answer both the verifier's challenges $c = 0$ and $c = 1$ in a way that makes the verifier accept, unless P^* breaks the binding property of the commitment (which happens with negligible probability). For simplicity, assume the commitments are always binding. If P^* can successfully answer to $c = 0$, then there is a permutation π of G such that $\pi(G) = G'$, and hence we know that G and G' are isomorphic graphs. If P^* can also successfully answer to $c = 1$, then there is a Hamiltonian cycle in G'. Consequently, graph G, being isomorphic to G', has a Hamiltonian cycle too, which contradicts the case where $x \in \Pi_{\mathrm{N}}$.

To reduce the soundness error while maintaining the zero knowledge property and round complexity, we repeat Protocol 2.5.1 a total of $O(\log n)$ times in *parallel* (cf., [BL, BLV]).

Conclusion. We conclude this chapter with the following observation based on Proposition 2.5.2.

> An instance-dependent commitment scheme for a problem in NP is sufficient to *unconditionally* construct a zero-knowledge protocol for that problem, and that the hiding and binding properties of the commitment scheme translates to the zero knowledge and soundness properties of the constructed protocol, respectively.

In the next chapter, we begin our technical exposition of the results contained in this dissertation, starting with statistically-hiding commitments and statistical zero-knowledge protocols.

3

STATISTICALLY-HIDING COMMITMENTS

The two main focuses of this present chapter is the complexity of statistically-hiding commitment schemes, and the relationship between statistically-hiding commitments and statistical zero-knowledge protocols. For the first, we investigate the minimal complexity assumption needed to construct statistically-hiding commitments, and prove the following theorem.

THEOREM 3.0.4
(First appeared in [HR2, Thm. 1.1].) If one-way functions exist, then there exist commitment schemes that are statistically hiding and computationally binding. Moreover, the commitment schemes obtained are public coin.

For the second, we study the relationship between statistically-hiding commitments and statistical zero-knowledge protocols. In Section 2.5, we showed that instance-dependent commitments for a problem $\Pi \in \mathrm{NP}$ that are statistical hiding on the YES instances and statistically binding on the NO instances imply that Π has a statistical zero-knowledge proof system (i.e., $\Pi \in \mathrm{SZKP}$). Here, we prove the converse.

THEOREM 3.0.5
Every problem in SZKP has an instance-dependent commitment scheme that is *statistically hiding* on the YES instances and *statistically binding* on the NO instances. Moreover, this instance-dependent commitment scheme is public coin and is constant round.

These are the two main theorems developed in this chapter. It will be helpful to keep these theorems in mind as they are used to establish the results in Chapter 4.

Chapter organization. The complexity of statistically-hiding commitment schemes, the first focus of this chapter, is explored in Sections 3.1 through 3.5. In Section 3.1, we review the history of the construction of statistically-hiding commitments from a complexity-theoretic viewpoint. After that, we work progressively towards lowering the complexity assumptions needed to construct statistically-hiding commitments: in Section 3.2, we present a construction based on *one-way permutations*; in Section 3.3, we show a construction based on *regular one-way functions* with *known* preimage size; in Section 3.4, we give a construction based on regular one-way functions with *unknown* preimage size; and finally in Section 3.5, we develop a construction based on any one-way function, which is the minimal complexity assumption needed [IL].

The second focus of this chapter, explored in Section 3.6, is the relationship between statistically-hiding commitments and statistical zero-knowledge protocols. In that section, we construct instance-dependent commitment schemes for any problem in SZKP; our construction is *unconditional* in that it does not rely on any unproven complexity assumptions.

3.1 A Complexity-Theoretic History

Unlike their computational counterpart (namely, computationally-hiding commitments) whose existence has been shown to be equivalent to the existence of one-way functions by 1990 [HILL, Nao], the complexity of statistically-hiding commitments was still an open problem up till very recently. The early constructions of statistically-hiding commitments were based on specific number-theoretic complexity assumptions like *hardness of factoring large integers* [BCC] and *hardness of discrete logarithm* [BKK, CDG, Ped].[1] Later constructions have reduced the complexity assumption to any family of *claw-free permutations* [GK1], and then to any *collision-resistant hash family* [NY] (see also [DPP]).[2] All these assumptions, explicitly or implicitly, have a collision resistance property attached to them, and for quite a while it seemed that this property might be necessary for statistically-hiding commitments.

In 1992, Naor, Ostrovsky, Venkatesan and Yung [NOVY] dispelled the idea that a collision resistance criterion is needed by giving an elegant construction of statistically-hiding (in fact, perfectly-hiding) commitments from any *one-way permutation*.[3] They left as an open question whether statistically-hiding commitments can be based on any one-way function, the minimal complexity assumption needed [IL]. The first progress in the past decade came

[1] The constructions of Boyar et al. [BKK], Chaum et al. [CDG] and Pedersen [Ped] yield *perfectly*-hiding commitments. In perfectly-hiding schemes, commitments to any message are identically distributed.

[2] The fact that claw-free permutations imply a collision-resistant hash family was shown in [GMR2, Dam1], and the early constructions of claw-free permutations based on specific number-theoretic complexity assumptions were given by [GMR2, BKK].

[3] A ***one-way permutation*** is a one-way function that is also a permutation. We note that one-way permutations and collision-resistant hashing are known to be incomparable under *black-box reductions* [Sim, Rud, KSS].

in 2005 when Haitner et al. [HHK$^+$] constructed statistically-hiding commitments from any *regular one-way function with known preimage size*.[4] (Actually, their construction is more general in that in works for any *approximable preimage size one-way function*, which is a one-way function where we can efficiently approximate the preimage size of points in the range.)

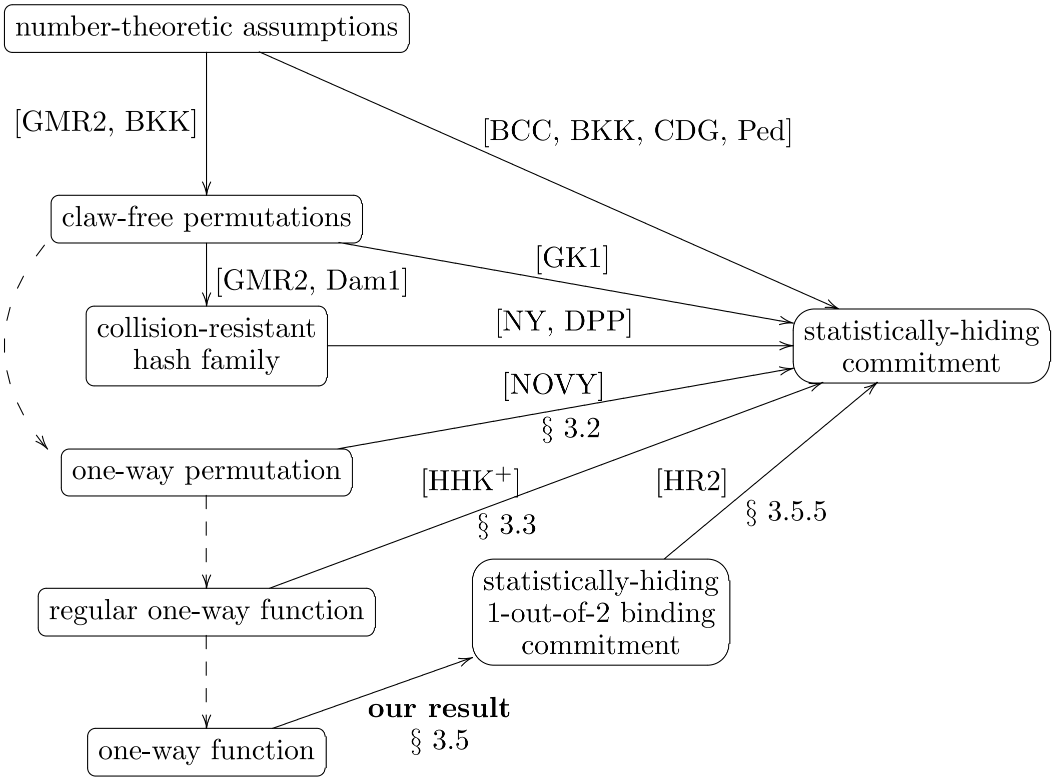

Figure 3.1: A complexity-theoretic history of statistically-hiding commitments.

Inspired by this recent development, we construct a relaxed variant of statistically-hiding commitment schemes called *1-out-of-2-binding commitments* (a notion introduced by Nguyen and Vadhan [NV]), based on any one-way function. These commitments have a weaker 1-out-of-2 binding property, a notion that will be discussed in Section 3.4.1. Up till very recently, we do not know if these 1-out-of-2-binding commitments would yield commitments with the standard binding property. Nevertheless, it turns out that 1-out-of-2-binding commitments suffice for obtaining statistical zero-knowledge arguments for all of NP, and thus our result shows that statistical zero-knowledge arguments for NP can be based on any one-way function (Theorem 1.2.5).

Building on our result, Haitner and Reingold [HR2] settled the complexity of statistically-hiding commitments by providing an elegant transformation technique converting 1-out-

[4] A ***regular one-way function*** is a one-way function where all points in the range have the same preimage size. Refer back to Section 2.4.1 if needed.

of-2-binding commitments into commitments with the standard binding property. Their transformation uses a novel application of a *universal one-way hash family*, a cryptographic primitive that can constructed from any one-way function [Rom] (see also [KK]).

Figure 3.1 summarizes the complexity-theoretic history of statistically-hiding commitment schemes as presented in this section. The dotted arrows in Figure 3.1 indicate trivial implications, and the section numbers, preceded by the symbol §, indicate where those results are presented in this chapter.

3.2 From One-Way Permutations

Consider a one-way permutation $f\colon \{0,1\}^n \to \{0,1\}^n$. Naor, Ostrovsky, Venkatesan, and Yung [NOVY] obtained a statistically-hiding commitment scheme based on f by using a protocol called *interactive hashing* as a subroutine. Our agenda for this section is as follows: we will first informally describe interactive hashing and state the two main properties that we want from it; then, in Section 3.2.1 we give an informal description of the Naor et al. scheme, henceforth called the NOVY commitment scheme; and finally, in Section 3.2.2, we give a formal definition of interactive hashing and a protocol satisfying that definition.

Interactive hashing is a protocol between a sender S_{IH} and receiver R_{IH}. The sender begins with a private input z, and at the end both parties outputs z_0 and z_1 such that $z \in \{z_0, z_1\}$. Informally, an interactive hashing protocol has the following two properties:

1. *Hiding*: if the sender's private input z is uniformly random, then every receiver, even computationally-unbounded malicious ones, does not learn which of z_0 or z_1 equals to z, and

2. *Binding*: the sender, including PPT malicious ones, can only control the value of at most one of the two outputs, and the value of the other output that it does not control is uniformly distributed.

3.2.1 The NOVY commitment scheme

Using an interactive hashing protocol as a subroutine, Naor et al. [NOVY] constructed the following statistically-hiding commitment scheme.

1. S chooses a uniform $x \leftarrow \{0,1\}^n$, and computes $z = f(x)$.

2. S and R engage in an interactive hashing protocol. Let z_0 and z_1 be the common outputs, and let $z = z_d$, for some $d \in \{0,1\}$, be S's private output.

3. To commit to bit b, S sends $c = b \oplus d$ to R.

4. To decommit, S sends b, c, d, and x to R. R verifies the decommitment by checking if $c = b \oplus d$ and $z_d = f(x)$.

Let us informally argue why the above scheme constitutes a statistically-hiding and computationally-binding commitment. First, we argue its hiding property. We have mentioned that z is uniform in $\{0,1\}^n$ because f is a permutation and x is chosen uniformly in $\{0,1\}^n$. By the hiding property of interactive hashing, even a computationally-unbounded malicious receiver does not know if $z = z_0$ or $z = z_1$, or equivalently, it does not know if $d = 0$ or $d = 1$. Therefore, the scheme is statistically hiding. Next, we argue its binding property. By the binding property of interactive hashing, at least one of the outputs, say z_α, is uniform in $\{0,1\}^n$ and outside the sender's control. Therefore if the sender is able to decommit to both 0 and 1, it must find a preimage of z_α. This is equivalent to finding a preimage of $f(U_n)$, and this task is computationally infeasible since f is a one-way permutation. Hence, the scheme is computationally binding.

3.2.2 Interactive hashing

Interactive hashing was introduced by Ostrovsky, Venkatesan, and Yung [OVY] in the context of *oblivious transfer* protocols. Although we do not study oblivious transfer, interactive hashing will prove to be a powerful and useful tool in our construction of statistically-hiding commitments based on any one-way function. For our application, we will need the sender to commit to multiple bits in one execution of interactive hashing. Consequently, we extend the notion of interactive hashing to allow multiple outputs (instead of just two output strings). Since the number of outputs could be possibly superpolynomial, we succinctly describe the set of outputs as the image of a polynomial-sized circuit $C\colon \{0,1\}^k \to \{0,1\}^q$, where k and q are polynomially related to the security parameter.

In addition to allowing for multiple outputs, our application of interaction hashing also requires a more refined notion of computational binding that the one provided by Naor, Ostrovsky, Venkatesan, and Yung [NOVY].[5] It is for this reason we define the notion of what it means to be a witness for a given relation W as follows: For a relation W, define the ***set of witnesses*** for z as $W_z = \{x : W(z,x) = 1\}$, and we naturally refer to any $x \in W_z$ as a ***witness*** for z.

DEFINITION 3.2.1

An ***interactive hashing with multiple outputs*** protocol is a polynomial-time protocol $(S_{\mathrm{IH}}, R_{\mathrm{IH}})$ where both parties receive common inputs $(1^q, 1^k)$ and S_{IH} receives a private input $z \in \{0,1\}^q$. At the end of the interaction, the common output is a polynomial-sized circuit $C\colon \{0,1\}^k \to \{0,1\}^q$, and the private output of S_{IH} is a string $d \in \{0,1\}^k$. We call q the input length, and k the output length. The protocol $(S_{\mathrm{IH}}, R_{\mathrm{IH}})$ has to satisfy the following security properties.

[5] Although the notion of interactive hashing was introduced by Ostrovsky et al. [OVY], it was Naor et al. [NOVY] who proved a computational binding property of interactive hashing that allows for its application to statistically-hiding commitments based on any one-way permutation.

1. *Correctness*: for all R^* and all $z \in \{0,1\}^q$, it is the case that $C(d) = z$, where $C = (S_{\mathrm{IH}}(z), R^*)(1^q, 1^k)$ is the common output, and $d = \mathrm{output}_{S_{\mathrm{IH}}}(S_{\mathrm{IH}}(z), R^*)$ is the private output of S_{IH}.[6]

2. *Hiding*: for all R^*, random variables (V, Z) and (V, U_k) are identically distributed, where the view of receiver R^* is $V = \mathrm{view}_{R^*}(S_{\mathrm{IH}}(U_q), R^*)$, and the private output of S_{IH} is $Z = \mathrm{output}_{S_{\mathrm{IH}}}(S_{\mathrm{IH}}(U_q), R^*)$.

3. *Binding*: there exists an oracle PPT algorithm A such that for every adversary S^* and any relation W, denoting the common output as $C = (S^*, R_{\mathrm{IH}})(1^q, 1^k)$, and private output of S^* as $((x_0, d_0), (x_1, d_1)) = \mathrm{output}_{S^*}(S^*, R_{\mathrm{IH}})$, if it is the case that

$$\Pr[x_0 \in W_{C(d_0)} \wedge x_1 \in W_{C(d_1)} \wedge d_0 \neq d_1] > \varepsilon \ ,$$

where the above probability is over the coins of R_{IH} and S^*, then it is also the case that

$$\Pr_{z \leftarrow \{0,1\}^q}[A^{S^*}(z, 1^q, 1^k, \varepsilon) \in W_z] > 2^{-k} \cdot (\varepsilon/q)^{O(1)} \ .$$

REMARK 3.2.2

We make three remarks regarding Definition 3.2.1.

1. The security requirements should hold for computationally unbounded R^* (for correctness and hiding) and computationally unbounded S^*. In addition, the relation W need not be polynomial-time computable.

2. To simplify notation, we often write $A^{S^*}(z)$, or even $A(z)$, to denote $A^{S^*}(z, 1^q, 1^k, \varepsilon)$. Since we are dealing with nonuniform security (see discussion in Section 2.3.4), we can hardwire the (approximate) value of ε as nonuniform advice.

3. Although the private output of the honest sender S_{IH} is always a string d, the private output of the cheating sender S^* is arbitrary; hence, we can assume without loss of generality that S^* breaks binding by producing two pairs of strings (x_0, d_0) and (x_1, d_1).

The interactive hashing protocol given in [OVY, NOVY], henceforth called the NOVY Interactive Hashing, satisfies Definition 3.2.1 with $k = 1$. To obtain an interactive hashing with multiple outputs protocol (i.e., the case when $k > 1$), we simply end the NOVY Interactive Hashing protocol $k - 1$ rounds earlier.

[6] The correctness property of protocols is typically defined for honest parties, in our setting this would be S_{IH} and R_{IH}. Our applications, however, need a stronger correctness property that would hold against malicious receivers R^*.

PROTOCOL 3.2.3

Interactive hashing with multiple outputs $(S_{\mathrm{IH}}, R_{\mathrm{IH}})$.

Inputs:

1. Input length 1^q and output length 1^k, both given as common input.
2. String $z \in \{0,1\}^q$, given as private input to sender S_{IH}.

Protocol:

R_{IH}: Select $h_0, h_1, \ldots, h_{q-k-1}$ such that each h_i is a random vector over GF[2] of the form $0^i 1\{0,1\}^{q-i-1}$ (i.e., i number of 0's followed by a 1, and random choice for the last $q-i-1$ positions).

For $j = 0, \ldots, q-k-1$, do the following:

$R_{\mathrm{IH}} \to S_{\mathrm{IH}}$: Send h_j.

$S_{\mathrm{IH}} \to R_{\mathrm{IH}}$: Send $c_j = \langle h_j, z \rangle$.

Output:

- Common output is a circuit $C\colon \{0,1\}^k \to \{0,1\}^q$ computing an affine transformation whose image is $\{z : \langle h_j, z\rangle = c_j \;\; \forall j = 0, \ldots, q-k-1\}$.
- Private output of S_{IH} is a string $d \in \{0,1\}^k$ such that $C(d) = z$. (In fact, d can be taken to be the last k bits of z.)

We defer the proof that Protocol 3.2.3 satisfies Definition 3.2.1 to Appendix A.1, as it is just an extension of the proof given in [NOVY]; here we just state the theorem that follows.

THEOREM 3.2.4

There exists an interactive hashing with multiple outputs protocol, namely Protocol 3.2.3.

Information-theoretic bounds

We think of the string d as a k-bit string commitment associated to one of the 2^k outputs strings, namely $z = C(d)$, and a witness $x \in W_z = W_{C(d)}$ as a decommitment to d. Intuitively, the knowledge of x gives the sender the ability to decommit to d. The binding property, read in its contrapositive, says that if it is hard to find a witness for a uniformly random string z, then it is hard for a sender to successfully decommit to two different values. Notice that this property holds even if the set of z's for which is it hard to find a witness is not fixed in advance, but depends on the algorithm trying to find a witness for z (namely, an element in W_z). In several places, however, we will only need the special case of a static set of z's as captured in the following lemma.

LEMMA 3.2.5

(Binding for Static Sets.) For any protocol $(S_{\text{IH}}, R_{\text{IH}})$ satisfying the binding condition of Definition 3.2.1, the following holds: For all S^* and any set $\Gamma \subseteq \{0,1\}^q$, denoting the common output as $C = (S^*, R_{\text{IH}})(1^q, 1^k)$, we have

$$\Pr[\exists d_0 \neq d_1 \text{ such that } C(d_0), C(d_1) \in \Gamma] < (\mu(\Gamma) \cdot 2^k)^{\Omega(1)} \cdot \text{poly}(q) \ ,$$

where the above probability is taken over the coins of S^* and R_{IH}.

Setting $k = 1$ in the above lemma gives an information-theoretic bound of the NOVY Interactive Hashing; information-theoretic bounds on NOVY Interactive Hashing were studied in the context of memory-bounded oblivious transfer [CCM, DHRS, CS]. Our bound is not tight, but suffices for our applications. For tighter bounds, we refer the reader to [CCM, CS], or for a *constant-round* interactive hashing protocol that is binding for static sets, we refer the reader to [DHRS].

Compare the bound of the Lemma 3.2.5 to the case where the adversarial sender S^* had control of only one output string. This means that the rest of the $2^k - 1$ outputs strings are distributed uniformly on $\{0,1\}^q$, and hence the bound would be $\mu(\Gamma) \cdot (2^k - 1)$. The reason for this is that S^* will make the string that it controls lie in Γ, and the probability that at least one of the rest of the $2^k - 1$ strings lie in Γ is at most $\mu(\Gamma) \cdot (2^k - 1)$, by a union bound argument. The above bound is almost as good, and in particular if $\mu(\Gamma)$ is negligible and k logarithmic, both probabilities are negligible.

Proof of Lemma 3.2.5. Define the relation $W = \{(a,b) : a \in \Gamma\}$, that is $W(a,b) = 1$ if $a \in \Gamma$ (for all values of b), and 0 if $a \notin \Gamma$ (no matter what the value of b is). Suppose there exists an S^* that with probability ε, produces two elements $d_0 \neq d_1$ such that both $C(d_0), C(d_1) \in \Gamma$. Then, by the binding condition of Definition 3.2.1, there will be a procedure that is given a random $z \leftarrow \{0,1\}^q$ makes $z \in \Gamma$ with probability $2^{-k} \cdot (\varepsilon/q)^{O(1)}$. Since Γ is a fixed set, it must be the case that $2^{-k} \cdot (\varepsilon/q)^{O(1)} \leq \mu(\Gamma)$. This implies that $\varepsilon < (\mu(\Gamma) \cdot 2^k)^c \cdot \text{poly}(q)$, for some constant $c > 0$. □

3.3 From Regular One-Way Functions with Known Preimage Size

Our first hurdle is to relax the permutation structure of f to just assuming that f is a regular one-way function with *known* preimage size of say 2^{n-t}, for some known value of $t \in \{1, 2, \ldots, n\}$. This is the setting considered by Haitner et al. [HHK$^+$], and we review ideas from their construction in this section. To simplify the construction and analysis, we further assume f has a *known* superpolynomial security $s(n) = n^{\omega(1)}$. (We stress that Haitner et al. [HHK$^+$] does not make this assumption.)

Observe that the statistical hiding property of the NOVY commitment scheme based on one-way permutation f only rely on the fact that f is a permutation because we require that $f(U_n)$ be uniform. Now if f just a regular function, then $f(U_n)$ might no longer be uniform, but instead all we can say is that $f(U_n)$ is a flat distribution with support $\mathrm{Supp}(f(U_n))$ of size 2^t. We will use *pairwise-independent hash functions*, a notion to be discussed next, to obtain an *almost-uniform* distribution from $f(U_n)$.

3.3.1 Hashing and randomness extraction

A family of hash functions $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ is ***pairwise independent*** if for any two $x \neq x' \in \{0,1\}^n$ and any two $y, y' \in \{0,1\}^m$, when we randomly choose $h \leftarrow \mathcal{H}$, we have $\Pr[h(x) = y \text{ and } h(x') = y'] = 2^{-2m}$.

An example of a pairwise-independent family of hash functions is the family $\mathcal{H} = \{h_{a,b} \colon \{0,1\}^n \to \{0,1\}^m\}$, where $h_{a,b}(x) = (a \cdot x + b)|_m$, arithmetic is done in the field $\mathrm{GF}(2^n)$, and $|_m$ denote taking the first m bits. We define $\ell(n,m)$ to be the number of bits required to describe an element of the hash function family $\mathcal{H}$. In our example, it takes $2n$ bits to describe each hash function $h_{a,b}$ since both a and b are elements of $\mathrm{GF}(2^n)$; hence, we now know that a family of pairwise-independent hash functions $\mathcal{H}$ mapping n-bit strings to m-bit strings exists with $\ell(n,m) = 2n$. We will use the following property of pairwise-independent hash functions to obtain an almost-uniform random variable from a random variable with sufficient *min-entropy*. (The definition of min-entropy is given in Section 2.2.1.)

LEMMA 3.3.1

(Leftover Hash Lemma [BBR, ILL].) Let random variable H denote a uniformly random hash function from a family of pairwise-independent hash functions $\mathcal{H}$ mapping n-bit strings to m-bit strings, and let X be a random variable taking values in $\{0,1\}^n$. For any $\varepsilon > 0$, if the min-entropy $\mathrm{H}_\infty(X) \geq m + 2\log(1/\varepsilon)$, and H is independent from X, then random variable $(H, H(X))$ is ε-close in statistical distance to uniform.

3.3.2 The commitment scheme

Let us return to our regular one-way function $f \colon \{0,1\}^n \to \{0,1\}^n$ with known preimage size 2^{n-t} and known security $s(n) = n^{\omega(1)}$.[7] Consider a family of pairwise-independent hash functions $\mathcal{H} = \left\{h \colon \{0,1\}^n \to \{0,1\}^{t-\Delta}\right\}$, where $t = \mathrm{H}(f(U_n))$ and $\Delta = \frac{1}{2}\log s(n)$. Let

[7] To avoid introducing new parameters, we consider only length-preserving functions, that is $|f(x)| = |x|$ for all $x \in \{0,1\}^*$. Our construction, nevertheless, can be easily generalized to regular one-way functions that are not length preserving.

random variable H represent a random hash function selected from $\mathcal{H}$. By the Leftover Hash Lemma 3.3.1, random variable $Z = (H, H(f(U_n)))$ is $(1/s(n))^{\Omega(1)}$-close to uniform, which in turn is statistically close to uniform since $s(n) = n^{\omega(1)}$. So if we designate $z = (h, h(f(x)))$ as the sender's private input to the interactive hashing protocol (Protocol 3.2.3), even an all-powerful receiver will not get more than a negligible advantage to guess which one of the outputs is z. This hints to the following commitment scheme.

PROTOCOL 3.3.2

Commitment scheme (S, R) based on a regular one-way function $f\colon \{0,1\}^n \to \{0,1\}^n$ with known preimage size 2^{n-t} and known security $s(n) = n^{\omega(1)}$.

Commit stage.

1. Let $\mathcal{H} = \left\{h\colon \{0,1\}^n \to \{0,1\}^{t-\Delta}\right\}$, where $t = \mathrm{H}(f(U_n))$ and $\Delta = \frac{1}{2}\log s(n)$. S selects a uniform $x \leftarrow \{0,1\}^n$ and hash function $h \leftarrow \mathcal{H}$, and computes $y = f(x)$ and $z = (h, h(y))$.
2. S and R engage in interactive hashing (Protocol 3.2.3) with S acting as S_{IH}, R acting as R_{IH}, parameters $k = 1$ and $q = |z|$, and S_{IH} having private input z. Their common output is a circuit $C\colon \{0,1\} \to \{0,1\}^q$, and the sender receives a bit $d \in \{0,1\}$ such that $C(d) = z$.
3. To commit to the bit b, S sends $c = d \oplus b$ to R. The commitment of b is represented as the pair (C, c).

Reveal stage. To decommit, S sends bits b and d, string x, and hash function h to R. R verifies the decommitment by checking if $c = d \oplus b$ and $C(d) = (h, h(f(x)))$.

As we have argued previously, the sender's private input z is statistically close to uniform, and hence by the hiding property of interactive hashing, this implies that the commitment scheme is statistically hiding. As for the binding property, the one-wayness of f intuitively guarantees that the set Γ of w's for which a sender S^* can compute an element of $f^{-1}(w)$ is of density at most $1/s(n)$ in the range of f, that is the size of Γ is at most $2^{\mathrm{H}(f(U_n))-\log s(n)}$. Thus for any fixed h, the fraction of $z = (h, h(w))$ such that $w \in \Gamma$ is at most $2^{\mathrm{H}(f(U_n))-\log s(n)}/2^{t-\Delta} = s(n)^{-1/2} = \mathrm{neg}(n)$. By the binding property of interactive hashing (refer to Lemma 3.2.5), the probability that S^* can force both $C(0), C(1) \in \Gamma$ is negligible and hence, the scheme is computationally binding. The complete argument to prove the binding property is actually more subtle because the set Γ is not actually fixed in advance, and so we need to employ the binding property given in Definition 3.2.1.

3.4 From Regular One-Way Functions with Unknown Preimage Size

Our next hurdle is to remove to the constraint on knowing (i.e., being able to efficiently compute) the preimage size. For this setting, let us consider a regular one-way function $f\colon \{0,1\}^n \to \{0,1\}^n$ with preimage size 2^{n-t}, for an *unknown*[8] value of $t \in \{1, 2, \ldots, n\}$, but with known security $s(n) = n^{\omega(1)}$.[9] Constructing statistically-hiding commitments even in this setting was still an open problem prior to our work.

Let us examine why we need to know the correct value of t in the previous scheme of Protocol 3.3.2. If the value of t is too high, that is $t \gg \mathrm{H}(f(U_n))$, then the scheme is no longer hiding (but would be binding). This is because the Leftover Hash Lemma 3.3.1 no longer applies, since in this case the min-entropy $\mathrm{H}(f(U_n))$ is too small relative to t. On the other hand, if the value of t is too low, that is $t \ll \mathrm{H}(f(U_n))$, then the scheme is no longer binding (but would be hiding). To see this, at least intuitively, observe that when t is very small, we are hashing $f(U_n)$ to a very small set $\{0,1\}^{t-\Delta}$; in other words, h collapses too many elements in $f(U_n)$. As a consequence, inverting $h(f(U_n))$ could be easy (even though inverting $f(U_n)$ is hard), and this allows us to break the binding property of our scheme.

All hope, however, is not lost. We can still use Protocol 3.3.2, trying all values of $t \in \{1, 2, \ldots, n\}$, to do our *first phase* commitments. And to overcome the difficulty of ensuring both hiding and binding, we will introduce a *second phase* that will be binding when $t \lesssim H(f(U_n))$, and hiding when $t \gtrsim H(f(U_n))$; this is obtained by the sender using a hash of the preimage x as an input to another execution of interactive hashing. This means that for the right value of $t = H(f(U_n))$, both phases will be hiding, but for any value of t, at least one phase is binding. What we are describing here is a ***2-phase commitment scheme*** with a ***1-out-of-2 binding*** property, notions that we formally define in the next section.

3.4.1 2-phase commitment schemes

As mentioned previously, we will work with 2-phase commitment schemes, an alternate variant of commitments introduced by Nguyen and Vadhan [NV]. These are commitment schemes with two *sequential* and *related* stages such that in each stage, the sender commits to and reveals a value.

[8] What we mean by *unknown* is that we are not able to compute the preimage size efficiently.

[9] Like in Section 3.3, we consider only length-preserving functions, that is $|f(x)| = |x|$ for all $x \in \{0,1\}^*$, to avoid introducing new parameters. Our construction can nevertheless be easily generalized to regular one-way functions that are not length preserving.

DEFINITION 3.4.1

A ***2-phase commitment scheme*** (S, R), with security parameter n and message lengths $(k_1(n), k_2(n))$, consists of four interactive protocols: the first commitment stage (S_c^1, R_c^1), the first reveal stage (S_r^1, R_r^1), the second commitment stage (S_c^2, R_c^2), and the second reveal stage (S_r^2, R_r^2). For us, both reveal phases will always be noninteractive, consisting of a single message from the sender to the receiver.

1. In the first commitment stage, S_c^1 receives a private input $\sigma^{(1)} \in \{0,1\}^{k_1}$ and coin tosses r_S. At the end of the interaction, both S_c^1 and R_c^1 output a commitment $c^{(1)}$. (Without loss of generality, we can assume that $c^{(1)}$ is the transcript of the first commitment stage.)

2. In the first (noninteractive) reveal stage, both S_r^1 and R_r^1 receive as common inputs the commitment $c^{(1)}$, and S_r^1 receives as private input its previous coin tosses r_S. S_r^1 sends R_r^1 a pair $(\sigma^{(1)}, \gamma^{(1)})$ with $\gamma^{(1)}$ interpreted as a decommitment for $\sigma^{(1)} \in \{0,1\}^{k_1}$. R_r^1 accepts or rejects based on $c^{(1)}$, $\sigma^{(1)}$, and $\gamma^{(1)}$. After that, both S_r^1 and R_r^1 outputs a string τ. (Without loss of generality, we can assume that τ is the transcript of the first commitment stage and the first reveal stage and includes R_r^1's decision to accept or reject.)

3. In the second commitment stage, both S_c^2 and R_c^2 receive as common input the string τ, and S_c^2 receives a private input $\sigma^{(2)} \in \{0,1\}^{k_2}$ and its previous coin tosses r_S. At the end of the interaction, both S_c^2 and R_c^2 output a commitment $c^{(2)}$. (Without loss of generality, we can assume that $c^{(2)}$ is the concatenation of τ and the transcript of the second commitment stage.)

4. In the second (noninteractive) reveal stage, both S_r^2 and R_r^2 receive as common input the commitment $c^{(2)}$, and S_r^2 receives as private input its previous coin tosses r_S. S_r^2 sends R_r^2 a pair $(\sigma^{(2)}, \gamma^{(2)})$ with $\gamma^{(2)}$ interpreted as a decommitment for $\sigma^{(2)} \in \{0,1\}^{k_2}$. R_r^2 accepts or rejects based on $c^{(2)}$, $\sigma^{(2)}$, and $\gamma^{(2)}$.

- We insist that scheme (S, R) have ***perfect completeness***. That is to say, if both sender S and receiver R follow their prescribed strategy, then R will always accept (with probability 1).

- The sender and receiver's algorithms, denoted by $S = (S^1, S^2) = ((S_c^1, S_r^1), (S_c^2, S_r^2))$ and $R = (R^1, R^2) = ((R_c^1, R_r^1), (R_c^2, R_r^2))$ respectively, are computable in polynomial time.

- Scheme (S, R) is ***public coin*** if all messages sent by R to S are independent random coins.

REMARK 3.4.2

We make several remarks regarding Definition 3.4.1.

1. We generally consider schemes that have the same message length for both phases. When this is the case, namely $k = k_1 = k_2$, we say our 2-phase commitment scheme has message length k. It is only in Section 3.5.5 that we will use this feature of different message lengths.

2. Instead of providing sender S with decommitment values as private outputs of the commitment phases, we simply provide it with the same coin tosses r_S throughout (so it can recompute any private state from the transcripts of the previous phases). The receiver R, however, operates using independent coin tosses in each phase as it does not need to keep private states.

3. The 2-phase commitment schemes that we construct will be public coin scheme where the receiver R strategy is just to send random coins in each round.

Hiding for 2-phase commitment schemes. As for standard commitment schemes, we define the security of the sender in terms of a hiding property. Stated informally, the hiding property for a 2-phase commitment scheme says that *both* commitment phases are hiding. Note that since the phases are run sequentially, the hiding property for the second commitment stage is required to hold even given the receiver's view of the first stage.

DEFINITION 3.4.3

2-phase commitment scheme (S, R), with security parameter n and message lengths $(k_1(n), k_2(n))$, is ***statistically hiding*** if for all adversarial receiver R^*,

1. The views of R^* when interacting with the sender in the first phase on any two messages are statistically indistinguishable. Namely, for all $\sigma^{(1)}, \widetilde{\sigma}^{(1)} \in \{0,1\}^{k_1}$, the probability ensembles $\left\{\text{view}_{R^*}(S_c^1(\sigma^{(1)}), R^*)(1^n)\right\}_{n\in\mathbb{N}}$ and $\left\{\text{view}_{R^*}(S_c^1(\widetilde{\sigma}^{(1)}), R^*)(1^n)\right\}_{n\in\mathbb{N}}$ are statistically indistinguishable.

2. The views of R^* when interacting with the sender in the second phase are statistically indistinguishable no matter what the sender committed to in the first phase. Namely, for all $\sigma^{(1)} \in \{0,1\}^{k_1}$, and all $\sigma^{(2)}, \widetilde{\sigma}^{(2)} \in \{0,1\}^{k_2}$, the probability ensembles $\left\{\text{view}_{R^*}(S_c^2(\sigma^{(2)}), R^*)(\mathrm{T}, 1^n)\right\}_{n\in\mathbb{N}}$ and $\left\{\text{view}_{R^*}(S_c^2(\widetilde{\sigma}^{(2)}), R^*)(\mathrm{T}, 1^n)\right\}_{n\in\mathbb{N}}$, where $\mathrm{T} = \text{transcript}(S^1(\sigma^{(1)}), R^*)(1^n)$, are statistically indistinguishable.

We stress that the second condition of the above hiding definition (Definition 3.4.3) requires that the view of receiver in the second phase be indistinguishable for any two messages even given the transcript of the first phase, $\mathrm{T} = \text{transcript}(S^1(\sigma^{(1)}), R^*)(1^n)$.

1-out-of-2 binding for 2-phase commitment schemes. The 1-out-of-2 binding property, informally stated, says that *at least* one of the two commitment phases is binding. In other words, for every (nonuniform PPT) malicious sender S^*, at most one of the two phases is bad in that S^* can decommit a given commitment to two different messages in that phase. We allow this bad phase to be determined dynamically by S^*. Moreover, we require that the second phase be *statistically* binding if the sender breaks the first phase. Our construction achieves this stronger property, and using it simplifies some of our proofs.

DEFINITION 3.4.4

2-phase commitment scheme (S, R), with security parameter n and message lengths $(k_1(n), k_2(n))$, is ***statistically [resp., computationally] 1-out-of-2 binding*** if there exist a set $\mathcal{B}$ of first phase transcripts and a negligible function $\varepsilon(n)$ such that:

1. For all [resp., nonuniform PPT] adversary S^*, S^* succeeds in the following game with probability at most $\varepsilon(n)$ for all sufficiently large n:[10]

   (a) S^* and R_c^1 interact and output a first-phase commitment $c^{(1)}$.

   (b) S^* outputs two full transcripts $\lambda = (\tau, \kappa)$ and $\widetilde{\lambda} = (\widetilde{\tau}, \widetilde{\kappa})$ of *both* phases with the following three properties:

   - Transcripts λ and $\widetilde{\lambda}$ both start with prefix $c^{(1)}$.
   - Transcript λ contains a successful opening of $c^{(1)}$ to the value $\sigma^{(1)} \in \{0,1\}^{k_1}$ using a first-phase transcript τ not in $\mathcal{B}$, and R_r^1 and R_r^2 both accept in λ.
   - Transcript $\widetilde{\lambda}$ contains a successful opening of $c^{(1)}$ to the value $\widetilde{\sigma}^{(1)} \in \{0,1\}^{k_1}$ using a first-phase transcript $\widetilde{\tau}$ not in $\mathcal{B}$, and R_r^1 and R_r^2 both accept in $\widetilde{\lambda}$.

   (c) S^* succeeds if all of the above conditions hold and $\sigma^{(1)} \neq \widetilde{\sigma}^{(1)}$.

2. For every (even computationally unbounded) sender S^*, the first-phase transcripts in $\mathcal{B}$ make the second phase statistically binding. In other words, for all S^* and all $\tau \in \mathcal{B}$, with probability at least $1 - \varepsilon(n)$ over $c^{(2)} = (S^*, R_c^2)(\tau)$, there is at most one value $\sigma^{(2)} \in \{0,1\}^{k_2}$ such that $\text{output}_R(S^*, R_r^2)(c^{(2)}, \sigma^{(2)}) = \texttt{accept}$.

REMARK 3.4.5

For computationally 1-out-of-2 binding schemes, we require that Condition 1 holds against (nonuniform) PPT adversaries, but Condition 2 must hold against all, computationally-unbounded adversaries. For statistically 1-out-of-2 binding schemes, we require that both Conditions 1 and 2 hold against computationally-unbounded adversaries.

[10] Definitions of cryptographic primitives in the literature often use the reverse order of quantifiers, asking that for every (nonuniform) PPT adversary S^*, there exists a negligible function $\varepsilon(n)$ such that the success probability of S^* is at most $\varepsilon(n)$. The two resulting definitions, however, turn out to be equivalent [Bel].

3.4.2 Our 2-phase commitment scheme

We now describe our 2-phase commitment scheme for general functions $f\colon \{0,1\}^n \to \{0,1\}^n$, not necessarily regular nor one-way—as we shall later see, it is the regularity condition that gives the hiding property, and the one-wayness of the function that gives the binding property of our scheme. Let $\mathcal{H} = \{h\colon \{0,1\}^n \to \{0,1\}^m\}$ be a family of pairwise-independent hash functions. As shown in Section 3.3.1, we have a family whose description of each element takes $\ell(n,m) = 2n$ bits. It will be convenient to make $\ell(n,m)+m = q(n)$, for some fixed polynomial $q(n)$, so that for every $y \in \{0,1\}^n$, $|h, h(y)| = q(n)$. This can be done by padding random bits to the description of h.

In addition, it will be convenient to work with protocols where the sender has no input $\sigma^{(j)}$ to be committed to, but rather privately receives an output $d^{(j)}$ at the end of each phase $j \in \{1,2\}$ of the commitment. If we can ensure that $d^{(j)}$ is close to uniform given the receiver's view, such a protocol can be easily tuned into a commitment scheme: the sender can commit to an $\sigma^{(j)}$ of its choice by sending $d^{(j)} \oplus \sigma^{(j)}$ at the end of the commit stage.

PROTOCOL 3.4.6 . . . .

2-phase commitment scheme (S,R) based on $f\colon \{0,1\}^n \to \{0,1\}^n$.

Parameters: Integers $t \in \{1,2,\ldots,n\}$, $k_1 = k_2 = k \in \{1,2,\ldots,n\}$, $\Delta_1 \in \{0,1,\ldots,t\}$, and $\Delta_2 \in \{0,1,\ldots,n-t\}$.

Sender's private input: String $x \in \{0,1\}^n$. (Note that this is not the value to which the sender is committing, but is rather part of its coins, which will be chosen uniformly at random by S unless otherwise specified.)

First phase commit:

1. S_c^1 sets $y = f(x)$.
2. Let $\mathcal{H}_1 = \{h_1\colon \{0,1\}^n \to \{0,1\}^{t-\Delta_1}\}$ be a family of pairwise-independent hash functions. S_c^1 chooses a random hash $h_1 \leftarrow \mathcal{H}_1$, and computes $v = (h_1, h_1(y)) \in \{0,1\}^q$.
3. (S_c^1, R_c^1) run the interactive hashing protocol $(S_{\mathrm{IH}}(v), R_{\mathrm{IH}})(1^q, 1^k)$, given by Protocol 3.2.3, with S_c^1 and R_c^1 acting as S_{IH} and R_{IH} respectively.

   Let circuit $C^{(1)}\colon \{0,1\}^k \to \{0,1\}^q$ be the common output and $d^{(1)} \in \{0,1\}^k$ be S_{IH}'s private output in $(S_{\mathrm{IH}}(v), R_{\mathrm{IH}})(1^q, 1^k)$.

First phase sender's private output: String $d^{(1)} \in \{0,1\}^k$.

First phase reveal:

S_r^1 sends the tuple $\gamma^{(1)} = (d^{(1)}, y, h_1)$.

Receiver R_r^1 accepts if and only if $C^{(1)}(d^{(1)}) = (h_1, h_1(y))$.

Second phase commit:

Second phase common input: First-phase transcript $\tau = \text{transcript}(S^1(x), R^1)$, which in particular includes the string y.

1. Let $\mathcal{H}_2 = \{h_2 \colon \{0,1\}^n \to \{0,1\}^{n-t-\Delta_2}\}$ be a family of pairwise-independent hash functions. S_c^2 chooses a random hash $h_2 \leftarrow \mathcal{H}_2$, and computes $w = (h_2, h_2(x)) \in \{0,1\}^q$.
2. (S_c^2, R_c^2) run the interactive hashing protocol $(S_{\mathrm{IH}}(w), R_{\mathrm{IH}})(1^q, 1^k)$, given by Protocol 3.2.3, with S_c^2 and R_c^2 acting as S_{IH} and R_{IH} respectively.

   Let circuit $C^{(2)} \colon \{0,1\}^k \to \{0,1\}^q$ be the common output and $d^{(2)} \in \{0,1\}^k$ be S_{IH}'s private output in $(S_{\mathrm{IH}}(v), R_{\mathrm{IH}})(1^q, 1^k)$.

Second phase sender's private output: String $d^{(2)} \in \{0,1\}^k$.

Second phase reveal:

S_r^2 sends the tuple $\gamma^{(2)} = (d^{(2)}, x, h_2)$.

Receiver R_r^2 accepts if and only if $f(x) = y$ and $C^{(2)}(d^{(2)}) = (h_2, h_2(x))$.

THEOREM 3.4.7

If f is a regular one-way functions with known security $s(n) = n^{\omega(1)}$, then Protocol 3.4.6, with setting of parameters $t = \mathrm{H}(f(U_n))$, $k = O(\log n)$, and $\Delta_1 = \Delta_2 = \frac{1}{4}\log s$, is a 2-phase commitment scheme that is statistically hiding and computationally 1-out-of-2 binding.

Because we do not know how to efficiently compute the correct value of $t = \mathrm{H}(f(U_n))$, we are forced to try out all values of $t = 1, 2, \ldots, n$ to get a collection of commitment schemes, as stated in the next corollary. While having a collection of schemes instead of a single scheme may seem disconcerting, it is possible to convert this collection of 2-phase commitments into a *single* commitment scheme that is statistically hiding and computationally binding (in the standard sense of binding); we show how to do this in Section 3.5.5 using the Haitner & Reingold transformation [HR2].

COROLLARY 3.4.8

If regular one-way functions with known security $s(n) = n^{\omega(1)}$ exist, then on security parameter 1^n, we can construct in time polynomial in n a collection of public-coin 2-phase commitment schemes $\mathcal{COM} = \{\mathsf{Com}_1, \cdots, \mathsf{Com}_n\}$, such that:

- there exists an index $i \in \{1, 2, \ldots, n\}$ such that scheme Com_i is statistically hiding, and
- for every index $i \in \{1, 2, \ldots, n\}$, scheme Com_i is computationally 1-out-of-2 binding.

We divide the proof of Theorem 3.4.7 into Lemma 3.4.9 and Lemma 3.4.10 that establish the statistical hiding and computational 1-out-of-2 binding properties of Protocol 3.4.6, respectively.

LEMMA 3.4.9

If f is a regular function, then Protocol 3.4.6, with setting of parameters $t = \mathrm{H}(f(U_n))$, $k < q(n)$, and $\Delta_1 = \Delta_2 = \omega(\log n)$, is statistically hiding in the sense of Definition 3.4.3.

Proof. Since $t = \mathrm{H}(f(U_n))$, the Leftover Hash Lemma (Lemma 3.3.1) tells us that random variable $Z = (H_1, H_1(f(U_n)))$ is $2^{-\Omega(\Delta_1)}$-close to the uniform. Then by the hiding property of interactive hashing (Definition 3.2.1), the first commitment phase is $2^{-\Omega(\Delta_1)}$-hiding, which in turn is statistically hiding since $\Delta_1 = \omega(\log n)$.

Let τ be the transcript of the first phase and y the string sent in the first reveal phase. Let random variable X represent selecting at random a string from the set $f^{-1}(y)$. Since X is a flat source with entropy $n - \mathrm{H}(f(U_n)) = n - t$, and h_2 maps to strings of length $n-t-\Delta_2$, we apply the Leftover Hash Lemma once more to conclude that random variable $W = (H_2, H_2(X))$ is $2^{-\Omega(\Delta_2)}$-close to the uniform, even given τ. By the hiding property of interactive hashing, the second commitment phase is $2^{-\Omega(\Delta_2)}$-hiding, which in turn is statistically hiding since $\Delta_2 = \omega(\log n)$. □

LEMMA 3.4.10

If f is a $s(n)$-secure one-way function (not necessarily regular), then for any value of $t \in \{1, 2, \ldots, n\}$, Protocol 3.4.6, with setting of parameters $k = O(\log n)$, $\Delta_1 = \Delta_2 \leq \frac{1}{4}\log(s(n))$, is computationally 1-out-of-2 binding in the sense of Definition 3.4.4.

The proof of Lemma 3.4.10 will be broken into Claim 3.4.11 and 3.4.12 that establish the binding property for the first and second phase, respectively. Before stating the claims, we define the binding set $\mathcal{B}$ as follows:

> For every $t \in \{1, 2, \ldots, n\}$, define the set of *light* strings to be $L_t = \{y \in \{0,1\}^n : \Pr[f(U_n) = y] \leq 2^{-t-\Delta_3}\}$, for a parameter Δ_3 that we will set at the end of the proof. Define the binding set $\mathcal{B}$ to be the set of transcripts where the sender reveals $y \in L_t$.

CLAIM 3.4.11

For the binding set $\mathcal{B}$ defined above, if there exists a (nonuniform) PPT S^* that succeeds with probability ε in the game in Condition 1 of Definition 3.4.4, then there exists a nonuniform PPT B that can invert f with success probability at least

$$\varepsilon^{O(1)} \cdot 1/\operatorname{poly}(n) \cdot 2^{-(k+\Delta_1+\Delta_3)}$$

Proof. We define a relation W as follows:

$$W = \{(v, x) : \exists h_1 \text{ such that both } v = (h_1, h_1(f(x))) \text{ and } f(x) \notin L_t\} \quad .$$

Suppose we have a PPT S^* that succeeds with probability greater than ε in the game of in Condition 1 of Definition 3.4.4. In particular, this means that S^* after interacting with R_{IH} will, with probability greater than ε, produce pairs $(d_0^{(1)}, x_0)$ and $(d_1^{(1)}, x_1)$ such that $d_0^{(1)} \neq d_1^{(1)}$, $(C^{(1)}(d_0), x_0) \in W$, and $(C^{(1)}(d_1), x_1) \in W$. By the binding property of interactive hashing (Condition 3 of Definition 3.2.1), there exists a nonuniform PPT A such that

$$\Pr_{v \leftarrow \{0,1\}^q}[A(v) \in W_v] > 2^{-k} \left(\frac{\varepsilon}{q}\right)^{O(1)} \quad . \tag{3.1}$$

Without loss of generality, we may assume A to be deterministic since it can nonuniformly fix the random coins that maximize its success probability in (3.1). Hence, the probability in (3.1) is just taken over a random $v \leftarrow \{0,1\}^q$.

Let $v = (h_1, \eta)$ for some $h_1 \in \mathcal{H}_1$ and $\eta \in \{0,1\}^{t-\Delta_1}$, and let $x = A(v) = A(h_1, \eta)$. If $x \in W_{(h_1,\eta)}$, then it is the case that $\eta = h_1(f(x))$ and

$$\Pr[h_1(f(U_n)) = \eta] \geq \Pr[f(U_n) = f(x)] > 2^{-t-\Delta_3} \quad , \tag{3.2}$$

with the last inequality following from $f(x) \notin L_t$.

Consider an algorithm B that on input y, picks a random hash function $h_1 \leftarrow \mathcal{H}_1$, and outputs $A(h_1, h_1(y))$. Observe that B successfully finds a preimage of y if $A(h_1, h_1(y)) \in W_{(h_1,h_1(y))}$. We let $\eta = h_1(y)$, and compute the probability that B inverts f as follows:

$$\begin{aligned}
&\Pr[B(f(U_n)) \in f^{-1}(f(U_n))] \\
&\quad = \mathop{\mathrm{E}}_{h_1 \leftarrow \mathcal{H}_1}\left[\sum_{\eta \text{ s.t. } A(h_1,\eta) \in W_{(h_1,\eta)}} \Pr[h_1(f(U_n)) = \eta]\right] \\
&\quad > \mathop{\mathrm{E}}_{h_1 \leftarrow \mathcal{H}_1}\left[\sum_{\eta \text{ s.t. } A(h_1,\eta) \in W_{(h_1,\eta)}} 2^{-t-\Delta_3}\right] && \text{(by 3.2)} \\
&\quad = 2^{-t-\Delta_3} \cdot 2^{t-\Delta_1} \cdot \Pr_{(h_1,\eta) \leftarrow \mathcal{H}_1 \times \{0,1\}^{t-\Delta_1}}[A(h_1, \eta) \in W_{(h_1,\eta)}] \\
&\quad > 2^{-(\Delta_1+\Delta_3)} \cdot 2^{-k} \left(\frac{\varepsilon}{q}\right)^{O(1)} && \text{(by 3.1)} \\
&\quad = \varepsilon^{O(1)} \cdot 1/\operatorname{poly}(n) \cdot 2^{-(k+\Delta_1+\Delta_3)} && \text{(since } q = \operatorname{poly}(n)) \quad . \qquad \square
\end{aligned}$$

CLAIM 3.4.12

For the binding set $\mathcal{B}$ defined above, Condition 2 of Definition 3.4.4 is satisfied with $\varepsilon = \operatorname{poly}(n) \cdot 2^{-\Omega(\Delta_3-\Delta_2)}$.

Proof. Let $y \in L_t$ be the string sent in the first reveal phase. This means that $\Pr[f(U_n) = y] \leq 2^{-t-\Delta_3}$, or equivalently $\left|f^{-1}(y)\right| \leq 2^{n-t-\Delta_3}$. Define set $\Gamma = \{(h_2, h_2(x)) : h_2 \in \mathcal{H}_2, x \in f^{-1}(y)\}$, and let $\mu(\Gamma)$ denote the density of the subset Γ. Since h_2 maps $\{0,1\}^n$ to $\{0,1\}^{n-t-\Delta_2}$, we have

$$\mu(\Gamma) \leq \frac{\left|f^{-1}(y)\right|}{2^{n-t-\Delta_2}} \leq \frac{2^{n-t-\Delta_3}}{2^{n-t-\Delta_2}} = 2^{(\Delta_2-\Delta_3)} \ .$$

Applying Lemma 3.2.5, we have

$$\Pr\left[(w_0, w_1) = \text{output}(S^*, R_{\text{IH}}) \text{ satisfies } \ w_0, w_1 \in \Gamma\right] < 2^{-\Omega(\Delta_3-\Delta_2)} \cdot \text{poly}(q) \ ,$$

which then concludes our proof since q is a fixed polynomial in n. □

Proof of Lemma 3.4.10. Set $\Delta_3 = \frac{1}{2}\log s(n)$, and we are given that $k = O(\log n)$, and $\Delta_1 = \Delta_2 \leq \frac{1}{4}\log(s(n))$. With this setting, Claim 3.4.12 shows that Condition 2 in Definition 3.4.4 is satisfied with $\varepsilon(n) = \text{poly}(n) \cdot 2^{-\Omega(\log s(n))} = \text{neg}(n)$, since $s(n) = n^{\omega(1)}$. Condition 1 of Definition 3.4.4 is also satisfied with negligible probability $\varepsilon(n)$ because otherwise f can be inverted with probability

$$\begin{aligned}\varepsilon^{O(1)} \cdot 1/\text{poly}(n) \cdot 2^{-(k+\Delta_1+\Delta_3)} &\geq \varepsilon^{O(1)} \cdot 1/\text{poly}(n) \cdot 2^{-(O(\log n)+(3/4)\cdot(\log s(n)))} \\ &= \varepsilon^{O(1)} \cdot 1/\text{poly}(n) \cdot s(n)^{-3/4} \ ,\end{aligned}$$

which is greater than $1/s(n)$ if ε is nonnegligible. □

3.5 From Any One-Way Function

Our final hurdle is to remove the regularity assumption. It turns out that this is the most technically challenging step. Similar to our construction from regular one-way functions (with unknown preimage size) in Section 3.4, our construction based on any one-way function yields a collection 2-phase commitments, as stated below.

THEOREM 3.5.1

If one-way functions exist, then on security parameter 1^n, we can construct in time polynomial in n a collection of public-coin 2-phase commitment schemes $\mathcal{COM} = \{\mathsf{Com}_1, \cdots, \mathsf{Com}_m\}$, where $m = \text{poly}(n)$, such that:

- there exists an index $i \in \{1, 2, \ldots, m\}$ such that scheme Com_i is statistically hiding, and
- for every index $i \in \{1, 2, \ldots, m\}$, scheme Com_i is computationally 1-out-of-2 binding.

The above collection of 2-phase commitment schemes suffices for obtaining statistical zero-knowledge arguments for all of NP (cf., [Ngu, Chap. 6]). Hence, Theorem 3.5.1 suffices to establish Theorem 1.2.5, which states that statistical zero-knowledge arguments for all of NP can be based on any one-way function.

We prove Theorem 3.5.1 in Sections 3.5.1 through 3.5.3. In Section 3.5.5, we present a transformation technique, due to Haitner and Reingold [HR2], that takes the above collection of 2-phase commitment schemes and converts it into a *single* commitment scheme that is statistically hiding and computationally binding (in the standard sense of binding). Doing so would establish Theorem 3.0.4, one of the main theorems of this present chapter, and would also provide an alternative way to establish Theorem 1.2.5 (since by Corollary 2.5.3, statistically-hiding commitments imply statistical zero-knowledge arguments for all of NP).

3.5.1 Overview

We now present an overview of how we generalize our construction for regular one-way functions with unknown preimage size (Protocol 3.4.6) to arbitrary one-way functions. As shown in Lemma 3.4.10, this protocol already achieves 1-out-of-2 binding when f is any one-way function (for every value of t). Thus the challenge is the hiding property. (Another issue is that Protocol 3.4.6 requires a one-way function with known security. It turns out that our method for handling the hiding property also eliminates the need to know the security.)

As discussed in Section 3.4, for regular one-way functions with unknown preimage size, Protocol 3.4.6 has a hiding first phase when the parameter t satisfies $t \lesssim \mathrm{H}(f(U_n))$ and has a hiding second phase when t satisfies $t \gtrsim \mathrm{H}(f(U_n))$. Intuitively, when f is not regular, we should replace the fixed value $\mathrm{H}(f(U_n))$ with the dynamic value $\mathrm{H}_f(y) \stackrel{\text{def}}{=} \log(1/\Pr[f(U_n) = y])$, where $y = f(x)$ is the value chosen by the sender in the pre-processing step, because $\mathrm{H}_f(y)$ can be viewed as measuring the amount of *entropy* in y. The *approximable preimage-size one-way functions* studied by Haitner et al. [HHK$^+$] come equipped with an algorithm that estimates $\mathrm{H}_f(y)$, but for general one-way functions, this quantity may be infeasible to compute.

A weakly-hiding scheme (details in Section 3.5.2). One natural approach is to have the sender choose t at random and hope that it is close to $\mathrm{H}_f(y)$. When we choose t too small, only the first phase will be hiding, and when we choose t too large, only the second phase will be hiding. But we have a nonnegligible probability of $\delta = 1/n$ that $t \approx \mathrm{H}_f(y)$, and thus both phases will be hiding. Thus we have a 1-out-of-2-binding commitment scheme satisfying a ***weak hiding*** property, where with probability $\delta = 1/n$, both phases are hiding, and it is always the case that at least one phase is hiding. Actually, in order to simplify our analysis, we will include t as a parameter to the protocol. Then there exists a choice of t such that the protocol is weakly hiding in the sense above, and for all choices of t the protocol is 1-out-of-2 binding. At the end, we will enumerate over all values of t, resulting in a *collection* of commitment schemes as claimed in Theorem 3.5.1, albeit with a weak hiding property.

Unfortunately, we do not know how to directly construct zero-knowledge arguments from weakly-hiding 1-out-of-2-binding commitments. Thus instead, much of the effort in this paper is devoted to amplifying the weak hiding property, where $\delta = 1/n$, into a ***strong hiding*** property, where $\delta = 1 - \text{neg}(n)$, while maintaining the 1-out-of-2 binding property.

Amplifying the hiding property (details in Section 3.5.3). We do not amplify the hiding probability from $\delta = 1/n$ to $\delta = 1 - \text{neg}(n)$ in one shot, but instead perform a sequence of $\log n$ iterations, each one of which increases δ by a roughly factor of 2. This results in $\delta = \Omega(1)$, and then we are able to get $\delta = 1 - \text{neg}(n)$ in just one more amplification step.

How do we double δ? A natural idea is to consider several executions of the previous weakly-hiding scheme. Specifically, if we take $m = O(1)$ executions, the probability that at least one of the executions has both phases hiding is roughly $m \cdot \delta$. Moreover, each of the remaining $m - 1$ executions have either the first phase hiding or the second phase hiding. Thus for some value of β, there are $\beta + 1$ first phases that are hiding and $m - \beta$ second phases that are hiding. It turns out that we can choose β so that this exact $(\beta + 1, m - \beta)$ breakdown given that one execution has both phases hiding occurs with probability $\Omega(1/\sqrt{m})$. Thus we are in the situation described with probability $m \cdot \delta \cdot \Omega(1/\sqrt{m}) = \Omega(\sqrt{m} \cdot \delta) > 2\delta$, for a large enough constant m.

Now our aim is to combine the outcomes of the weakly-hiding schemes in such a way that when the above-described situation occurs, which happens with probability at least 2δ, both phases are hiding. Notice that the secret values $\sigma_1, \ldots, \sigma_m \in \{0,1\}^k$ to which the sender commits in the first commit phases have entropy (even min-entropy) at least $(\beta+1) \cdot k$ conditioned on the receiver's view (because $(\beta + 1)$ of them are hiding), and similarly the sender's secrets in the second commit phases have entropy at least $(m - \beta) \cdot k$ conditioned on the receiver's view. Let us compare this to the situation with binding. Since each execution is 1-out-of-2 binding, a cheating polynomial-time sender can break the binding property for either at most β of the first phases or at most $m - \beta - 1$ of the second phases. Thus the number of possible values to which the sender can open in each case is at most $2^m \cdot 2^{k \cdot \beta}$ in the first phase or at most $2^{k \cdot (m - \beta - 1)}$, where the 2^m factor in the first bound comes from the sender's ability to choose which subset of executions to break (and it is this factor that limits us to taking m to be a constant). We can view these as strong forms of entropy upper bounds $m + k\beta$ and $k \cdot (m - \beta - 1)$. In at least one phase, there will be an *entropy gap* of at least $k - m$.

How can we exploit these entropy gaps? It turns out that interactive hashing, again, is a useful tool. Specifically, in the first phase we have the sender choose a random pairwise-independent hash function h_1 mapping to approximately $(\beta + 1) \cdot k$ bits and use $(h_1, h_1(\sigma_1, \ldots, \sigma_m))$ as the input to the interactive hashing protocol, and analogously for the second phase. By the Leftover Hash Lemma, this pairwise-independent hashing converts the min-entropy lower bound described above to an almost-uniform distribution, so the interactive hashing hiding property applies. As for the binding property, the bound on the number of the sender's choices gets translated to saying that the sender's input (in the first phase) comes from a set Γ of density $2^{-(k-m)}$, so the interactive hashing binding property applies. The analyses for the second phase are similar. Formalizing these ideas, we get a new 1-out-of-2-binding commitment scheme in which both phases are hiding with probability at least 2δ.

When we try to iterate this amplification step $O(\log n)$ times, we run into a new difficulty. Specifically, the above sketch hides the fact that we pay entropy losses of $\omega(\log n)$ in both the hiding and binding analyses. The entropy loss of $\omega(\log n)$ in the hiding property comes from the Leftover Hash Lemma, in order to ensure that $(h_1, h_1(\sigma_1, \ldots, \sigma_m))$ has negligible statistical distance from uniform. The entropy loss of $\omega(\log n)$ in the binding property comes from needing the $\mu(\Gamma) \cdot 2^k$ factor to be negligible when applying Lemma 3.2.5. This forces us to go, in one step of amplification, from a commitment scheme for secrets of length k to a scheme for secrets of length $k - m - \omega(\log n)$. As in Lemma 3.4.10, we can take the initial secret length to be $k = \Theta(\log s(n)) = \omega(\log(n))$ if the one-way function has known security $s(n) = n^{\omega(1)}$. But to tolerate $\log n$ losses of $\omega(\log n)$, we would need $s(n) = n^{\omega(\log n)}$ (i.e., at least quasipolynomial security).

To get around this difficulty, in the amplification, we work with more relaxed, average-case measures of entropy for both the hiding and binding analyses. Specifically, for hiding, we keep track of the expected collision probability of the sender's secret, conditioned on the receiver's view. (Actually, we use the expected square root of the collision probability.) For binding, we work with the expected number of values to which the sender can open. In both cases, we only require these expectations to be within a constant factor of the ideal values, which are 2^{-k} and 1 respectively. With these measures, it turns out that we need only lose $O(m) = O(1)$ bits in the entropy gap with each amplification step. Moreover, once we amplify δ to a constant, we can afford to take the number of executions m to equal the security parameter n and get an $\Omega(n)$-bit entropy gap in the final amplification step. This allows us to achieve exponentially strong statistical hiding even when we do not know the security and start with secret length of $k = O(\log n)$.

The hiding analysis of the above construction works only for certain values of t in the weakly-hiding scheme, and for certain values of the β's in the amplification steps. We try out all possible values of t and β's, thus obtaining a collection of poly(n) schemes, at least one of which is strongly hiding and all of which are 1-out-of-2 binding. Notice that the number of possible choices of t and the β's are polynomial in n since $t \in \{1, 2, \ldots, n\}$, the β's in the each step except for the last is in the range $\{0, 1, \ldots, m-1\}$, for some constant m, and the last β is in the range $\{0, 1, \ldots, n\}$.

Converting 1-out-of-2-binding commitments to standard commitments (details in Section 3.5.5). Having obtained this collection of poly(n) schemes, we use convert it into a single commitment scheme that is statistically hiding and computationally binding using a transformation provided by Haitner and Reingold [HR2], henceforth called the Haitner & Reingold transformation.

3.5.2 Weakly-hiding and 1-out-of-2-binding commitments

As discussed in Section 3.4, for the case of regular one-way functions with unknown preimage size, Protocol 3.4.6 has a hiding first phase when the parameter t satisfies $t \lesssim \mathrm{H}(f(U_n))$ and has a hiding second phase when t satisfies $t \gtrsim \mathrm{H}(f(U_n))$. When f is not regular, then there will be one value of $t \in \{1, 2, \ldots, n\}$ such that $\mathrm{H}(f(U_n)) \approx t$ with probability $1/n$. This is the case because there are only n possible choices for the value of t.

With this observation in mind, our 2-phase commitment scheme from general one-way functions will be the same as the scheme in Protocol 3.4.6, with setting of parameters $t = t_0$, $k = O(\log n)$, and $\Delta_1 = \Delta_2 = 2\log n$, for some $t_0 \in \{1, 2, \ldots, n\}$. In other words, the same scheme—with slightly different setting of parameters—used in the case of regular one-way functions is also applicable to general one-way functions.

This commitment scheme (using general one-way functions), as we will show, is computationally 1-out-of-2 binding, but only statistically hiding in both phases with probability at least $1/n$ (hence, called ***weakly hiding***). In order to obtain a tighter analysis when we amplify this scheme, we depart from the standard measures of hiding and binding used in Section 3.4. Instead, we measure the statistical hiding property of our 2-phase commitments using the *expected square root of the collision probability* of the sender's secret, denoted as $\mathrm{CP}^{1/2}$, and defined in Section 3.5.2. We measure the binding property by analyzing the *expected* number of values to which an adversarial sender can open.

Later in Section 3.5.3, we show how to boost the statistical hiding probability to $1 - 2^{-\Omega(n)}$ while maintaining the computational 1-out-of-2 binding property.

Properties of collision probability

DEFINITION 3.5.2

For any random variable A, we define its ***collision probability*** as the probability that two independent samples from A are equal. In other words,

$$\mathrm{CP}(A) \overset{\mathrm{def}}{=} \sum_{a \in \mathrm{Supp}(A)} (\Pr[A = a])^2 = \mathop{\mathrm{E}}_{a \leftarrow A} [\Pr[A = a]] \quad .$$

Measuring the collision probability of a random variable is equivalent to measuring its ***Renyi entropy of order 2***, defined as

$$\mathrm{H}_2(A) = \log \frac{1}{\mathrm{E}_{a \leftarrow A}[\Pr[A = a]]} = \log \frac{1}{\mathrm{CP}(A)} \quad .$$

DEFINITION 3.5.3

For any random variable A, we define its ***expected square root of the collision probability*** as

$$\mathrm{CP}^{1/2}(A) \stackrel{\mathrm{def}}{=} \sqrt{\mathrm{CP}(A)} \ .$$

For any two (possibly correlated) random variables A and B, we define

$$\mathrm{CP}^{1/2}(A|B) \stackrel{\mathrm{def}}{=} \mathop{\mathrm{E}}_{b \leftarrow B}\left[\mathrm{CP}^{1/2}(A|_{B=b})\right] \ .$$

We think of $\mathrm{CP}^{1/2}(A|B) \leq \sqrt{2^k}$ as saying that A has ***conditional Renyi entropy*** of at least k given B. We use the expected *square root* of the collision probability (as our measure of hiding) instead of just expected collision probability in order to ensure that conditioning on a random variable Z can only decrease the conditional Renyi entropy by at most $\log(|\mathrm{Supp}(Z)|)$ bits. (See Lemma 3.5.7 below for details.)

The following lemmas show that $\mathrm{CP}^{1/2}$ behaves nicely as an entropy measure. Proofs are in Appendix A.2.

LEMMA 3.5.4

For independent pairs of random variables $(X_1, Y_1), \ldots, (X_m, Y_m)$,

$$\mathrm{CP}^{1/2}((X_1, \ldots, X_m)|(Y_1, \ldots, Y_m)) = \prod_{i=1}^{m} \mathrm{CP}^{1/2}(X_i|Y_i) \ .$$

Note that X_i and Y_i can be correlated, it is only required that the pair (X_i, Y_i) be independent from the other tuples.

In terms of conditional Renyi entropy, Lemma 3.5.4 states that the entropy is additive for independent random variables. We will actually need a generalization of Lemma 3.5.4 to deal with somewhat dependent random variables, as stated in the next lemma.

LEMMA 3.5.5

Suppose random variables $(X_1, Y_1), \ldots, (X_m, Y_m)$ satisfy the following conditions for some values of $\alpha_1, \ldots, \alpha_m \in \mathbb{R}^+$ and all $i = 1, 2, \ldots, m$:

1. For every $(y_1, \ldots, y_{i-1}) \in \mathrm{Supp}(Y_1, Y_2, \ldots, Y_{i-1})$,

$$\mathrm{CP}^{1/2}(X_i|_{Y_1=y_1,\ldots,Y_{i-1}=y_{i-1}} \mid Y_i|_{Y_1=y_1,\ldots,Y_{i-1}=y_{i-1}}) \leq \alpha_i \ .$$

2. For every $(y_1, \ldots, y_i) \in \mathrm{Supp}(Y_1, Y_2, \ldots, Y_i)$, the $i+1$ random variables $X_1, X_2, \ldots, X_i$, and Y_{i+1} are independent, even if we condition on $Y_1 = y_1, \ldots, Y_i = y_i$.

Then,

$$\mathrm{CP}^{1/2}((X_1,\ldots,X_m)|(Y_1,\ldots,Y_m)) \leq \prod_{i=1}^{m} \alpha_i \ .$$

The next lemma shows that pairwise-independent randomness extraction $(h, h(x))$ preserves the $\mathrm{CP}^{1/2}$ measure.

LEMMA 3.5.6

(Randomness Extraction Lemma.) Let (X, Y) be any (possibly correlated) pair of random variables, and let random variable H denote a random hash function from a family of pairwise-independent hash functions $\mathcal{H}$ with range $\{0,1\}^\alpha$. Suppose the hash functions from $\mathcal{H}$ are represented by $(q-\alpha)$-bit strings and $\mathrm{CP}^{1/2}(X|Y) \leq \sqrt{2^{-(\alpha+3)}}$. If H is independent from (X, Y), then

$$\mathrm{CP}^{1/2}((H, H(X))|Y) \leq \sqrt{2^{-(q-1)}} \ .$$

In other words, if X has at least $\alpha+3$ bits of conditional Renyi entropy given Y, then we can extract α bits from X that have conditional Renyi entropy at least $\alpha - 1$. Notice that this entropy loss is only 4 bits, as compared to $2\log(1/\varepsilon)$ if we require that the output be ε-close to uniform as in the Leftover Hash (Lemma 3.3.1). This constant loss of conditional Renyi entropy allows us to do a tighter hiding analysis in Section 3.5.3.

LEMMA 3.5.7

For any triple of (possibly correlated) random variables X, Y and Z,

$$\mathrm{CP}^{1/2}(X|Y) \leq \mathrm{CP}^{1/2}(X|(Y,Z)) \leq \sqrt{|\mathrm{Supp}(Z)|} \cdot \mathrm{CP}^{1/2}(X|Y) \ .$$

This says that conditioning on random variable Z can only decrease the conditional Renyi entropy, but does so by at most $\log(|\mathrm{Supp}(Z)|)$ bits. The final lemma is a stronger variant of the previous Leftover Hash Lemma of Lemma 3.3.1, with its hypothesis stated in terms of collision probability.

LEMMA 3.5.8

(A stronger variant of the Leftover Hash Lemma [BBR, ILL].) Let random variable H denote a random hash function from a family of pairwise-independent hash functions $\mathcal{H}$ with range $\{0,1\}^\alpha$. For any $\varepsilon > 0$, if $\mathrm{CP}(X) \leq \varepsilon^2 \cdot 2^{-\alpha}$ and H is independent from X, then random variable $(H, H(X))$ is ε-close in statistical distance to uniform.

Average-case hiding and binding properties of interactive hashing

We now analyze the interactive hashing protocol, namely Protocol 3.2.3, in terms of *average-case* measures. For hiding, we use the $\mathrm{CP}^{1/2}$ measure introduced in the previous section. For the binding property, we present an average-case variant of Lemma 3.2.5, where we look at the *expected* number of outputs that lies in any set Γ, rather than bound the probability that there is more than one output in Γ.

LEMMA 3.5.9

(Hiding of interactive hashing in $\mathrm{CP}^{1/2}$ measure.) Let $(S_{\mathrm{IH}}, R_{\mathrm{IH}})$ be the interactive hashing protocol in Protocol 3.2.3. If the sender S_{IH}'s input comes from a random variable Y over $\{0,1\}^q$ and W is any (possibly correlated) random variable (representing the receiver's a priori information about Y), then for any receiver R^*,

$$\mathrm{CP}^{1/2}(Z|(W,V)) \leq \sqrt{2^{q-k}} \cdot \mathrm{CP}^{1/2}(Y|W) \ ,$$

where $Z = \mathrm{output}_{S_{\mathrm{IH}}}(S_{\mathrm{IH}}(Y), R^*)(1^q, 1^k)$ and $V = \mathrm{view}_{R^*}(S_{\mathrm{IH}}(Y), R^*)(1^q, 1^k)$.

Proof. Without loss of generality, we may assume that R^* is deterministic. (The randomized case then follows by taking expectation over R^*'s coins.) Now that since R^* is deterministic, the hash functions sent $h_0, \ldots, h_{q-k-1}$ are fully determined by S_{IH}'s responses $c_0, \ldots, c_{q-k-1} \in \{0,1\}$ (refer to Protocol 3.2.3). Hence, the number of possible different receiver's view is bounded by 2^{q-k}. This implies that $|\mathrm{Supp}(V)| \leq 2^{q-k}$, where $V = \mathrm{view}_{R^*}(S_{\mathrm{IH}}(Y), R^*)(1^q, 1^k)$. By Lemma 3.5.7,

$$\mathrm{CP}^{1/2}(Y|(W,V)) \leq \sqrt{|\mathrm{Supp}(V)|} \cdot \mathrm{CP}^{1/2}(Y|W) \leq \sqrt{2^{q-k}} \cdot \mathrm{CP}^{1/2}(Y|W) \ .$$

Observe that given any particular instantiation of $W = w$ and $V = v$, the distributions $\mathrm{output}_{S_{\mathrm{IH}}}(S_{\mathrm{IH}}(Y), R_{\mathrm{IH}})(1^q, 1^k)|_{W=w,V=v}$ has the same collision probability with $Y|_{W=w,V=v}$ (indeed they are in bijective correspondence). Hence, $\mathrm{CP}^{1/2}(Z|(W,V)) = \mathrm{CP}^{1/2}(Y|(W,V)) \leq \sqrt{2^{q-k}} \cdot \mathrm{CP}^{1/2}(Y|W)$. □

LEMMA 3.5.10

(Binding of interactive hashing in expected measure.) Let $(S_{\mathrm{IH}}, R_{\mathrm{IH}})$ be the interactive hashing protocol in Protocol 3.2.3. For any fixed subset $\Gamma \subseteq \{0,1\}^q$, and for any sender S^*, setting $C = \mathrm{output}((S^*, R_{\mathrm{IH}})(1^q, 1^k))$, we have

$$\mathrm{E}\left[|\{z : C(z) \in \Gamma\}|\right] < \max\{24, 2^{k+1} \cdot \mu(\Gamma)\} \leq 24 + 2^{k+1} \cdot \mu(\Gamma) \ ,$$

where the above expectation is taken over the coins of S^* and R_{IH}.

This lemma and its proof are inspired by the work of Goldriech, Goldwasser, and Linial [GGL], who studied a protocol similar to interactive hashing for a different purpose (namely, random selection protocols).

Proof. Without loss of generality, we may assume that R^* is deterministic. (Else, we can fix its coins to maximize the expectation.) Note that for iteration $j = 0, \ldots, q-k-1$, R_{IH} will send a random h_j, partitioning the set of possible outputs into two sets $\{y : h_j(y) = 0\}$ and $\{y : h_j(y) = 1\}$, and S^* chooses a side of the partition by sending a bit c_j. Let $\Gamma_0 = \Gamma$, and for all $j > 0$, $\Gamma_j = \{y \in \Gamma : h_i(y) = c_i \, \forall i < j\}$ denote the set of compatible elements at iteration j. Let $\mu_j = \mathrm{E}[|\Gamma_j| \cdot 2^{-(q-j)}]$, where the expectation is taken over random choices of $h_0, \ldots, h_{j-1}$. For convenience of notation, assume that the hash function h_i's range is $\{\pm 1\}$, instead of $\{0,1\}$.

Consider a particular set Γ_j, and a particular hash function h_j. Observe that for every $y \neq y' \in \Gamma_j$, $\Pr_{h_j}[h_j(y) = h_j(y')] \leq 1/2$. Hence,

$$\mathop{\mathrm{E}}_{h_j}[h_j(y)h_j(y')] \leq 0 \quad \forall y \neq y' \in \Gamma_j \ . \tag{3.3}$$

Observe that the set $\Gamma_{j+1} = \{y \in \Gamma_j : h_j(y) = c_j\}$. Therefore,

$$\begin{aligned}
\mathop{\mathrm{E}}_{h_j}[\mu(\Gamma_{j+1})] &= \mu(\Gamma_j) + 2^{-(q-j)} \cdot \mathop{\mathrm{E}}_{h_j}\left[\left|\sum_{y \in \Gamma_j} h_j(y)\right|\right] \\
&\leq \mu(\Gamma_j) + 2^{-(q-j)} \cdot \sqrt{\mathop{\mathrm{E}}_{h_j}\left[\left(\sum_{y\in\Gamma_j} h_j(y)\right)^2\right]} && \text{(Cauchy-Schwartz/Jensen)} \\
&= \mu(\Gamma_j) + 2^{-(q-j)} \cdot \sqrt{|\Gamma_j| + \sum_{y \neq y'} \mathop{\mathrm{E}}_{h_j}[h_j(y)h_j(y')]} \\
&\leq \mu(\Gamma_j) + 2^{-(q-j)} \cdot \sqrt{|\Gamma_j|} && \text{(by 3.3)} \\
&= \mu(\Gamma_j) + \sqrt{2^{-(q-j)} \cdot \mu(\Gamma_j)} \ .
\end{aligned}$$

Consequently,

$$\begin{aligned}
\mu_{j+1} &= \mathop{\mathrm{E}}_{h_0,\ldots,h_j}[\mu(\Gamma_{j+1})] \\
&= \mathop{\mathrm{E}}_{h_0,\ldots,h_{j-1}}[\mathop{\mathrm{E}}_{h_j}[\mu(\Gamma_{j+1})]] \\
&\leq \mathop{\mathrm{E}}_{h_0,\ldots,h_{j-1}}\left[\mu(\Gamma_j) + \sqrt{2^{-(q-j)} \cdot \mu(\Gamma_j)}\right] \\
&\leq \mathop{\mathrm{E}}_{h_0,\ldots,h_{j-1}}[\mu(\Gamma_j)] + \sqrt{2^{-(q-j)} \cdot \mathop{\mathrm{E}}_{h_0,\ldots,h_{j-1}}[\mu(\Gamma_j)]} && \text{(Cauchy-Schwartz/Jensen)} \\
&= \mu_j + \sqrt{2^{-(q-j)} \cdot \mu_j} \ .
\end{aligned}$$

Assume that the μ_j's are monotonically increasing (otherwise, we can make it so). This gives us

$$\begin{aligned}
\mu_{q-k} &\leq \mu_0 + \sum_{j=0}^{q-k-1} \sqrt{2^{-(q-j)} \cdot \mu_j} \\
&\leq \mu_0 + \sqrt{\mu_{q-k}} \cdot \sum_{j=0}^{q-k-1} \sqrt{2^{-(q-j)}} && (\mu_j\text{'s are monotonically increasing}) \\
&< \mu_0 + \sqrt{\mu_{q-k}} \cdot \sqrt{6/2^k} \\
&\leq \mu_0 + \frac{\mu_{q-k}}{2} && (\text{if } \mu_{q-k} \geq 24 \cdot 2^{-k}) \ ,
\end{aligned}$$

giving us $\mu_{q-k} < 2\mu_0 = 2\mu(\Gamma)$ if $\mu_{q-k} \geq 24 \cdot 2^{-k}$. This means that μ_{q-k} is either less than $24 \cdot 2^{-k}$ or less than $2\mu(\Gamma)$. Therefore, we can conclude that

$$\begin{aligned}
\mathrm{E}\left[|\{z : C(z) \in \Gamma\}| : C = \mathrm{output}((S^*, R_{\mathrm{IH}})(1^q, 1^k))\right] &= \mu_{q-k} \cdot 2^k \\
&< \max\{24, 2^{k+1} \cdot \mu(\Gamma)\} \ . \qquad \square
\end{aligned}$$

Protocol 3.4.6 is hiding in CP$^{1/2}$ measure

We are now ready to analyze the hiding property of Protocol 3.4.6 in terms of the CP$^{1/2}$ measure. To do so, we say what it means for a scheme to be δ-hiding in CP$^{1/2}$ measure in Definition 3.5.11 below. But before going into that definition, we first establish some notations that are used throughout this part of the section.

With the sender's input being x, we let random variable $\mathrm{view}_{R^*}(S^1_c(x), R^*)$ denote the view of receiver R^* in the first commit phase, let random variable $\mathrm{output}_S(S^1_c(x), R^*)$ denote the sender's private output in the first phase, and let random variable $\mathrm{transcript}(S^1(x), R^*)$ denote the first (commit and reveal) phase transcript.

Using similar notations, with the transcript being τ and sender's input being x, we let random variable $\mathrm{view}_{R^*}(S^2_c(x), R^*)(\tau)$ denote the view of receiver R^* in the second commit phase, let random variable $\mathrm{output}_S(S^2_c(x), R^*)(\tau)$ denote the sender's private output in the second phase, and let random variable $\mathrm{transcript}(S^2(x), R^*)(\tau)$ denote the second (commit and reveal) phase transcript. We write $\mathit{\Gamma}_1$ in $\mathrm{view}_{R^*}(S^1_c(\mathit{\Gamma}_1), R^*)$—and similarly for others—to mean that the sender's private input is chosen uniformly from a set Γ_1.

DEFINITION 3.5.11

For a parameter $\delta \in [0, 1]$, 2-phase commitment scheme (S, R) is said to be ***δ-hiding in* CP$^{1/2}$ *measure*** if there exists two sets $\Gamma_1, \Gamma_2 \subseteq \{0, 1\}^n$ such that the following three properties hold.

(H.1) $\Gamma_1 \cup \Gamma_2 = \{0, 1\}^n$ and $\mu(\Gamma_1 \cap \Gamma_2) \geq \delta$.

(H.2) When the sender's private input x is chosen uniformly from Γ_1, the sender's private output in the first phase has low collision probability given the receiver's view. Formally, for any adversarial receiver R^*,

$$\mathrm{CP}^{1/2}(A|V) \leq \sqrt{2^{-(k-1)}} \ ,$$

for $(A, V) = (\mathrm{output}_S(S_c^1(\mathit{\Gamma}_1), R^*), \mathrm{view}_{R^*}(S_c^1(\mathit{\Gamma}_1), R^*))$.

(H.3) When the sender's private input x is chosen uniformly from Γ_2, the sender's private output in the second phase has low collision probability given the receiver's view. Formally, for every adversarial receiver R^* and every $\tau \in \mathrm{Supp}(\mathrm{T})$, where $\mathrm{T} = \mathrm{transcript}(S^1(\mathit{\Gamma}_2), R^*)$, we have

$$\mathrm{CP}^{1/2}(B_\tau|W_\tau) \leq \sqrt{2^{-(k-1)}} \ ,$$

for $(B_\tau, W_\tau) = (\mathrm{output}_S(S_c^2(\mathit{\Gamma}_2), R^*), \mathrm{view}_{R^*}(S_c^2(\mathit{\Gamma}_2), R^*))|_{\mathrm{T}=\tau}$.

REMARK 3.5.12

Being δ-hiding in $\mathrm{CP}^{1/2}$ measure in the above Definition 3.5.11 roughly means that the scheme is always hiding in at least one phase, and hiding in both phases occurs with probability δ.

LEMMA 3.5.13

(Protocol 3.4.6 is $(1/n)$-hiding in $\mathrm{CP}^{1/2}$ measure.) Let $f \colon \{0,1\}^n \to \{0,1\}^n$ be any function, not necessarily one-way. There exist an integer $t_0 \in \{1, 2, \ldots, n\}$ such that Protocol 3.4.6, with setting of parameters $t = t_0$, $k \leq q(n)$, $\Delta_1 \geq \log n + 4$, and $\Delta_2 \geq 3$, is $(1/n)$-hiding in $\mathrm{CP}^{1/2}$ measure.

Proof. Without loss of generality, we may assume that R^* is deterministic since we can fix the coins of R^* that maximizes the above collision probabilities. We prove that (S, R) satisfies the above three properties of Definition 3.5.11 as follows:

Property (H.1). Define $p(y) = \Pr[f(U_n) = y]$, and let $A_1 = \{y \in \{0,1\}^n : 1/2 \leq p(y) \leq 1\}$, and for $t \in \{2, 3, \ldots, n\}$, let $A_t = \{y \in \{0,1\}^n : 2^{-t} \leq p(y) < 2^{-t+1}\}$. Since $\cup_t A_t = f(\{0,1\}^n)$, there exists an index t_0 such that $\Pr[f(U_n) \in A_{t_0}] \geq 1/n$. Define sets Γ_1 and Γ_2 as follows:

$$\begin{aligned} \Gamma_1 &= \{x : p(f(x)) < 2^{-t_0+1}\} \\ \Gamma_2 &= \{x : p(f(x)) \geq 2^{-t_0}\} \end{aligned}$$

By the definition of Γ_1 and Γ_2, we have that $\mu(\Gamma_1 \cap \Gamma_2) = \Pr[f(U_n) \in A_{t_0}] \geq 1/n$, and also $\Gamma_1 \cup \Gamma_2 = \{0,1\}^n$.

Property (H.2). In the case when the sender's private input $x \in \Gamma_1$, we bound the collision probability of the first phase secret as follows:

$$\begin{aligned}
\mathrm{CP}(f(\mathit{\Gamma}_1)) &= \sum_{y \in f(\Gamma_1)} \left(\frac{p(y)}{\mu(\Gamma_1)}\right)^2 \\
&\leq \left(\max_{y \in f(\Gamma_1)} p(y)\right) \cdot \left(\sum_{y \in f(\Gamma_1)} p(y)\right) \cdot \frac{1}{\mu(\Gamma_1)^2} \\
&< 2^{-t_0+1} \cdot \mu(\Gamma_1) \cdot \mu(\Gamma_1)^{-2} \\
&\leq 2^{-(t_0 - \log n - 1)} \qquad\qquad (\text{since } \mu(\Gamma_1) \geq 1/n) \ .
\end{aligned}$$

Observe that $\mathrm{CP}(f(\mathit{\Gamma}_1)) \leq 2^{-(t_0 - \log n - 1)} \leq 2^{-(t_0 - \Delta_1 + 3)}$. Therefore we can apply Randomness Extraction Lemma 3.5.6 to get $\mathrm{CP}^{1/2}(Q) \leq \sqrt{2^{-(q-1)}}$, where $Q = (H_1, H_1(f(\mathit{\Gamma}_1)))$ and H_1 is an independent random hash from $\mathcal{H}_1$.

Next, let $A = \mathrm{output}_S(S_c^1(\mathit{\Gamma}_1), R^*)$ denote the private output of the sender S in the first phase of Protocol 3.4.6, which in turn is equal to the output of S_{IH} in the interactive hashing protocol, so equivalently $A = \mathrm{output}_{S_{\mathrm{IH}}}(S_{\mathrm{IH}}(Q), R^*)$. Similarly, let $V = \mathrm{view}_{R^*}(S_c^1(\mathit{\Gamma}_1), R^*)$ denote the view of the adversarial receiver R^* in the first phase, which in turn is equal to the view of R^* in the interactive hashing protocol, so equivalently $V = \mathrm{view}_{R^*}(S_{\mathrm{IH}}(Q), R^*)$.

The final step is to use the hiding property of interactive hashing given by Lemma 3.5.9 to bound the collision probability of A (the private output of the sender S) given V (the view of the adversarial receiver R^*) as follows:

$$\mathrm{CP}^{1/2}(A|V) \leq \sqrt{2^{q-k}} \cdot \sqrt{\mathrm{CP}(Q)} \leq \sqrt{2^{q-k}} \cdot \sqrt{2^{-(q-1)}} = \sqrt{2^{-(k-1)}} \ .$$

Property (H.3). In the case when the sender's private input $x \in \Gamma_2$, we analyze the collision probability of the second phase secret as follows. First we observe that for any $x, x' \in \{0,1\}^n$ such that $f(x) = f(x')$, the first phase transcripts for both x and x' are identically distributed, that is $\mathrm{transcript}(S^1(x), R^*) \equiv \mathrm{transcript}(S^1(x'), R^*)$. Thus, if we fix a first phase transcript $\tau \in \mathrm{transcript}(S^1(x), R^*)$ containing a value $y = f(x)$ in the reveal phase, any element in $\Gamma_{2,y} = f^{-1}(y) \subseteq \Gamma_2$ is equally likely to have generated τ. Also observe that the $\Gamma_{2,y}$'s form a partition of Γ_2.

Note that by definition, $|\Gamma_{2,y}| \geq 2^{n-t_0}$, and hence $\mathrm{CP}(\mathit{\Gamma}_{2,y}) \leq 2^{-(n-t_0)} \leq 2^{-(n-t_0-\Delta_2+3)}$. Therefore we can apply Randomness Extraction Lemma 3.5.6 to get $\mathrm{CP}^{1/2}(Q') \leq \sqrt{2^{-(q-1)}}$, for $Q' = (H_2, H_2(\mathit{\Gamma}_{2,y}))$.

Next, let $B_\tau = \mathrm{output}_S(S_c^2(\mathit{\Gamma}_{2,y}), R^*)(\tau)$ denote the private output of the sender S in the second phase, which in turn is equal to the output of S_{IH} in the interactive hashing protocol, so equivalently $B_\tau = \text{output}_{S_{\text{IH}}}(S_{\text{IH}}(Q'), R^*)$. Similarly, let $W_\tau = \text{view}_{R^*}(S_c^2(\Gamma_{2,y}), R^*)(\tau)$ denote the view of the adversarial receiver R^* in the second phase, which in turn is equal to the view of R^* in the interactive hashing protocol, so equivalently $W_\tau = \text{view}_{R^*}(S_{\text{IH}}(Q'), R^*)$.

The final step is to use the hiding property of interactive hashing given by Lemma 3.5.9 to bound the collision probability of B_τ (the private output of the sender S) given W_τ (the view of the adversarial receiver R^*) as follows:

$$\text{CP}^{1/2}(B_\tau | W_\tau) \leq \sqrt{2^{q-k}} \cdot \sqrt{\text{CP}(Q')} \leq \sqrt{2^{q-k}} \cdot \sqrt{2^{-(q-1)}} = \sqrt{2^{-(k-1)}} \ . \qquad \square$$

Protocol 3.4.6 is 1-out-of-2 binding in expected measure

The definition of 1-out-of-2 binding in Definition 3.4.4 considers the first phase (resp., second phase) to be broken if the sender S^* produces valid decommitments to *two* different values after the first commit stage (resp., second commit stage). In this section and Section 3.5.3, we will work with a relaxed notion where we simply bound the *expected* number of values to which the sender can open. To this end, we define openings(S^*, R^1) [resp., openings(S^*, R^2)] to be a random variable denoting the number of values to which the sender successfully opens in phase 1 [resp., phase 2], where 'successfully' opens is defined for each phase analogously to Definition 3.4.4. More formally, for a two-phase commitment scheme (S, R) and a 'binding' set $\mathcal{B}$, we define openings$(S^*, R^1)(\mathcal{B})$ as follows:

- S^* and R_c^1 interact to get first phase commitment $c^{(1)}$.
- After the interaction, S^* outputs a sequence of values $d_1^{(1)}, \dots, d_\ell^{(1)}$ and corresponding full transcripts $\lambda_1, \dots, \lambda_\ell$ of *both* phases. Recall that $\lambda_i = (\tau_i, \kappa_i)$, where τ_i and κ_i are the first-phase and second-phase transcripts, respectively.
- We let openings$(S^*, R^1)(\mathcal{B})$ be the set of distinct values $d_i^{(1)}$ whose opening λ_i is valid, where by valid we mean that λ_i begins with prefix $c^{(1)}$, λ_i contains a decommitment of $c^{(1)}$ to the value $d_i^{(1)}$ with first-phase transcript $\tau_i \notin \mathcal{B}$, and both R_r^1 and R_r^2 accept in λ_i.

Analogously, we define openings$(S^*, R^2)(\tau)$, where τ is a first-phase transcript, as follows:

- S^* and R_c^2 interact to get second phase commitment $c^{(2)}$.
- After the interaction, S^* outputs a sequence of values $d_1^{(2)}, \dots, d_\ell^{(2)}$ and corresponding second-phase transcripts $\kappa_1, \dots, \kappa_\ell$.
- We let openings$(S^*, R^2)(\tau)$ be the set of distinct values $d_i^{(2)}$ whose opening κ_i is valid, where by valid we mean that κ_i starts with prefix c, κ_i contains a decommitment of $c^{(2)}$ to the value $d_i^{(2)}$, and R_r^2 accepts in κ_i.

Now, we can describe the binding property of Protocol 3.4.6 in this language (even when the underlying one-way function has unknown security).

LEMMA 3.5.14

(Protocol 3.4.6 is 1-out-of-2 binding in expected measure.) For every integer $t \in \{1, 2, \ldots, n\}$, $k = O(\log n)$, $\Delta_1 = O(\log n)$, and $\Delta_2 = O(\log n)$, the following holds for the 2-phase commitment scheme (S, R) in Protocol 3.4.6 based on one-way function $f \colon \{0,1\}^n \to \{0,1\}^n$:

There exists a binding set $\mathcal{B}$ for (S, R) where:

(B.1) No (nonuniform) PPT adversary S^* can break the first phase binding with nonnegligible probability in the sense of Definition 3.4.4. That is, for any nonuniform PPT S^*, we have $|\operatorname{openings}(S^*, R^1)(\mathcal{B})| \leq 1$ with probability $1 - \operatorname{neg}(n)$ over the coins of S^* and R^1_c.

(B.2) For all $\tau \in \mathcal{B}$ and any adversarial sender S^*,

$$\mathrm{E}\left[\left|\operatorname{openings}(S^*, R^2)(\tau)\right|\right] < 2 \ ,$$

where the above expectation is taken over the coins of S^* and R^2.

Proof. We follow the proof of the binding property in Lemma 3.4.10, using both Claims 3.4.12 and 3.4.11 from that proof. Let $\mathcal{B} = \{y \in \{0,1\}^n : \Pr[f(U_n) = y] \leq 2^{-t-\Delta_3}\}$ be the same binding set as defined in both claims. We set $\Delta_3 = \Delta_2 + O(\log n)$ to be large enough so that the binding parameter $\operatorname{poly}(n) \cdot 2^{-\Omega(\Delta_3 - \Delta_2)}$ in Claim 3.4.12 is at most 2^{-k}. (This can be done since $k = O(\log n)$.) Now, Claim 3.4.12 states that if $\tau \in \mathcal{B}$, then the second commitment phase is *not* binding—i.e., $\left|\operatorname{openings}(S^*, R^2)(\tau)\right| \geq 2$—with probability at most 2^{-k}. Since $\left|\operatorname{openings}(S^*, R^2)(\tau)\right| \leq 2^k$ (the commitment is to a k-bit string), taking expectations we have

$$\mathrm{E}\left[\left|\operatorname{openings}(S^*, R^2)(\tau)\right|\right] \leq 2^k \cdot 2^{-k} + 1 \cdot (1 - 2^{-k}) < 2 \ .$$

To see why property (B.1) holds, let ε be the probability for which PPT S^* breaks the first phase binding. Observe that the inversion success probability of f from Claim 3.4.11 is

$$\begin{aligned} \varepsilon^{O(1)} \cdot 1/\operatorname{poly}(n) \cdot 2^{-(k+\Delta_1+\Delta_3)} &= \varepsilon^{O(1)} \cdot 1/\operatorname{poly}(n) \cdot 2^{-(k+\Delta_1+\Delta_2+O(\log n))} \\ &= \frac{\varepsilon^{O(1)}}{\operatorname{poly}(n)} \ , \end{aligned}$$

since all $k, \Delta_1, \Delta_2 = O(\log n)$. This forces ε to be a negligible function. □

3.5.3 Converting weakly-hiding to strongly-hiding commitments

In the previous section, we established that Protocol 3.4.6, with appropriate choice of parameters, is $1/n$-hiding in $\mathrm{CP}^{1/2}$ measure (hence, only *weakly hiding*), and 1-out-of-2 binding in expected measure. Our goal in this section is to show how to boost the hiding probability to $\delta = 1 - \operatorname{neg}(n)$, therefore making the scheme *strongly hiding*, while maintaining the 1-out-of-2 binding property.

We first show how to double the hiding probability by combining a constant number of schemes to obtain a new scheme. We then repeat this doubling amplification process $O(\log n)$ times to boost the hiding probability from $1/n$ to a constant $c > 0$, hence obtaining an $\Omega(1)$-hiding scheme. Finally we boost it all the way to $1 - \text{neg}(n)$ by combining polynomial number of $\Omega(1)$-hiding schemes. This is all achieved via a hiding amplification procedure stated next.

ALGORITHM 3.5.15

Hiding amplification procedure, denoted as $\mathsf{Amplify}$.

Input: 2-phase commitment (S, R)

Additional Input Parameters: These are given in unary, and listed below:

1. Security parameter n.
2. Number m of schemes (S, R) to be combined.
3. Integer r denoting S's private input length.
4. Integer k denoting S's private output length.
5. Integer k' denoting $\mathbf{S}$'s private output length.
6. Integer thresholds α_1 and α_2, for the first and second commit phases respectively.

Output: 2-phase commitment $(\mathbf{S}, \mathbf{R})$, as described by Protocol 3.5.16.

To reduce unnecessary clutter, we write $(\mathbf{S}, \mathbf{R}) = \mathsf{Amplify}(S, R)$ when the rest of the parameters are clear from context.

PROTOCOL 3.5.16

Amplified scheme $(\mathbf{S}, \mathbf{R})$ from hiding amplification of base scheme (S, R).

Sender's private input: $x = (x_1, \ldots, x_m) \in \{0,1\}^{mr}$.

First phase commit:

1. $(\mathbf{S}^1_c, \mathbf{R}^1_c)$ does m sequential executions of (S^1_c, R^1_c), using x_i for S^1_c's secret in the i-th execution. Let $(S^1_c[i](x_i), R^1_c[i])$ denote the i-th execution of (S^1_c, R^1_c). Define $a_i = \text{output}_S(S^1_c[i](x_i), R^1_c[i]) \in \{0,1\}^k$, and let $a = (a_1, \ldots, a_m)$.
2. Let $\mathcal{H}_1 = \{h_1 \colon \{0,1\}^{mk} \to \{0,1\}^{\alpha_1}\}$ be a family of pairwise independent hash functions. $\mathbf{S}^1$ chooses a random hash $h_1 \leftarrow \mathcal{H}_1$, and computes $y^{(1)} = (h_1, h_1(a)) \in \{0,1\}^q$.
3. $(\mathbf{S}^1_c, \mathbf{R}^1_c)$ runs the interactive hashing protocol $(S^1_{\text{IH}}(y^{(1)}), R^1_{\text{IH}})(1^q, 1^k)$, given by Protocol 3.2.3, with $\mathbf{S}^1$ and $\mathbf{R}^1$ acting as S^1_{IH} and R^1_{IH}, respectively.

   Let circuit $C \colon \{0,1\}^{k'} \to \{0,1\}^q$ be the common output, and $d^{(1)} \in \{0,1\}^{k'}$ be S^1_{IH}'s private output in $(S^1_{\text{IH}}(y^{(1)}), R^1_{\text{IH}})(1^q, 1^k)$.

First phase sender's private output: String $d^{(1)} \in \{0,1\}^{k'}$.

First phase reveal:

S_r^1 sends tuple $\gamma^{(1)} = (d^{(1)}, a, h_1) \circ (\gamma_1^{(1)}, \ldots, \gamma_m^{(1)})$, where $\gamma_i^{(1)}$ is the first phase revelation string of $S_r^1[i]$ in the above execution of $(S_r^1[i](x_i), R_r^1[i])$.

Receiver $\mathbf{R}_r^1$ accepts if only if $C(d^{(1)}) = (h_1, h_1(a))$ and $R_r^1[i]$ accepts $(\gamma_i^{(1)}, a_i)$ for all $i \in \{1, 2, \ldots, m\}$.

Second phase commit:

Second phase common input: Transcript $\tau = (\tau_1, \ldots, \tau_m)$, where each
$\tau_i = \text{transcript}(S_i^1(x_i), R_i^1)$.

1. $(\mathbf{S}_c^2, \mathbf{R}_c^2)$ does m sequential executions of (S_c^2, R_c^2), using x_i for S^2's secret and transcript τ_i in the i-th execution. Let $(S_c^2[i](x_i), R_c^2[i])(\tau_i)$ denote the i-th execution of (S^2, R^2). Define $b_i = \text{output}_S(S_c^2[i](x_i), R_c^2[i])(\tau_i) \in \{0,1\}^k$, and let $b = (b_1, \ldots, b_m)$.
2. Let $\mathcal{H}_2 = \{h_2 \colon \{0,1\}^{mk} \to \{0,1\}^{\alpha_2}\}$ be a family of pairwise independent hash functions. $\mathbf{S}^2$ chooses a random hash $h_2 \leftarrow \mathcal{H}_2$, and computes $y^{(2)} = (h_2, h_2(b)) \in \{0,1\}^q$.
3. $(\mathbf{S}_c^2, \mathbf{R}_c^2)$ runs the interactive hashing protocol $(S_{\text{IH}}^2(y^{(2)}), R_{\text{IH}}^2)(1^q, 1^k)$, given by Protocol 3.2.3, with $\mathbf{S}_c^2$ and $\mathbf{R}_c^2$ acting as S_{IH}^2 and R_{IH}^2, respectively.

   Let circuit $C \colon \{0,1\}^{k'} \to \{0,1\}^q$ be the common output, and $d^{(2)} \in \{0,1\}^{k'}$ be S_{IH}^2's private output in $(S_{\text{IH}}^2(y^{(2)}), R_{\text{IH}}^2)(1^q, 1^k)$.

Second phase sender's private output: String $d^{(2)} \in \{0,1\}^{k'}$.

Second phase reveal:

S_r^2 sends tuple $\gamma^{(2)} = (d^{(2)}, b, h_2) \circ (\gamma_1^{(2)}, \ldots, \gamma_m^{(2)})$, where $\gamma_i^{(2)}$ is the second phase revelation string of $S_r^2[i]$ in the above execution of $(S_r^2[i](x_i), R_r^2[i])$.

Receiver $\mathbf{R}_r^2$ accepts if only if $C^{(2)}(d^{(2)}) = (h_2, h_2(b))$ and $R_r^2[i]$ accepts $(\gamma_i^{(2)}, b_i)$ for all $i \in \{1, 2, \ldots, m\}$.

Starting from a weakly-hiding scheme (S_0, R_0) of Protocol 3.4.6, we iteratively apply the amplification process $\mathsf{Amplify}$, in a way described by Algorithm 3.5.17 below, to achieve a new scheme (S, R) that we will show to be statistically-hiding. Let $D > 1$ denote a large enough integer constant. We will set the number of schemes to be combined to be $m = D$ in all but the last iteration, in which we set $m = n$.

ALGORITHM 3.5.17

Iterative amplification procedure.

Input: Security parameter n, constant integer $D > 1$, and thresholds $t \in \{1, 2, \ldots, n\}$, $\beta_1, \ldots, \beta_\ell \in \{0, 1, \ldots, D-1\}$, $\beta_{\ell+1} \in \{0, 1, \ldots, n\}$.

1. Let $k_0 = (16D) \cdot \log n$, $\ell = \log n$, and (S_0, R_0) be the 2-phase commitment scheme based on one-way function $f\colon \{0,1\}^n \to \{0,1\}^n$ from Protocol 3.4.6 using parameters t, $k = k_0$, and $\Delta_1 = \Delta_2 = 2\log n$.
2. For $j = 1, 2, \ldots, \ell$, repeat the following:
   (a) Set $k_j = k_{j-1} - 8D - 8$.
   (b) Set $(S_j, R_j) = \mathsf{Amplify}(S_{j-1}, R_{j-1})$ for settings of parameters $m = D$, $r = n \cdot D^{j-1}$, $k = k_{j-1}$, $k' = k_j$, $\alpha_1 = (\beta_j + 1)(k_{j-1} - 1) - 3$ and $\alpha_2 = (D - \beta_j)(k_{j-1} - 1) - 3$.
3. Set $(\mathsf{S}, \mathsf{R}) = \mathsf{Amplify}(S_\ell, R_\ell)$ for settings of parameters $m = n$, $r = n \cdot D^\ell$, $k = k_\ell$, $k' = 1$, $\alpha_1 = \lfloor (\beta_{\ell+1} + \frac{1}{3}\delta n)k \rfloor$ and $\alpha_2 = \lfloor (n - \beta_{\ell+1} + \frac{1}{3}\delta n)k \rfloor$, where $\delta = 1/(2D)$.

Output: 2-phase commitment scheme (S, R).

LEMMA 3.5.18

If scheme (S_0, R_0) used by Algorithm 3.5.17 runs in polynomial time, then scheme (S, R), the output of Algorithm 3.5.17, also runs in polynomial time.

Proof. Scheme (S, R) consists of $n \cdot D^\ell = n \cdot D^{O(\log n)} = \mathrm{poly}(n)$ executions of (S_0, R_0). In addition, each amplification procedure $\mathsf{Amplify}$ adds an overhead time of $\mathrm{poly}(n)$ since both the sender and receiver are doing interactive hashing. Since there are only $1 + n + nD + nD^2 + \cdots + D^{\ell-1} = \mathrm{poly}(n)$ amplifications steps, the overhead time is polynomial. Hence, scheme (S, R) runs in polynomial time if (S_0, R_0) does too. □

The rest of this section, which is technically involved, is devoted to proving the hiding and binding properties of the final scheme (S, R). (In process of doing so, we also analyze the the hiding and binding properties of intermediate schemes (S_j, R_j).)

Hiding amplification

The following two lemmas, Lemma 3.5.19 and 3.5.20, provide us a way to understand the hiding property (in the $\mathrm{CP}^{1/2}$ measure) of amplified scheme $(\mathbf{S}, \mathbf{R})$, in terms of its base scheme (S, R). Lemma 3.5.19 basically say that the hiding probability doubles when we go from (S_{j-1}, R_{j-1}) to $(S_j, R_j) = \mathsf{Amplify}(S_{j-1}, R_{j-1})$ (refer to Step 2b in Algorithm 3.5.17). So if we start up with $1/n$-hiding scheme (S_0, R_0), in $\ell = \log n$ iterations, we will get a scheme (S_ℓ, R_ℓ) with $\Omega(1)$-hiding. Lemma 3.5.20 essentially argues that the final amplification step boost the hiding probability all the way to $1 - \mathrm{neg}(n)$ (in both phases) when starting from a scheme that is $\Omega(1)$-hiding. With these two lemmas, we can establish that the final scheme $(\mathsf{S}, \mathsf{R}) = \mathsf{Amplify}(S_\ell, R_\ell)$ is statistically hiding in both phases.

LEMMA 3.5.19

(Intermediate step hiding amplification.) For any sufficiently large constant $D \in \mathbb{Z}^+$, the following holds:

> If scheme (S, R) is δ-hiding, then there exist an integer $\beta \in \{0, 1, \ldots, D-1\}$ such that scheme $(\mathbf{S}, \mathbf{R}) = \mathsf{Amplify}(S, R)$, with parameters $m = D$, $k' = k - 8D - 8$, $\alpha_1 = (\beta+1)(k-1) - 3$, and $\alpha_2 = (D-\beta)(k-1) - 3$, is δ'-hiding, for $\delta' = \min\{2\delta, 1/D\}$.

Proof. Without loss of generality, we may assume that R^* is deterministic since we can fix the coins of R^* that maximizes the collision probability. Throughout this proof, the value of m will be fixed to D, although we will keep writing m. Let the δ-hiding properties, as stated in Definition 3.5.11, of (S, R) be (H.1), (H.2) and (H.3), respectively. We will prove that $(\mathbf{S}, \mathbf{R})$ satisfies Definition 3.5.11 with Properties (H'.1), (H'.2) and (H'.3) by showing that Property (H.1) implies (H'.1), and so forth.

Property (H.1) implies (H'.1). Let Γ_1 and Γ_2 be the corresponding sets for (S, R). Define the sets Γ'_1 and Γ'_2 in terms as follows (the value of β will be determined later).

$$\begin{aligned}\Gamma'_1 &= \{(x_1, \ldots, x_m) : \exists\, i_1, \ldots, i_{\beta+1} \text{ such that } x_{i_1}, \ldots, x_{i_{\beta+1}} \in \Gamma_1\} \ , \\ \Gamma'_2 &= \{(x_1, \ldots, x_m) : \exists\, i_1, \ldots, i_{m-\beta} \text{ such that } x_{i_1}, \ldots, x_{i_{m-\beta}} \in \Gamma_2\} \ .\end{aligned}$$

By the way we defined Γ'_1 and Γ'_2 together with the fact that $\Gamma_1 \cup \Gamma_2 = \{0,1\}^r$, it is the case that $\Gamma'_1 \cup \Gamma'_2 = \{0,1\}^{mr}$. This is because either at least $\beta+1$ of the x_i are in Γ_1 (in which case, $(x_1, \ldots, x_m) \in \Gamma'_1$) or else at most β of the x_i are in Γ_1, which implies that at least $m - \beta$ of the x_i are in Γ_2 (in which case, $(x_1, \ldots, x_m) \in \Gamma'_2$).

We are given that $\mu(\Gamma_1 \cap \Gamma_2) \geq \delta$. Define $\delta' = \min\{\delta, 1/(2m)\}$. What we need to show is that $\mu(\Gamma'_1 \cap \Gamma'_2) \geq \delta'$. Choose any subset $S \subseteq \Gamma_1 \cap \Gamma_2$ such that $\mu(S) = \delta'$. Hence, we have

$$\Pr_{x_1, \ldots, x_m \leftarrow \{0,1\}^r}[\text{exactly one } x_i \in S] = m\delta'(1-\delta')^{m-1} \geq m\delta'(1 - 1/(m-1))^{m-1} = \Omega(m\delta') \ .$$

Given that exactly one $x_i \in S$, assume without loss of generality that $x_m \in S$. Let p_t denote the conditional probability that exactly t of the rest of the $m-1$ x_i's are in $\Gamma_1 \setminus \Gamma_2$. Choose $\beta \in [0, m-1]$ to maximize p_t, i.e., $\beta = \mathrm{argmax}_t\, p_t$. Let I_i, for $i = 1, 2, \ldots, m-1$, be a binary random variable indicating whether $x_i \in \Gamma_1$ or not; note that these are independent random variables conditioned on the fact that $x_m \in S$. Let the μ the mean of the I_i's. By a Chernoff bound,

$$\Pr\left[\left|\sum_i I_i - \mu\cdot(m-1)\right| > 3\sqrt{m-1}\right] \leq 2e^{((m-1)/3)\cdot(3/\sqrt{m-1})^2} < 1/2 \ .$$

This means that greater 1/2 of the weight is centered around $\mu\cdot(m-1)\pm 3\sqrt{m-1}$. Since we chose $\beta = \mathrm{argmax}_t\, p_t$ in a maximal way, we have

$$\Pr_{x_1,\ldots,x_m \leftarrow \{0,1\}^r}[\text{exactly } \beta \text{ of } x_i\text{'s are in } \Gamma_1 \setminus S \mid \text{exactly one } x_i \in S] = \Omega\left(\frac{1}{\sqrt{m}}\right) \ .$$

Knowing that $\Gamma_1 \cup \Gamma_2 = \{0,1\}^r$, if exactly β of x_i's in $\Gamma_1 \setminus S$ and exactly one $x_i \in S$, then there must be at least $m-1-\beta$ of x_i's in $\Gamma_2 \setminus S$. Consequently,

$$\begin{aligned}\Pr_{x_1,\ldots,x_m \leftarrow \{0,1\}^r}[(x_1,\ldots,x_m) \in \Gamma_1' \cap \Gamma_2'] &= \Omega(m\delta')\cdot\Omega\left(\frac{1}{\sqrt{m}}\right)\\ &= \Omega(\sqrt{m}\delta')\\ &> 2\delta' = \min\{2\delta, 1/m\},\end{aligned}$$

where the last inequality holds when $m = D$ is a large enough constant.

Property (H.2) implies (H'.2). In the first commitment phase $(\mathbf{S}_c^1, R^*)$, the cheating receiver R^* interacts with m sequential executions of S_c^1. Here we must analyze the case when S_c^1's private input in these m executions, given by $x = (x_1,\ldots,x_m)$, are distributed uniformly in Γ_1'. We let $A_i(x)$ denote the private output of the sender and $V_i(x)$ the view of the receiver in the i'th execution, for x being the private input for $\mathbf{S}_c^1$. That is, for $i = 1,\ldots,m$,

$$\begin{aligned}A_i(x) &= \mathrm{output}_S(S_c^1(x_i), R^*(V_1,\ldots,V_{i-1}));\\ V_i(x) &= \mathrm{view}_{R^*}(S_c^1(x_i), R^*(V_1,\ldots,V_{i-1})).\end{aligned}$$

Note that while the sender's behavior in the i'th execution is independent of the previous executions, the cheating receiver may base its strategy on its previous views. We want to bound $\mathrm{CP}^{1/2}(A''(\mathit{\Gamma}_1')|V''(\mathit{\Gamma}_1'))$, where $A''(\mathit{\Gamma}_1') = (A_1(\mathit{\Gamma}_1'),\ldots,A_m(\mathit{\Gamma}_1'))$ represents the combined first-phase private outputs of the m senders, and $V''(\mathit{\Gamma}_1') = (V_1(\mathit{\Gamma}_1'),\ldots,V_m(\mathit{\Gamma}_1'))$ represents the view of R^* when interacting with these m senders. Note that random variable $\mathit{\Gamma}_1'$ represents an independent random element from the set Γ_1'. To do this, we consider, for each $I \subseteq [m]$ of size at least $\beta+1$, the random variable $\mathit{\Gamma}_1'|_I$ for private input of the sender $\mathbf{S}$, where $\mathit{\Gamma}_1'|_I$ represents choosing x_i uniformly in Γ_1 for $i \in I$, and uniformly in $\overline{\Gamma_1}$ for $i \notin I$. To get a bound on $\mathrm{CP}^{1/2}(A''(\mathit{\Gamma}_1'|_I)|V''(\mathit{\Gamma}_1'|_I))$, we will have to refer to Lemma 3.5.5 and see why the (A_i, V_i)'s satisfy the two conditions of the lemma.

Conditioned on the any previous view—namely, $V_1(\Gamma_1'|_I) = v_1, \ldots, V_{i-1}(\Gamma_1'|_I) = v_{i-1}$ for any $v_1, \ldots, v_{i-1}$—it is the case that $\mathrm{CP}^{1/2}(A_i(\Gamma_1'|_I)|V_i(\Gamma_1'|_I)) \le \sqrt{2^{-(k-1)}}$ if $i \in I$. This follows from Property (H.2) because the (unbounded) receiver R^* can incorporate the previous view $v_1, \ldots, v_{i-1}$ as nonuniform advice, and then the only randomness in the definition of A_i and V_i is the sender's coins $x_i \leftarrow (\Gamma_1'|_I)_i$, which are uniform in Γ_1 (even conditioned on $v_1, \ldots, v_{i-1}$). This shows that the first condition of Lemma 3.5.5 is satisfied.

For the second condition, what we need to show is that conditioned on $V_1(\Gamma_1'|_I) = v_1, \ldots, V_i(\Gamma_1'|_I) = v_i$, the random variables $A_1(\Gamma_1'|_I), \ldots, A_i(\Gamma_1'|_I), V_{i+1}(\Gamma_1'|_I)$ are independent. This can be seen by induction on i as follows. It is vacuously true for $i = 0$. Assuming it is true for $i = j - 1$, we prove it for $i = j$ as follows. First condition on $v_1, \ldots, v_{j-1}$. By inductive hypothesis, $A_1, \ldots, A_{j-1}, V_j$ are independent (omitting $\Gamma_1'|_I$ from the notation for readability). Moreover, since we have conditioned on $v_1, \ldots, v_{j-1}$, A_j and V_j are functions of only $(\Gamma_1'|_I)_j$, the sender's coins in the j'th execution, which is independent of $A_1, \ldots, A_{j-1}$ (because we have only used $(\Gamma_1'|_I)_1, \ldots, (\Gamma_1'|_I)_{j-1}$ so far). Thus, if we condition on $V_j = v_j$, A_j remains independent of $A_1, \ldots, A_{j-1}$. V_{j+1} is independent of $A_1, \ldots, A_j$ because now it is only a function of $(\Gamma_1'|_I)_{j+1}$, which has not been used yet.

Applying Lemma 3.5.5, we have

$$\mathrm{CP}^{1/2}(A''(\Gamma_1'|_I)|V''(\Gamma_1'|_I)) \le \sqrt{2^{-(\beta+1)(k-1)}}, \tag{3.4}$$

since from property (H.2), it is the case that for all $i \in I$, $\mathrm{CP}^{1/2}(A_i|V_i) \le \sqrt{2^{-(k-1)}}$ (even conditioned on the previous views), and $|I| \ge \beta + 1$.

Now, to bound $\mathrm{CP}^{1/2}(A''(\Gamma_1')|V''(\Gamma_1'))$ where X is uniform in Γ_1', we observe that $\Gamma_1' = \Gamma_1'|_{\mathcal{I}}$, where $\mathcal{I}$ is the random variable on subsets I of size at least $\beta + 1$ given by

$$\Pr[\mathcal{I} = I] = \Pr_{(x_1,\ldots,x_m)\leftarrow\Gamma_1'}[\{i : x_i \in \Gamma_1\} = I].$$

In other words, sampling from Γ_1' can be broken into two steps; first sampling an $I \leftarrow \mathcal{I}$, and then sampling $x_i \leftarrow \Gamma_1$ for $i \in I$, and $x_i \leftarrow \overline{\Gamma_1}$ for $i \notin I$. Therefore, we have

$$\begin{aligned}
\mathrm{CP}^{1/2}(A''(\Gamma_1'|_{\mathcal{I}})|V''(\Gamma_1'|_{\mathcal{I}})) &\le \mathrm{CP}^{1/2}(A''(\Gamma_1'|_{\mathcal{I}})|(V''(\Gamma_1'|_{\mathcal{I}}), \mathcal{I})) \qquad \text{(by Lemma 3.5.7)} \\
&= \mathop{\mathrm{E}}_{I\leftarrow\mathcal{I}}\left[\mathrm{CP}^{1/2}(A''(\Gamma_1'|_I)|V''(\Gamma_1'|_I)\right] \\
&\le \sqrt{2^{-(\beta+1)(k-1)}} \qquad\qquad (3.5) \\
&= \sqrt{2^{-(\alpha_1+3)}},
\end{aligned}$$

with the last inequality following from (3.4). Therefore we can apply Randomness Extraction Lemma 3.5.6 to get $\mathrm{CP}^{1/2}(H_1, H_1(A''(\Gamma_1'))|V''(\Gamma_1')) \le \sqrt{2^{-(q-1)}}$, where H_1 is an independent random hash from $\mathcal{H}_1$.

Next, let $A' = \mathrm{output}_{\mathbf{S}}(\mathbf{S}^1(\Gamma_1'), R^*)$ denote the private output of the sender $\mathbf{S}$ in the first phase, which in turn is equal to the output of S_{IH} in the interactive hashing protocol, so equivalently $A' = \text{output}_{S_{\text{IH}}}(S_{\text{IH}}(Q), R^*_{\text{IH}})$, where $Q = (H_1, H_1(A''(\varGamma'_1)))$. Similarly, let $V' = \text{view}_{R^*}(\mathbf{S}^1(\Gamma'_1), R^*)$ denote the view of the adversarial receiver R^* in the first phase, which in turn is equal to the view of R^* in the interactive hashing protocol, so equivalently $V' = (\text{view}_{R^*_{\text{IH}}}(S_{\text{IH}}(Q), R^*_{\text{IH}}), V'')$, for the same $Q = (H_1, H_1(A''(\varGamma'_1)))$.

The final step is to use the hiding property of interactive hashing given by Lemma 3.5.9 to bound the collision probability of A' (the private output of the sender $\mathbf{S}$) given V' (the view of the adversarial receiver R^*) as follows:

$$\text{CP}^{1/2}(A'|V') \le \sqrt{2^{q-k'}} \cdot \text{CP}^{1/2}(Q|V'') \le \sqrt{2^{q-k'}} \cdot \sqrt{2^{-(q-1)}} = \sqrt{2^{-(k'-1)}}\ .$$

Property (H.3) implies (H'.3). Fix a transcript $\tau' \in \text{Supp}(\text{T}')$, where random variable $\text{T}' = \text{transcript}(\mathbf{S}^1(\varGamma'_2), R^*)$. Transcript τ' contains first-phase transcripts $(\tau_1, \ldots, \tau_m)$ for the m executions of (S, R). Similarly to the above proof of Property (H'.2), we define the following random variables:

$$B_i(x) = \text{output}_S(S^2_c(x_i), R^*(W_1, \ldots, W_{i-1})(\tau_i));$$
$$W_i(x) = \text{view}_{R^*}(S^2_c(x_i), R^*(W_1, \ldots, W_{i-1})(\tau_i)),$$

where x_i are the coins of the sender in the i'th execution of the the (S, R). For notational simplicity, we omit the sender's coin-tosses from the first-phase interactive hashing (they can be considered fixed for the analysis below). As above, we want to bound $\text{CP}^{1/2}(B''(X_{\tau'})|W''(X_{\tau'}))$, where random variable $B''(X_{\tau'}) = (B_1(X_{\tau'}), \ldots, B_m(X_{\tau'}))$ represents the combined second-phase private outputs of the m senders, and random variable $W''(X_{\tau'}) = (W_1(X_{\tau'}), \ldots, W_m(X_{\tau'}))$ represents the view of R^* when interacting with these m senders, with $X_{\tau'}$ being a shorthand for $\varGamma'_2|_{\text{T}(\varGamma'_2)=\tau'}$. To do this, we consider, for each subset $J \subseteq [m]$ of size at least $m - \beta$, the random variable X_J for private input of the sender $\mathbf{S}$, where X_J represents choosing x_i uniformly in Γ_2 for $i \in J$, and uniformly in $\overline{\Gamma_2}$ for $i \notin J$.

It is important to note that even when we condition on $\text{T}'(X_J) = \tau'$, the components $(X_1, \ldots, X_m)$ of X_J remain independent, and the distribution of $X_i|_{\text{T}'(X_J)=\tau'}$ is equivalent to $X_i|_{\text{T}(X_i)=\tau_i}$, where only condition on the transcript of the i'th execution. (Similarly to the inductive proof above, it can be shown that $(X_1, \ldots, X_m)$ are independent given the receiver's view V_m of the m executions of S^1_c. The only additional information revealed about the X_i's in the first phase is $(A_1, \ldots, A_m)$, where A_i is a function only of X_i once we condition on V_m.)

Thus from property (H.3), we have for all $i \in J$, $\text{CP}^{1/2}(B_i(X_{J,\tau'})|W_i(X_{J,\tau'})) \le \sqrt{2^{-(k-1)}}$, where $X_{J,\tau'} = \varGamma'_2|_J|_{\text{T}'(\varGamma'_2|_J)=\tau'}$, and this holds even conditioned on the previous views. Similar to the first phase, we apply Lemma 3.5.5 to show that

$$\text{CP}^{1/2}(B''(X_{J,\tau'})|W''(X_{J,\tau'})) \le \sqrt{2^{-(m-\beta)(k-1)}}\ .$$

Again analogous to the first phase, we observe that $X_{\tau'} = X_{\mathcal{J},\tau'}$ for an appropriate random variable $\mathcal{J}$ on sets of size at least $m - \beta$, and thus

$$\begin{aligned}\mathrm{CP}^{1/2}(B''(X_{\tau'})|W''(X_{\tau'})) &\leq \sqrt{2^{-(m-\beta)(k-1)}} \\ &= \sqrt{2^{-(\alpha_2+3)}}.\end{aligned} \tag{3.6}$$

By the Randomness Extraction Lemma 3.5.6, we get $\mathrm{CP}^{1/2}(H_2, H_2(B''(X_{\tau'}))|W''(X_{\tau'})) \leq \sqrt{2^{-(q-1)}}$.

The final step is to use the hiding property of interactive hashing given by Lemma 3.5.9 to bound the collision probability of B_τ (the private output of the sender S) given W_τ (the view of the adversarial receiver R^*) as follows:

$$\mathrm{CP}^{1/2}(B'_{\tau'}|W'_{\tau'}) \leq \sqrt{2^{q-k'}} \cdot \sqrt{2^{-(q-1)}} = \sqrt{2^{-(k'-1)}}\ . \qquad \square$$

LEMMA 3.5.20

(Final step hiding amplification.) The following statement holds for every constant $\delta > 0$ and every integer $k \geq 100/\delta$:

> If scheme (S, R) is δ-hiding, then there exist an integer $\beta \in [0, n]$ such that scheme $(\mathbf{S}, \mathbf{R}) = \mathsf{Amplify}(S, R)$, with parameters $m = n$, $k' = 1$, $\alpha_1 = \lfloor(\beta + \frac{1}{3}\delta n)k\rfloor$ and $\alpha_2 = \lfloor(n - \beta + \frac{1}{3}\delta n)k\rfloor$, is statistically hiding in the sense of Definition 3.4.3.

Proof. Let the δ-hiding properties, as stated in Definition 3.5.11, of (S, R) be (H.1), (H.2) and (H.3), respectively. To prove that scheme $(\mathbf{S}, \mathbf{R})$ is statistically hiding, it suffices to show that there exists sets $\Gamma'_1, \Gamma'_2 \subseteq \{0,1\}^{nr}$ such that the following holds for every adversarial receiver R^*:

(H'.1) Both $\mu(\Gamma'_1), \mu(\Gamma'_2) \geq 1 - 2^{-\Omega(n)}$.

(H'.2) (A', V') is $2^{-\Omega(n)}$-close to (U_1, V'), where $A' = \mathrm{output}_{\mathbf{S}}(\mathbf{S}^1_c(\mathit{\Gamma}'_1), R^*)$ denotes the private output of the sender $\mathbf{S}$ in the first phase, and $V' = \mathrm{view}_{R^*}(\mathbf{S}^1_c(\mathit{\Gamma}'_1), R^*)$ denotes the view of the adversarial receiver R^* in the first phase.

(H'.3) For all $\tau' \in \mathrm{Supp}(\mathrm{T}')$, $(B'_{\tau'}, W'_{\tau'})$ is $2^{-\Omega(n)}$-close to $(U_1, W'_{\tau'})$, where random variable $(B'_{\tau'}, W'_{\tau'}) = (\mathrm{output}_{\mathbf{S}}(\mathbf{S}^2_c(\mathit{\Gamma}'_2), R^*), \mathrm{view}_{R^*}(\mathbf{S}^2_c(\mathit{\Gamma}'_2), R^*))|_{\mathrm{T}'=\tau'}$, and random variable $\mathrm{T}' = \mathrm{transcript}(\mathbf{S}^1(\mathit{\Gamma}'_2), R^*)$. We view $B'_{\tau'}$ as representing the private output of the sender $\mathbf{S}$ in the second phase given that the first-phase transcript is τ'. Similarly, we view $W'_{\tau'}$ as representing the view of the adversarial receiver R^* in the second phase given that the first-phase transcript is τ'.

Property (H.1) implies (H'.1). Let Γ_1 and Γ_2 be the corresponding sets for (S, R), and let $p = \mu(\Gamma_1)$. Set $\beta = \lfloor pn - \frac{1}{2}\delta n\rfloor$, $\gamma_1 = \lfloor pn - \frac{1}{12}\delta n\rfloor$ and $\gamma_2 = \lfloor(1 - p + \delta)n - \frac{1}{12}\delta n\rfloor$.

Note that $\beta \in [0, n]$ since $p \in [\delta, 1]$.

Define the sets Γ'_1 and Γ'_2 as follows:

$$\begin{aligned}
\Gamma'_1 &= \{(x_1, \ldots, x_n) : \exists\, i_1, \ldots, i_{\gamma_1} \text{ such that } x_{i_1}, \ldots, x_{i_{\gamma_1}} \in \Gamma_1\}, \\
\Gamma'_2 &= \{(x_1, \ldots, x_n) : \exists\, i_1, \ldots, i_{\gamma_2} \text{ such that } x_{i_1}, \ldots, x_{i_{\gamma_2}} \in \Gamma_2\}.
\end{aligned}$$

To lower bound $\mu(\Gamma'_1)$, note that $\mu(\Gamma_1) - \gamma_1/n = p - \left\lfloor pn - \frac{1}{12}\delta n \right\rfloor / n \geq \frac{1}{12}\delta = \Omega(1)$ since $\delta = \Omega(1)$. Using a Chernoff bound, we get

$$\begin{aligned}
\mu(\Gamma'_1) &= 1 - \Pr_{(x_1,\ldots,x_n)}[\text{less than } \gamma_1 \text{ of the } x_i\text{'s are in } \Gamma_1] \\
&= 1 - 2^{-\Omega(n)}.
\end{aligned}$$

To analyze $\mu(\Gamma'_2)$, we note that $\mu(\Gamma_2) - \gamma_2/n = (1-p+\delta) - \left\lfloor (1-p+\delta)n - \frac{1}{12}\delta n \right\rfloor / n \geq \frac{1}{12}\delta = \Omega(1)$. Using a similar analysis as above, we get $\mu(\Gamma'_2) = 1 - 2^{-\Omega(n)}$.

Property (H.2) implies (H'.2). Using the same notations and analysis as in the proof of Lemma 3.5.19, we let $A_i(x)$ denote the private output of the sender and $V_i(x)$ the view of the receiver in the i'th execution, for x being the private input for $\mathbf{S}^1_c$. That is, for $i = 1, \ldots, n$,

$$\begin{aligned}
A_i(x) &= \mathrm{output}_S(S^1_c(x_i), R^*(V_1, \ldots, V_{i-1})); \\
V_i(x) &= \mathrm{view}_{R^*}(S^1_c(x_i), R^*(V_1, \ldots, V_{i-1})).
\end{aligned}$$

Let $A''(\varGamma'_1) = (A_1(\varGamma'_1), \ldots, A_n(\varGamma'_1))$ represent the combined first-phase private outputs of the n senders, and $V''(\varGamma'_1) = (V_1(\varGamma'_1), \ldots, V_n(\varGamma'_1))$ represent the view of R^* when interacting with these n senders, before interactive hashing is done. From now on, we simplify notation by making $A'' = A''(\varGamma'_1)$ and $V'' = V''(\varGamma'_1)$.

Similar to (3.5) as in the proof of Lemma 3.5.19, we obtain

$$\mathrm{CP}^{1/2}(A'' | V'') \leq \sqrt{2^{-\gamma_1 \cdot (k-1)}} \ .$$

And by a Markov bound, we know that with probability greater than $1 - 2^{-n}$ over $v'' \leftarrow V''$,

$$\mathrm{CP}(A''|_{V''=v''}) \leq 2^{-\gamma_1(k-1)} \cdot 2^{2n} \leq 2^{-\alpha_1 - (1/24)\delta k n + 3n} \leq 2^{-(\alpha_1 + n)}, \tag{3.7}$$

with the last inequality following from $k \geq 100/\delta$.

Consider $v'' \in V''$ such that the above (3.7) holds. Let $Q = (H_1, H_1(A''))$, where H_1 is an independent random hash from $\mathcal{H}_1$. Because H_1 is independent, $Q|_{V''=v''} = (H_1, H_1(A''|_{V''=v''}))$, and we can apply the Leftover Hash Lemma 3.5.8 to obtain that $Q|_{V''=v''}$, the input to the interactive hashing protocol, is $2^{-\Omega(n)}$-close to uniform.

Next, let $A' = \mathrm{output}_{\mathbf{S}}(\mathbf{S}^1(\varGamma'_1), R^*)$ denote the private output of $\mathbf{S}$ in the first phase, which in turn is equal to the output of $S_{\rm IH}$ in the interactive hashing protocol, so equivalently $A' = \text{output}_{S_{\rm IH}}(S_{\rm IH}(Q), R^*)$. Similarly, let $V' = \text{view}_{R^*}(S^1_c(\mathit{\Gamma}_1), R^*)$ denote the view of the adversarial receiver R^* in the first phase, and let $V_{\rm IH} = (\text{view}_{R^*_{\rm IH}}(S_{\rm IH}(Q), R^*_{\rm IH})$ denote the view of receiver R^* during the interactive hashing execution only. Observe that $V' = (V'', V_{\rm IH})$, recalling that V'' is the view of R^* when interacting with these n senders, before interactive hashing is done.

Because $Q|_{V''=v''}$, the input to interactive hashing, is $2^{-\Omega(n)}$-close to uniform, we know that $(A'|_{V''=v''}, V_{\rm IH}|_{V''=v''})$ is $2^{-\Omega(n)}$-close to $(U_1, V_{\rm IH}|_{V''=v''})$, as guaranteed by the hiding property of interactive hashing (see Definition 3.2.1). So the $\mathbf{S}$'s private output $A'|_{V''=v''}$ is hidden for any $v'' \in V''$ satisfying the above (3.7). Finally note that (3.7) is satisfied for all but a 2^{-n} fraction of $v'' \leftarrow V''$, so it follows that (A', V') is $2^{-\Omega(n)}$-close to (U_1, V'), as required.

Property (H.3) implies (H'.3). Using similar ideas in the proof of Lemma 3.5.19, we can proceed as above and obtain that Property (H'.3) holds assuming (H.3). □

Binding preservation

In the execution of Algorithm 3.5.17, we obtained ℓ intermediate commitment schemes $[(S_j, R_j)]_{1\le j\le \ell}$, and one final commitment scheme (S, R). Our goal is to prove that the final scheme (S, R) satisfies the 1-out-of-2 binding property of Definition 3.4.4. To achieve our goal, we inductively show that the *expected* number of openings a sender can produce in the intermediate schemes is bounded by some constant, namely 32. (This is captured by Lemma 3.5.22 below.) Then in the final step, for scheme (S, R), we show how to shrink this expectation to value that is very close to 1, effectively proving that scheme (S, R) is satisfies the 1-out-of-2 binding property. (This in turn is captured by Lemma 3.5.24.)

In the definition of the computational 1-out-of-2 binding property (Definition 3.4.4), we stipulated that the adversarial sender in the second phase can be computationally unbounded, whereas the adversarial sender in the first phase must be probabilistic polynomial time (PPT). It will be rather messy to work with PPT senders, hence we will first abstract away the PPT requirement by showing, in the next section, how to convert any PPT sender violating the 1-out-of-2 binding property in the first phase into a computationally unbounded sender with a special *unique binding* property. A sender with the unique binding property, intuitively, will not break the (first-phase) binding property of any execution of the initial schemes (S_0, R_0), but might still break the binding property of the intermediate schemes (S_j, R_j) (or final scheme (S, R)). Intuitively, we can restrict to such senders because of the computational 1-out-of-2 binding property of the initial scheme (S_0, R_0). Once we have a sender with the unique binding property, the analysis of the amplification steps is entirely information theoretic.

To formally define the unique binding property for senders, we observe that schemes $[(S_j, R_j)]_{1\le j\le \ell}$ and (S, R) each contain multiple executions of initial scheme (S_0, R_0). Hence, when a cheating sender S^* interacts with R_j, it is actually also interacting with the i-th execution of R_0, for each $i = 1, 2, \ldots$, which we will denote by $R_0[i]$. Formally, we obtain a (computationally unbounded) cheating sender strategy $S^*[i]$ that interacts with this single execution of $R_0[i]$ (more precisely, the first commit stage $R^1_{0,c}[i]$), by simulating all of the other messages of R_j on its own until the end of the first commit stage of $R_0[i]$. Then it enumerates over all choices for the subsequent messages of R_j and outputs all of the resulting transcripts of S^*'s interactions with $R_0[i]$ together with the corresponding first-phase decommitment values.

DEFINITION 3.5.21

(Unique binding property of sender.) For intermediate schemes $[(S_j, R_j)]_{1 \le j \le \ell}$ and final scheme (S, R), a (deterministic) sender S^* has the ***unique binding*** property if for all i, we have $|\operatorname{openings}(S^*[i], R_0[i])| \le 1$ with probability 1 (over the coins of $S^*[i]$[11] and $R_0[i]$) where openings$(\cdot)$ is defined as in Section 3.5.2.

The following two lemmas, Lemma 3.5.22 and 3.5.24, provide us a way to understand the binding property (in an average case sense) of $(\mathbf{S}, \mathbf{R})$, the amplified hiding scheme as presented in Protocol 3.5.16, in terms of (S, R). We might occasionally drop the superscript notations (1) and (2) from the notations if it is clear which phase we are referring to.

LEMMA 3.5.22

(Intermediate step binding preservation.) For some constant $D \in \mathbb{N}$ and any integers $t \in [1, n]$, $\beta_1, \ldots, \beta_\ell \in \{0, 1, \ldots, D-1\}$, and $\beta_{\ell+1} \in [0, n]$, letting $[(S_j, R_j)]_{1 \le j \le \ell}$ be the intermediate commitment schemes obtained in the execution of Algorithm 3.5.17 with parameters D, t, and $(\beta_1, \ldots, \beta_{\ell+1})$, there exists a binding set $\mathcal{B}$ such that the following two conditions hold for each $j = 1, 2, \ldots, \ell$:

(B.1) For every deterministic sender S^* with the *unique binding property*,

$$\mathrm{E}\left[\left|\operatorname{openings}(S^*, R^1_j)(\mathcal{B})\right|\right] < 32 \ ,$$

where the expectation is taken over the coins tosses of R^1_j.

(B.2) For every $\tau \in \mathcal{B}$ and for every deterministic sender S^*,

$$\mathrm{E}\left[\left|\operatorname{openings}(S^*, R^2_j)(\tau)\right|\right] < 32 \ ,$$

where the expectation is taken over the coins tosses of R^2_j.

Proof. We proceed to prove by induction on j. In fact, we will start with a base case of

[11] Note that $S^*[i]$ is probabilistic even if S^* is deterministic, because it simulates all of the random choices of R_j other than those of $R_0[i]$.

$j = 0$, i.e., consider the scheme (S_0, R_0) from Section 3.5.2. By Lemma 3.5.14, we know that scheme (S_0, R_0) satisfies both conditions (B.1) and (B.2). (Although Lemma 3.5.14 guarantees that (S_0, R_0) satisfies condition (B.1) only for PPT S^*, it is also trivially satisfied for computationally unbounded S^* with the unique binding property.)

For the inductive step, we assume (S_j, R_j) satisfy both (B.1) and (B.2), and show that so does (S_{j+1}, R_{j+1}). Note that (S_{j+1}, R_{j+1}) is obtained by the amplification procedure (Protocol 3.5.16) that combines m sequential executions of (S_j, R_j), i.e., $(S_{j+1}, R_{j+1}) = \mathsf{Amplify}(S_j, R_j)$. Hence, for convenience of notation we will denote (S_j, R_j) and (S_{j+1}, R_{j+1}) as (S, R) and $(\mathbf{S}, \mathbf{R})$ respectively. The i-th execution of (S, R) in $(\mathbf{S}, \mathbf{R})$ is denoted as $(S[i], R[i])$, not to be confused with the subscript indexing notation of (S_j, R_j).

Also throughout this proof, the value of m will be fixed to D, although we will keep writing m. Let $\mathcal{B}$ be the binding set for (S, R). We define our new binding set $\mathcal{B}'$ for $(\mathbf{S}, \mathbf{R})$ in terms of $\mathcal{B}$ as follows:

$$\mathcal{B}' = \{(\tau_1, \ldots, \tau_m) : \exists\ j_1, \ldots, j_{\beta+1} \text{ such that } \tau_{j_1}, \ldots, \tau_{j_{\beta+1}} \in \mathcal{B}\} \ .$$

That is, a transcript $\tau' = (\tau_1, \ldots, \tau_m) \in \mathcal{B}'$ if and only if at least $\beta + 1$ of τ_j's are in $\mathcal{B}$. Conversely, $\tau' \notin \mathcal{B}'$ if and only if at least $m - \beta$ of the τ_j's are not in $\mathcal{B}$.

Property (B.1). Consider a deterministic S^* with the unique binding property interacting with $\mathbf{R}^1$. The random coins of $\mathbf{R}^1$ can be broken up into independent random coins of $R^1[1], \ldots, R^1[m]$ and R^1_{IH}, the receiver in the interactive hashing.

Recall that the m executions of (S, R) in $(\mathbf{S}, \mathbf{R})$ are sequential. We want to focus on the interaction of S^* with (the commit phase of) $R^1[i]$. To do so, define $S^*[i]$, the sender interacting with $R^1[i]$, as follows: $S^*[i]$ simulates S^* using fixed coins r_j for all the previous $R^1[j]$'s (for all $j < i$) and after the interaction with $R^1[i]$, $S^*[i]$ outputs all the valid openings that occur in some continuation of S^*'s interaction with $R[i]$ (by enumerating over all coins of the future $R[j]$'s, $j > i$, the coins of R^1_{IH}, and the coins of $\mathbf{R}^2$). Observe that $S^*[i]$ inherits the unique binding property from S^*. We will write $S^*[i](r_1, \ldots, r_{i-1})$ to indicate the fixed coins r_j that are used by $S^*[i]$ in simulating $R^1[j]$.

Let $X_i(r_1, \ldots, r_i) = \left|\text{openings}(S^*[i](r_1, \ldots, r_{i-1}, R^1[i](r_i))(\mathcal{B})\right|$; in other words, count of the number of valid decommitment in i-th execution, when the sender uses simulated coins $r_1, \ldots, r_{i-1}$ and $R^1[i]$ uses coins r_i. Let $U = (U_1, \ldots, U_m)$, where U_i denotes the uniform random variable on coins r_i for $R[i]$; note that these are independent because the honest receiver tosses independent coins for each execution. We now consider the random variables $X_i(U) = X_i(U_1, \ldots, U_i)$.

By our induction hypothesis, for all fixed $(r_1, \ldots, r_{i-1})$, we have

$$\mathrm{E}\left[X_i(U) | U_1 = r_1, \ldots, U_{i-1} = r_{i-1}\right] = \mathrm{E}\left[X_i(r_1, \ldots, r_{i-1}, U_i)\right] < 32 \ .$$

Because the previous $X_j(U)$'s, for $j < i$, only depend on $U_1, \ldots, U_j$, we have that the expected value of X_i is less than 32 even given any previous values of X_j's. That is,

$\mathrm{E}\left[X_i|_{X_1=x_1,\ldots,X_{i-1}=x_{i-1}}\right] < 32$ for any $(x_1,\ldots,x_{i-1}) \in \mathrm{Supp}(X_1,\ldots,X_{i-1})$. The following claim allows us to bound the expectation of the product of these random variables.

CLAIM 3.5.23

Let $Y_1,\ldots,Y_\ell$ be nonnegative real-valued random variables such that for all $i = 1,2,\ldots,\ell$, we have $\mathrm{E}[Y_i|_{Y_1=y_1,\ldots,Y_{i-1}=y_{i-1}}] < \alpha_i \in \mathbb{R}^+$, for every $(y_1,\ldots,y_{i-1}) \in \mathrm{Supp}(Y_1,\ldots,Y_{i-1})$. Then,

$$\mathrm{E}\left[\prod_{i=1}^{\ell} Y_i\right] < \prod_{i=1}^{\ell} \alpha_i \ .$$

Proof of Claim. Note that

$$\begin{aligned}
\mathrm{E}[Y_1\cdots Y_\ell] &= \mathrm{E}\big[\mathrm{E}[Y_1\cdots Y_\ell \mid Y_1\cdots Y_{\ell-1}]\big] \\
&= \mathrm{E}\big[Y_1\cdots Y_{\ell-1}\cdot \mathrm{E}[Y_\ell \mid Y_1\cdots Y_{\ell-1}]\big] \\
&< \mathrm{E}[Y_1\cdots Y_{\ell-1}\cdot \alpha_\ell] \\
&= \alpha_\ell \cdot \mathrm{E}[Y_1\cdots Y_{\ell-1}] \ ,
\end{aligned}$$

and the claim follows by induction on ℓ. □

As noted above, it is always the case that $\mathrm{E}\left[X_i\right] < 32$, regardless of the instantiation of previous X_j's, for $j < i$. Note that Claim 3.5.23 also applies to computing the expectation of $\prod_{i\in J} X_i$, for any subset $J \subset [m]$, since any subset of the X_i's (preserving the right order) satisfy the condition of claim.

Once the m commitments $R^1[i]$ are complete, we can define a random variable $A = A(U)$ that denotes the set of values $a = (a_1,\ldots,a_m)$'s for which the sender S^* produces a valid opening with respect to $\mathcal{B}'$ in some continuation of the protocol. By the definition of $\mathcal{B}'$, this means that $a = (a_1,\ldots,a_m)$ is valid if at least $m-\beta$ of those are a_i's correspond to decommitments that are in $\mathcal{B}$. For those a_i's corresponding to decommitments that are in $\mathcal{B}$, the number of possible values that a_i can take on is $X_i(U)$. And for those a_i's correspond to decommitments that are not in $\mathcal{B}$, we can only bound the number of possible values that a_i can take on by 2^k (since a_i is a k-bit string).

$$\begin{aligned}
\mathop{\mathrm{E}}_{U}\left[|A(U)|\right] &\le \mathop{\mathrm{E}}_{U}\left[\sum_{J\subseteq[m],|J|\ge m-\beta}\ \prod_{i\in J} X_i(U) \prod_{i\notin J} 2^k\right] \\
&= \sum_{J\subseteq[m],|J|\ge m-\beta} \mathop{\mathrm{E}}_{U}\left[\prod_{i\in J} X_i(U) \prod_{i\notin J} 2^k\right] \\
&< \sum_{J\subseteq[m],|J|\ge m-\beta}\ \prod_{i\in J} 32 \cdot \prod_{i\notin J} 2^k && \text{(by Claim 3.5.23)} \\
&\le 2^m \cdot 32^{m-\beta} \cdot (2^k)^\beta && \text{(because } 32 < 2^k\text{)} \\
&\le 2^{(\beta+1)(k-1)+6m-k+1} = 2^{\alpha_1-(k-6m-4)} \ .
\end{aligned}$$

Let random variable $\mathit{\Gamma}_1 = (H_1, H_1(A))$. Since $\mathrm{E}[|A|] \le 2^{\alpha_1-(k-6m-4)}$ and the range of $h_1 \in \mathcal{H}_1$ is α_1, the expected density of $\mathit{\Gamma}_1$ satisfies $\mathrm{E}[\mu(\mathit{\Gamma}_1)] \le \mathrm{E}[|A|] \cdot 2^{-\alpha_1} \le 2^{-(k-6m-4)}$, where the expectation is taken over the coins tosses $U = (U_1, \ldots, U_m)$. Note that $\mathit{\Gamma}_1$ is independent of the coins of R^1_{IH} in the first phase interactive hashing (though not independent of the coins of $\mathbf{R}^1$).

Finally, we have

$$\mathop{\mathrm{E}}_{\text{coins } \mathbf{R}^1} \left[\left| \mathrm{openings}(S^*, \mathbf{R}^1)(\mathcal{B}') \right| \right] \le \mathop{\mathrm{E}}_{\text{coins } R^1_{\mathrm{IH}}, \mathit{\Gamma}_1} \left[\left| \{ d^{(1)} : C^{(1)}(d^{(1)}) \in \mathit{\Gamma}_1 \} \right| \right] \ ,$$

where in the second expectation, $C = \mathrm{output}(S^*, R^1_{\mathrm{IH}})$. By Lemma 3.5.10,

$$\mathop{\mathrm{E}}_{\text{coins } R^1_{\mathrm{IH}}, \mathit{\Gamma}_1} \left[\left| \{ d^{(1)} : C^{(1)}(d^{(1)}) \in \mathit{\Gamma}_1 \} \right| \right] < 24 + 2^{k'+1} \cdot \mathrm{E}[\mu(\mathit{\Gamma}_1)] < 32 \ ,$$

with the last inequality following from $k' < k - 8m - 8$.

Property (B.2). We use the same approach as above, except this time, we consider all deterministic S^*, as opposed to only those with the unique binding property. Also we need to fix a binding transcript $\tau = (\tau_1, \ldots, \tau_m) \in \mathcal{B}'$. Let J be the set of indices such that $\tau_i \in \mathcal{B}$.

As done previously, we define $S^*[i]$ and set $X_i = \left|\mathrm{openings}(S^*[i], R^2[i])(\tau_i)\right|$, where $S^*[i]$. By our induction hypothesis, for all $i \in J$, we have

$$\mathrm{E}\left[X_i |_{X_1 = x_1, \ldots, X_{i-1} = x_{i-1}} \right] < 32 \ ,$$

for any $(x_1, \ldots, x_{i-1}) \in \mathrm{Supp}(X_1, \ldots, X_{i-1})$.

Let random variable B denote the denotes the set of values $b = (b_1, \ldots, b_m)$ for which the sender S^* produces a valid opening in some continuation of the protocol. Noting that X_i can be as large as 2^k for indices $i \notin J$, we have

$$\begin{aligned}
\mathrm{E}\left[|B|\right] &\le \mathop{\mathrm{E}}_{\text{coins } R^2[1], \ldots, R^2[m]} \left[\prod_{i \in J} X_i \prod_{i \notin J} 2^k \right] \\
&< \prod_{i \in J} 32 \cdot \prod_{i \notin J} 2^k && \text{(by Claim 3.5.23))} \\
&\le 32^{\beta+1} \cdot (2^k)^{m-\beta-1} && \text{(because } 32 < 2^k\text{)} \\
&\le 2^{(m-\beta)(k-1)-(k-6m)} && \text{(because } m > 5\text{)} \\
&= 2^{\alpha_2-(k-6m-3)}.
\end{aligned}$$

Let random variable $\mathit{\Gamma}_2 = (H_2, H_2(B))$. Since $\mathrm{E}[|B|] \le 2^{\alpha_2-(k-6m-3)}$ and the range of $h_2 \in \mathcal{H}_2$ is α_2, the expected density of $\mathit{\Gamma}_2$ satisfies $\mathrm{E}[\mu(\mathit{\Gamma}_2)] \le \mathrm{E}[|B|] \cdot 2^{-\alpha_2} \le 2^{-(k-6m-3)}$, where the expectation is taken over the coins tosses of $R^2_1, \ldots, R^2_m$. Note that $\mathit{\Gamma}_2$ is independent of the coins of R_{IH}^2 in the second phase interactive hashing (though not independent of the coins of $\mathbf{R}^2$). Finally, we have

$$\mathop{\mathrm{E}}_{\text{coins } \mathbf{R}^2}\left[\left|\text{openings}(S^*, \mathbf{R}^2)(\tau')\right|\right] \leq \mathop{\mathrm{E}}_{\text{coins } R_{\mathrm{IH}}^2, \Gamma_2}\left[\left|\{d^{(2)} : C^{(2)}(d^{(2)}) \in \Gamma_2\}\right|\right] \ ,$$

where in the second expectation, $C = \text{openings}(S^*(\Gamma_2), R_{\mathrm{IH}})$. By Lemma 3.5.10,

$$\mathop{\mathrm{E}}_{\text{coins } R_{\mathrm{IH}}^2, \Gamma_2}\left[\left|\{d^{(2)} : C^{(2)}(d^{(2)}) \in \Gamma_2\}\right|\right] < 24 + 2^{k'+1} \cdot \mathrm{E}[\mu(\Gamma_2)] < 32 \ ,$$

with the last inequality following from $k' < k - 8m - 8$. □

LEMMA 3.5.24

(Final step binding preservation.) For some constant $D \in \mathbb{N}$ and any integers $t \in [1, n]$, $\beta_1, \ldots, \beta_\ell \in \{0, 1, \ldots, D-1\}$, and $\beta_{\ell+1} \in [0, n]$, letting (S, R) be the final output of Algorithm 3.5.17 with parameters D, t, and $(\beta_1, \ldots, \beta_{\ell+1})$, there exists a binding set $\mathcal{B}'$ such that the following two conditions hold:

(B.1) For every deterministic sender S^* with the *unique binding property*, with probability $1 - 2^{-\Omega(n)}$ over the coins of R^1,

$$\left|\text{openings}(S^*, \mathsf{R}^1)(\mathcal{B}')\right| \leq 1 \ .$$

(B.2) For every $\tau \in \mathcal{B}'$ and for every deterministic sender S^*, with probability $1 - 2^{-\Omega(n)}$ over the coins of R^2,

$$\left|\text{openings}(S^*, \mathsf{R}^2)(\tau)\right| \leq 1 \ .$$

Proof. From Lemma 3.5.22, we have scheme (S_ℓ, R_ℓ) with an associated binding set $\mathcal{B}$ satisfying both conditions (B.1) and (B.2) in Lemma 3.5.22. Scheme $(\mathsf{S}, \mathsf{R}) = \mathsf{Amplify}(S_\ell, R_\ell)$, and hence we will need to show that the amplification boosts the binding by making sure both $\left|\text{openings}(S^*, \mathsf{R}^1)(\mathcal{B})\right| \leq 1$ and $\left|\text{openings}(S^*, \mathsf{R}^2)(\tau)\right| \leq 1$ with probability $1 - 2^{-\Omega(n)}$.

Throughout this proof, the value of m will be fixed to n (as in Step 3 of Algorithm 3.5.17), although we will keep writing m. We define our new binding set $\mathcal{B}'$ for (S, R) in terms of $\mathcal{B}$ as follows:

$$\mathcal{B}' = \{(\tau_1, \ldots, \tau_m) : \exists\, j_1, \ldots, j_{\beta+1} \text{ such that } \tau_{j_1}, \ldots, \tau_{j_{\beta+1}} \in \mathcal{B}\} \ .$$

That is, a transcript $\tau' = (\tau_1, \ldots, \tau_m) \in \mathcal{B}'$ if and only if at least $\beta + 1$ of τ_j's are in $\mathcal{B}$. Conversely, $\tau' \notin \mathcal{B}'$ if and only if at least $m - \beta$ of the τ_j's are not in $\mathcal{B}$.

Property (B.1). Using the same analysis and notations as in the proof of Lemma 3.5.22, we have that

$$\mathop{\mathrm{E}}_{\text{coins } R^1[1],\cdots,R^1[m]}[|A|] \leq 2^m \cdot 32^{m-\beta} \cdot (2^k)^\beta \leq 2^{\beta k+6m} \ ,$$

where A is the random variable denoting the set of values $a = (a_1, \dots, a_m)$'s for which the sender S^* produces a valid opening with respect to $\mathcal{B}'$ in some continuation of the protocol.

Since $\delta = \Omega(1)$ and $k_\ell \geq \log n$, observe that $\alpha_1 = \lfloor(\beta + \frac{1}{3}\delta n)k\rfloor > \beta k + 8n$, for large enough values of n. Let random variable $\mathit{\Gamma}_1 = (H_1, H_1(A))$. Since the range of $h_1 \in \mathcal{H}_1$ is $\{0,1\}^{\alpha_1}$, the density of $\mathit{\Gamma}_1$ satisfies

$$\mathop{\mathrm{E}}_{\text{coins } R^1[1],\cdots,R^1[m]}[\mu(\mathit{\Gamma}_1)] \leq \mathrm{E}[|A|] \cdot 2^{-\alpha_1} < 2^{\beta k+6m} \cdot 2^{-(\beta k+8n)} = 2^{-2n} \ ,$$

since $m = n$. Thus, with probability at least $1-2^{-n}$ over the coins tosses of $R^1[1], \dots, R^1[m]$, we have that

$$\mu(\mathit{\Gamma}_1) \leq 2^{-2n} \cdot 2^n = 2^{-n} \ .$$

By Lemma 3.2.5, we can conclude that for such a $\mathit{\Gamma}_1$ (with $\mu(\mathit{\Gamma}_1) \leq 2^{-n}$),

$$\Pr_{\text{coins } R^1_{\mathrm{IH}}}\left[\left|\{d^{(1)} : C^{(1)}(d^{(1)}) \in \mathit{\Gamma}_1\}\right| > 1\right] = 2^{-\Omega(n)} \ .$$

Finally, we have:

$$\begin{aligned}
&\Pr_{\text{coins } \mathbf{R}^1}\left[\left|\text{openings}(S^*, \mathbf{R}^1)\right| > 1\right] \\
&\leq \Pr_{\text{coins } R^1_1,\cdots,R^1_m}\left[\mu(\mathit{\Gamma}_1) > 2^{-n}\right] + \Pr_{\text{coins } R^1_{\mathrm{IH}}}\left[|\{d^{(1)} : C^{(1)}(d^{(1)}) \in \mathit{\Gamma}_1\}| > 1 \;\middle|\; \mu(\mathit{\Gamma}_1) \leq 2^{-n}\right] \\
&= 2^{-\Omega(n)} \ .
\end{aligned}$$

Property (B.2). Fix any $\tau' \in \mathcal{B}'$. Again, we use the same analysis and notations as in the proof of Lemma 3.5.22 to get:

$$\mathop{\mathrm{E}}_{\text{coins } R^2[1],\cdots,R^2[m]}[|B|] \leq 32^{\beta+1} \cdot (2^k)^{m-\beta-1} \leq 2^{(m-\beta)k+5m} \ ,$$

where B is the random variable denoting the set of values $b = (b_1, \dots, b_m)$'s for which the sender S^* produces a valid opening in some continuation of the protocol

Since $\delta = \Omega(1)$ and $k \geq \log n$, observe that $\alpha_2 = \lfloor(n - \beta + \frac{1}{3}\delta n)k\rfloor > (n-\beta)k + 7n$, for large enough values of n. Let random variable $\mathit{\Gamma}_2 = (H_2, H_2(B))$. Since the range of $h_2 \in \mathcal{H}_2$ is $\{0,1\}^{\alpha_2}$, the density of $\mathit{\Gamma}_2$ satisfies

$$\mathop{\mathrm{E}}_{\text{coins } R^2[1],\cdots,R^2[m]}[\mu(\mathit{\Gamma}_2)] \leq \mathrm{E}[|B|] \cdot 2^{-\alpha_2} < 2^{(m-\beta)k+5m} \cdot 2^{-((n-\beta)k+7n)} = 2^{-2n} \ ,$$

since $m = n$. Thus, with probability at least $1-2^{-n}$ over the coins tosses of $R^2[1], \dots, R^2[m]$, we have that

$$\mu(\Gamma_2) \leq 2^{-2n} \cdot 2^n = 2^{-n}.$$

By Lemma 3.2.5, we can conclude that for such a Γ_2 (with $\mu(\Gamma_2) \leq 2^{-n}$),

$$\Pr_{\text{coins } R^2_{\mathrm{IH}}} \left[\left| \{ d^{(2)} : C^{(2)}(d^{(2)}) \in \Gamma_2 \} \right| > 1 \right] = 2^{-\Omega(n)} \ .$$

Finally, we have:

$$\begin{aligned} &\Pr_{\text{coins } \mathbf{R}^2} \left[\left| \text{openings}(S^*, \mathbf{R}^2)(\tau') \right| > 1 \right] \\ &\leq \Pr_{\text{coins } R^2_1, \cdots, R^2_n} \left[\mu(\Gamma_2) > 2^{-n} \right] + \Pr_{\text{coins } R^2_{\mathrm{IH}}} \left[\left| \{ d^{(2)} : C^{(2)}(d^{(2)}) \in \Gamma_2 \} \right| \ \middle| \ \mu(\Gamma_2) \leq 2^{-n} \right] \\ &= 2^{-\Omega(n)} \ . \end{aligned}$$

□

3.5.4 A collection of 1-out-of-2-binding commitments

In this section, we prove Theorem 3.5.1 restated below.

RESTATEMENT OF THEOREM 3.5.1

If one-way functions exist, then on security parameter 1^n, we can construct in time polynomial in n a collection of public-coin 2-phase commitment schemes $\mathcal{COM} = \{\mathsf{Com}_1, \cdots, \mathsf{Com}_m\}$, where $m = \mathrm{poly}(n)$, such that:

- There exists an index $i \in \{1, 2, \ldots, m\}$ such that scheme Com_i is statistically hiding.
- For every index $i \in \{1, 2, \ldots, m\}$, scheme Com_i is computationally 1-out-of-2 binding.

Proof of Theorem 3.5.1

To obtain the desired collection of 2-phase commitment schemes, we apply Algorithm 3.5.17 to the weakly-hiding scheme (S_0, R_0), which can be constructed based on any one-way function. More precisely, we obtain a collection of commitments by enumerating over all the polynomially many choices of the integers $t \in \{1, 2, \ldots, n\}$, $\beta_1, \ldots, \beta_\ell \in \{0, 1, \ldots, D-1\}$, and $\beta_{\ell+1} \in \{0, 1, \ldots, n\}$. Note that the number of choices is $n \cdot D^\ell \cdot (n+1) = \mathrm{poly}(n)$, as $D = O(1)$ and $\ell = \log n$. By Lemma 3.5.18, the resulting commitment schemes $\mathsf{Com}_1, \cdots, \mathsf{Com}_m$ all run in polynomial time. The hiding and binding properties of these schemes are given by Lemmas 3.5.25 and 3.5.26, which together establish Theorem 3.5.1.

LEMMA 3.5.25

There exists a constant $D \in \mathbb{N}$, integers $t \in \{1, 2, \ldots, n\}$, $\beta_1, \ldots, \beta_\ell \in \{0, 1, \ldots, D-1\}$, and $\beta_{\ell+1} \in \{0, 1, \ldots, n\}$ such that the 2-phase commitment scheme (S, R) produced by Algorithm 3.5.17 with parameters D, t, and $(\beta_1, \ldots, \beta_{\ell+1})$ is statistically hiding in the sense

Definition 3.4.3 (regardless of whether the function f on which the scheme is based on is one-way or not).

Proof. We prove by induction on the properties of (S_j, R_j) for $j = 0, 1, \dots, \ell$. The induction hypothesis is that (S_j, R_j) has two associated sets $\Gamma_{1,j}, \Gamma_{2,j} \subseteq \{0,1\}^{nm^j}$ such that for all R^*, the following holds:

1. $\Gamma_{1,j} \cup \Gamma_{2,j} = \{0,1\}^{nm^j}$ and $\mu(\Gamma_{1,j} \cap \Gamma_{2,j}) \geq \min\{2^j/n, 1/2D\}$.
2. $\mathrm{CP}^{1/2}(A|V) \leq \sqrt{2^{-(k_j-1)}}$, where $A = \mathrm{output}_S(S^1_{c,j}(\mathit{\Gamma}_{1,j}), R^*)$ and $V = \mathrm{view}_{R^*}(S^1_{c,j}(\mathit{\Gamma}_{1,j}), R^*)$.
3. $\mathrm{CP}^{1/2}(B_\tau|W_\tau) \leq \sqrt{2^{-(k-1)}}$, where the joint distribution $(B_\tau, W_\tau) = (\mathrm{output}_S(S^2_c(\mathit{\Gamma}_{2,j}), R^*), \mathrm{view}_{R^*}(S^2_c(\mathit{\Gamma}_{2,j}), R^*))|_{\mathrm{T}=\tau}$, for every $\tau \in \mathrm{Supp}(\mathrm{T})$, for $\mathrm{T} = \mathrm{transcript}(S^1(\mathit{\Gamma}_{2,j}), R^*)$.

where k_j is defined as in Algorithm 3.5.17.

The base case of $j = 0$ follows from the fact that Protocol 3.4.6 is $(1/n)$-hiding as established by Lemma 3.5.13. The induction step is provided by the Intermediate Step Hiding Amplification Lemma 3.5.19. Finally, observe that $\mu(\Gamma_{1,\ell} \cap \Gamma_{2,\ell}) \geq \min\{2^\ell/n, 1/(2D)\} = \Omega(1)$ since $\ell = \log n$.

By the Final Step Hiding Amplification Lemma 3.5.20, there exists two sets $\Gamma_{1,\ell+1}$ and $\Gamma_{2,\ell+1}$ such that for all R^*, the following three conditions holds:

1. $\mu(\Gamma_{1,\ell+1}), \mu(\Gamma_{2,\ell+1}) > 1 - 2^{-\Omega(n)}$;
2. (A, V) is $2^{-\Omega(n)}$-close to (U_1, V), where $A = \mathrm{output}_{\mathsf{S}}(\mathsf{S}^1_c(\mathit{\Gamma}_{1,\ell+1}), R^*)$ and $V = \mathrm{view}_{R^*}(\mathsf{S}^1_c(\mathit{\Gamma}_{1,\ell+1}), R^*)$;
3. for all $\tau' \in \mathrm{Supp}(\mathrm{T}')$, $(B'_{\tau'}, W'_{\tau'})$ is $2^{-\Omega(n)}$-close to $(U_1, W'_{\tau'})$, where $(B'_{\tau'}, W'_{\tau'}) = (\mathrm{output}_{\mathsf{S}}(\mathsf{S}^2_c(\mathit{\Gamma}_{2,\ell+1}), R^*), \mathrm{view}_{R^*}(\mathsf{S}^2_c(\mathit{\Gamma}_{2,\ell+1}), R^*))|_{\mathrm{T}'=\tau'}$, and $\mathrm{T}' = \mathrm{transcript}(\mathsf{S}^1(\mathit{\Gamma}_{2,\ell+1}), R^*)$.

Since both $\mu(\Gamma_{1,\ell+1}), \mu(\Gamma_{2,\ell+1}) > 1 - 2^{-\Omega(n)}$, we can substitute random variables $\mathit{\Gamma}_{1,\ell+1}$ and $\mathit{\Gamma}_{2,\ell+1}$ with an independent uniform random variable U_N, where $N = nm^\ell$and get the following desired hiding properties.

- (A, V) is $2^{-\Omega(n)}$-close to (U_1, V), where $A = \mathrm{output}_{\mathsf{S}}(\mathsf{S}^1_c(U_N), R^*)$ and
  $V = \mathrm{view}_{R^*}(\mathsf{S}^1_c(U_N), R^*)$.
- (B', W', T') is $2^{-\Omega(n)}$-close to (U_1, W', T'), where $B' = \mathrm{output}_{\mathsf{S}}(\mathsf{S}^2_c(U_N), R^*)$, $W' = \mathrm{view}_{R^*}(\mathsf{S}^2_c(U_N))$, and $\mathrm{T}' = \mathrm{transcript}(\mathsf{S}^1(U_N), R^*)$.

The above two conditions are the requirements for being statistical hiding in the sense Definition 3.4.3. □

LEMMA 3.5.26

There exists a constant $D \in \mathbb{N}$ such that for all integers $t \in \{1, 2, \dots, n\}$, $\beta_1, \dots, \beta_\ell \in \{0, 1, \dots, D-1\}$, and $\beta_{\ell+1} \in \{0, 1, \dots, n\}$, the 2-phase commitment scheme (S, R) produced by Algorithm 3.5.17 with parameters D, t, and $(\beta_1, \dots, \beta_{\ell+1})$ is computationally 1-out-of-2 binding in the sense of Definition 3.4.4. (Here the function f for which the scheme is based on needs to be hard to invert.)

Proof. By Lemma 3.5.24, we have established that the 2-phase commitment scheme (S, R) produced by Algorithm 3.5.17 satisfies the first condition of Definition 3.4.4. In addition, it also satisfies the second condition for all S^* with the *unique binding* property. Stated formally, for every deterministic (and computationally unbounded) S^* with the unique binding property,

$$\Pr\left[\left|\text{openings}(S^*, \mathsf{R}^1)\right| \leq 1\right] = 1 - 2^{-\Omega(n)}, \tag{3.8}$$

where the probability is taken over the coins of R^1.

Thus, it suffices to prove is that any PPT S^* breaking the second condition of Definition 3.4.4 with probability ε will either (i) yield a *PPT* $\hat{S}$ that violates the computational 1-out-of-2 binding property of (S_0, R_0) with probability at least $\varepsilon^{O(1)}/\operatorname{poly}(n)$, or (ii) yield a computationally unbounded $\hat{S}$ that has the unique binding property and succeeds with probability greater than $\varepsilon/2$. In both cases, ε needs to be negligibly small in order to avoid a contradiction. Without loss of generality, we may assume adversarial PPT sender S^* to be deterministic since we can set its coins to maximizes its success probability.

From now on, let ε be the probability that the deterministic S^* breaks the second condition of Definition 3.4.4 with respect to scheme (S, R). By the way we defined (S, R), it contains polynomially many executions of (S_0, R_0). Let $N = n \cdot D^\ell$ denote such number.

Let $\mathbf{z}$ denote the transcript of (S^*, R). Contained in $\mathbf{z}$ is also a first-phase commitment $z[i]$ for the i-th execution of R_0, denoted $R_0[i]$ (for all $i = 1, 2, \dots, N$). Let $\hat{z}[i]$ be the partial transcript of $\mathbf{z}$ up to and including the first commit stage of $R_0[i]$. Note that $z[i]$ is a suffix of $\hat{z}[i]$, and $\hat{z}[i]$ is a prefix of $\mathbf{z}$.

For all index $i \in [N]$, partial transcripts $\hat{z}[i]$ ending with the first commit stage of $R_0[i]$ and $d \in \{0,1\}^{k_0}$, define

$$p_{i,\hat{z}[i],d} = \Pr_{\mathbf{z} \leftarrow (S^*, \mathsf{R}^1)} \left[\mathbf{z} \text{ contains a valid opening of } z[i] \text{ to value } d \,|\, \mathbf{z} \text{ begins with } \hat{z}[i]\right] \ ,$$

where as usual by a valid opening, we mean that the transcript $\tau[i]$ of S^*'s interaction with $R_0[i]$ contains an opening of $z[i]$ to the value d, the first phase of $\tau[i]$ is not in the binding set $\mathcal{B}_0$, and $R_0[i]$ accepts in both phases of $\tau[i]$.

Let $K = 2^{k_0}$, where k_0 is the message length in (S_0, R_0). We have two cases to consider.

Case 1. There exists an $i \in [N]$ such that with probability at least $\frac{\varepsilon}{4NK}$ over $\hat{z}[i]$, there exists $d \neq d'$ with both $p_{i,\hat{z}[i],d}, p_{i,\hat{z}[i],d'} > \frac{\varepsilon}{4NK}$.

In this case, we violate the computational 1-out-of-2 binding property of (S_0, R_0) by considering the following sender $\hat{S}$ interacting with $R_0[i]$.

1. Select a random $i \leftarrow [N]$.
2. Run S^* with R^1, simulating all of the messages of R^1 internally except for those of $R_0[i]$. Halting after the first commit stage of $R_0[i]$, we obtain a partial transcript $\hat{z}[i]$. From $\hat{z}[i]$, we get $z[i]$, the first-phase commitment of $R_0[i]$.
3. Record the current state ψ of S^* and R^1.
4. Continue the execution of S^* with R^1 from ψ to obtain a decommitment to a value d in the interaction with $R_0[i]$.
5. Repeat Step 4 with independent randomness in continuing the execution of S^* with R^1 to obtain a decommitment to a value d'. (This can be done since R is public coin, i.e., just sends independent random coins at each round, and S^* is deterministic.)

Because our goal is to violate the computational 1-out-of-2 binding property of (S_0, R_0), we succeed in the above algorithm if $d \neq d'$ and decommitments produced are valid. We calculate our success probability as follows: We guess correct index $i \in [N]$ with probability $1/N$. Given that we guess the correct i, we get the desired $\hat{z}[i]$ with probability at least $\frac{\varepsilon}{4NK}$. Now, when we do two independent continuations of $\hat{z}[i]$ we arrive at two different decommitted values with probability greater than $(\frac{\varepsilon}{4NK})^2$. Consequently, we violate the computational 1-out-of-2 binding property of (S_0, R_0) (i.e., win the game in Condition 2 of Definition 3.4.4) with probability greater than

$$\frac{1}{N} \cdot \frac{\varepsilon}{4NK} \cdot \left(\frac{\varepsilon}{4NK}\right)^2 = \frac{1}{N} \cdot \left(\frac{\varepsilon}{4NK}\right)^3 = \left(\frac{\varepsilon}{n}\right)^{O(1)} ,$$

since $K = 2^{k_0} = 2^{O(\log n)} = \mathrm{poly}(n)$ and $N = n \cdot D^\ell = n \cdot O(1)^{O(\log n)} = \mathrm{poly}(n)$. This forces ε to be a negligible function.

Case 2. For all $i \in [N]$, with probability greater than $1 - \frac{\varepsilon}{4NK}$ over $\hat{z}[i]$, there is at most one d such that $p_{i,\hat{z}[i],d} > \frac{\varepsilon}{4NK}$.

Define $d^*(\hat{z}[i])$ to be the value of d that maximizes $p_{i,\hat{z}[i],d}$. Taking a union bound over all the rest of the $p_{i,\hat{z}[i],d'} < \frac{\varepsilon}{4NK}$, we have that

$$\begin{aligned}
&\Pr_{\mathbf{z}\leftarrow(S^*,\mathsf{R})}\left[S^* \text{ opens some } z[i] \text{ to a value other than } d^*(\hat{z}[i])\right] \\
&\leq \sum_{i=1}^{N}\left(\frac{\varepsilon}{4NK}\cdot K + \Pr_{\hat{z}[i]}\left[\text{exists more than one } d \text{ such that } p_{i,\hat{z}[i],d} > \frac{\varepsilon}{4NK}\right]\right) \\
&< N\cdot\left(\frac{\varepsilon}{4NK}\cdot K + \frac{\varepsilon}{4NK}\right) \\
&< \frac{\varepsilon}{2}\ .
\end{aligned}$$

Let $\hat{S}$ be the adversary that mimics S^* except that it halts and fails if S^* attempts to open some $z[i]$ to a value other than $d^*(\hat{z}[i])$, for all $i \in [N]$ and all $\hat{z}[i]$. By the way we defined $\hat{S}$, the final outcome of $(\hat{S}, \mathsf{R}^1)$ will only differ with the original final outcome of (S^*, R^1) with probability at most $\varepsilon/2$ over the coins of R^1. In addition, $\hat{S}$ has the unique binding property. By (3.8) above, $\left|\text{openings}(\hat{S}, \mathsf{R}^1)\right| > 1$ occurs with at most negligible probability over the coins of R^1. Hence, $\left|\text{openings}(S^*, \mathsf{R}^1)\right| > 1$ occurs with probability at most $\text{neg}(n) + \varepsilon/2$. We started off assuming that S^* breaks property (B.1) of scheme (S, R) with probability at least ε, that is to say $\left|\text{openings}(S^*, \mathsf{R}^1)\right| > 1$ with probability at least ε. Thus $\varepsilon \leq \text{neg}(n) + \varepsilon/2$, which implies that $\varepsilon = \text{neg}(n)$. □

3.5.5 Standard commitments from 1-out-of-2-binding commitments

In the previous sections, we constructed statistically-hiding and computationally *1-out-of-2*-binding commitment schemes from any one-way function. In this section, we present a result of Haitner and Reingold [HR2] that transforms these 1-out-of-2-binding commitments into commitment schemes that are statistically hiding and computationally binding (in the standard sense of binding). They accomplished this using a novel application of a *universal one-way hash family*, whose existence can be based on any one-way function [Rom] (see also [KK]). Thus, the Haitner & Reingold transformation can be based on any one-way function. We state their transformation techniques in Algorithm 3.5.28 and Protocol 3.5.29, but first we give an overview of their techniques.

Overview of the Haitner & Reingold transformation

The 1-out-of-2 binding property of 2-phase commitment schemes states that it is infeasible for an adversarial sender S^* to break both phases of the commitment, but nonetheless it might be possible for S^* to break one of the two phases of its choice. With this in mind, suppose that after the first commitment phase, receiver R flips a coin $\mathit{phase} \leftarrow \{1, 2\}$. If $\mathit{phase} = 1$, the first commitment phase is used to do the commitment. On the other hand, if $\mathit{phase} = 2$, the second commitment phase is used to do the commitment (this is done by S^* revealing its first-phase commitment, and then proceeding to the second phase with R). Intuitively, this would make the scheme binding (with probability 1/2) if S^* chooses which of the two phases it wants to break in advance. The problem, however, is that S^* could choose the phase that it wants to break after seeing the value of *phase*.

A way to overcome this problem is to force the adversary S^* to decide which of the two phases it wants to break before seeing the value of *phase*. Haitner and Reingold [HR2] achieved this by having S^* send back a value $y = f(\sigma)$ before the value of *phase* is announced by the receiver R, where σ is the message committed to by S^* in the first phase, and f is a random hash function from a universal one-way hash family. A ***universal one-way hash family*** is a family of hash functions such that it is hard to find collisions with any particular value of x specified in advance.[12] In other words, for a value of x announced before a random hash function f is selected from that family, any efficient algorithm will not be able to find another x' such that $f(x') = f(x)$. This property of a universal one-way hash family is termed ***target collision resistance*** by Bellare and Rogaway [BR].

We first argue the hiding property of this new scheme. Before y is sent, the value of σ, the message committed in the first phase, is hidden. If hash function f is *compressing enough*, then the value of $y = f(\sigma)$ leaks at most a few bits of *information* about σ, so the *entropy* of σ given y is still large. This means that we can apply a pairwise-independent hash on σ to get an almost uniform value (recall the Leftover Hash Lemma 3.3.1). Thus, this new scheme is hiding when $phase = 1$. When $phase = 2$, the sender reveals σ and proceeds on to the second phase, which is used for the commitment. In this case, the hiding property of this new scheme follows from the hiding property of the second commitment phase.

Next, we argue the binding property of this new scheme by making the following observation: the 1-out-of-2 binding property says that after the first commitment phase, there exists at most one value of σ^* that allows an adversarial sender S^* to cheat in the second phase. In other words, if S^* reveals to a value other than σ^*, the second phase will be binding.

When it is the sender's turn to send y, after receiving a random hash function f from receiver R, sender S^* could decide to either send $y = f(\sigma^*)$ or send $y \neq f(\sigma^*)$. If it decides to send $y = f(\sigma^*)$, and if R selects $phase = 1$ following that, then S^* is bound to a single value, since to decommit to two different values it will have to reveal a $\sigma' \neq \sigma^*$ with $f(\sigma') = y = f(\sigma^*)$, and this is infeasible by the target collision resistance property of f. (The value of σ^* is determined by the first-phase commitment, which is completed before a random f is selected.) Instead if it decides to send $y \neq f(\sigma^*)$, and if R selects $phase = 2$ following that, then S^* will have to reveal to a value other than σ^* for its first-phase commitment. In this case, the commitments are done in the second phase, and by the 1-out-of-2 binding property, this phase is guaranteed to be binding. Since the value of *phase* is independent of y, both cases happen with probability 1/2, which would make our scheme binding with probability close to 1/2.

[12] See Definition 3.5.31 for the definition of a *universal one-way hash family*.

The Haitner & Reingold transformation

The Haitner & Reingold transformation [HR2], to be stated in Algorithm 3.5.28, requires a 2-phase commitment with message lengths $(k_1, k_2) = (n, 1)$: that is to say, the first phase deals with commitment to an n-bit message, and the second phase deals with commitment to a 1-bit message. Next, we show how to obtain these commitments.

Obtaining 2-phase commitments with suitable message lengths. The 2-phase commitments that we constructed based on any one-way function, as stated in Theorem 3.5.1, has message lengths $(k_1, k_2) = (1, 1)$. Nevertheless, it is possible to convert these scheme to one with message lengths $(k_1, k_2) = (n, 1)$.

CLAIM 3.5.27

(From [HR2, Lem. 2.14].) If there exists a 2-phase commitment scheme (S, R) with message lengths $(k_1, k_2) = (1, 1)$, then there exists another 2-phase commitment scheme $(\mathsf{S}', \mathsf{R}')$ with message lengths $(k_1, k_2) = (n, 1)$ such that:

- if (S, R) is statistically hiding, then $(\mathsf{S}', \mathsf{R}')$ is also statistically hiding, and
- if (S, R) is statistically [resp., computationally] 1-out-of-2 binding, then $(\mathsf{S}', \mathsf{R}')$ is also statistically [resp., computationally] 1-out-of-2 binding.

Proof Sketch. Here, we present an informal description of the new scheme $(\mathsf{S}', \mathsf{R}')$ with message lengths $(k_1, k_2) = (n, 1)$; the detailed construction of $(\mathsf{S}', \mathsf{R}')$, and the proof of Claim 3.5.27 is in [HR2]:

1. In the first phase, scheme $(\mathsf{S}', \mathsf{R}')$ commits to a message $\sigma = (\sigma_1, \ldots, \sigma_n) \in \{0,1\}^n$ by running n independent executions of the first phase of (S, R) in parallel, committing to σ_i in the i-th execution of (S, R).

2. In the second phase, scheme $(\mathsf{S}', \mathsf{R}')$ commits to a bit $b \in \{0,1\}$ by also running n independent executions of the second phase of (S, R) in parallel, but this time committing to the same bit b in all executions of (S, R).

Intuitively, scheme $(\mathsf{S}', \mathsf{R}')$ should be hiding since it is only running multiple executions of scheme (S, R), all of which are hiding. The 1-out-of-2 binding property of $(\mathsf{S}', \mathsf{R}')$ can be informally argued as follows: If we break the first phase of $(\mathsf{S}', \mathsf{R}')$, we will have to break the i-th first phase execution of (S, R), for some $i \in \{1, 2, \ldots, n\}$. By the 1-out-of-2 binding property of (S, R), the i-th execution of the second phase of (S, R) will be binding, and because we are committing to the same bit b in all executions, this will force the second phase of $(\mathsf{S}', \mathsf{R}')$ to be binding.

The Haitner & Reingold transformation algorithm. We present the transformation algorithm using an arbitrary family of functions $\mathcal{F}$, and will only require $\mathcal{F}$ to be a universal one-way hash family when we want to prove the hiding and binding security properties. Separating the security properties from the protocol description enables us to prove that the transformation also works in settings of *instance-dependent* cryptographic primitives, as later considered in Section 3.6.

ALGORITHM 3.5.28

The Haitner & Reingold transformation, denoted as $\mathsf{HR\text{-}Transform}$.

Input: 2-phase commitment scheme (S, R) with message lengths $(k_1, k_2) = (n, 1)$, and a family of functions $\mathcal{F} = \bigcup_n \mathcal{F}_n = \{f \colon \{0,1\}^n \to \{0,1\}^m\}$.

Output: Commitment scheme $(\mathbb{S}, \mathbb{R})$ as described by Protocol 3.5.29.

Hence, we write the commitment scheme obtained as $(\mathbb{S}, \mathbb{R}) = \mathsf{HR\text{-}Transform}((\mathsf{S}, \mathsf{R}), \mathcal{F})$.

PROTOCOL 3.5.29

Standard commitment scheme $(\mathbb{S}, \mathbb{R})$ from 2-phase commitment scheme (S, R).

Security parameter: 1^n, given as common input to both $\mathbb{S}$ and $\mathbb{R}$.

Sender's private input: Bit $b \in \{0,1\}$.

Commit stage:

1. $\mathbb{S}$ selects a uniform $\sigma \leftarrow \{0,1\}^n$.
2. $\mathbb{S}$ and $\mathbb{R}$ engages in $(\mathsf{S}_c^1(\sigma), \mathsf{R}_c^1)(1^n)$, with $\mathbb{S}$ acting as S_c^1 and $\mathbb{R}$ acting as R_c^1. Let $c^{(1)}$ be the common output of S_c^1 and R_c^1 after the interaction.
3. $\mathbb{R}$ chooses $f \leftarrow \mathcal{F}_n$ and sends it to $\mathbb{S}$.
4. $\mathbb{S}$ sends $y = f(\sigma)$ to $\mathbb{R}$.
5. $\mathbb{R}$ flips a random coin, represented by $\mathit{phase} \leftarrow \{1,2\}$.

   If $\mathit{phase} = 1$, then proceed as follows:

   (a) $\mathbb{S}$ selects a random hash $h \leftarrow \mathcal{H}$, where $\mathcal{H}$ is a family of pairwise-independent hash functions with domain $\{0,1\}^n$ and range $\{0,1\}$, and sends $(h, b \oplus h(\sigma))$ to $\mathbb{R}$.

   (b) $\mathbb{S}$ and $\mathbb{R}$ both output $(c^{(1)}, f, y, \mathit{phase} = 1, h, b \oplus h(\sigma))$ as the commitment.

   If $\mathit{phase} = 2$, then proceed as follows:

   (a) $\mathbb{S}$ runs S_r^1 to obtain the decommitment message $\gamma^{(1)}$ and first-phase transcript τ corresponding to both σ and $c^{(1)}$. $\mathbb{S}$ sends $(\sigma, \gamma^{(1)}, \tau)$ to $\mathbb{R}$.

   (b) $\mathbb{S}$ and $\mathbb{R}$ engage in $(\mathsf{S}_c^2(b), \mathsf{R}_c^2)(1^n, \tau)$, with $\mathbb{S}$ acting as S_c^2 and $\mathbb{R}$ acting as R_c^2. Let $c^{(2)}$ be the common output of S_c^2 and R_c^2 after the interaction.

(c) $\mathbb{S}$ and $\mathbb{R}$ both output $(c^{(1)}, f, y, \mathit{phase} = 2, c^{(2)})$ as the commitment.

Reveal stage:

To decommit to bit b, do the following depending the value of *phase*.

If $\mathit{phase} = 1$, then:

1. $\mathbb{S}$ sends (b, σ) to $\mathbb{R}$;
2. If $y = f(\sigma)$ and the last component of the commitment equals $b \oplus h(\sigma)$, then $\mathbb{R}$ *accepts*. Otherwise, $\mathbb{R}$ *rejects*.

If $\mathit{phase} = 2$, then:

1. $\mathbb{S}$ runs S_r^2 to obtain the decommitment message $\gamma^{(2)}$, and sends $(b, \gamma^{(2)})$ to $\mathbb{R}$;
2. If $y = f(\sigma)$ and both R_r^1 and R_r^2 accept $(c^{(1)}, \sigma, \gamma^{(1)})$ and $(c^{(2)}, b, \gamma^{(2)})$, respectively, then $\mathbb{R}$ *accepts*. Otherwise, $\mathbb{R}$ *rejects*.

Analyzing the Haitner & Reingold transformation

The hiding and binding security properties of Protocol 3.5.29 will rely on properties of $\mathcal{F}$ being a universal one-way hash family. In fact, we will analyze these security properties separately so that Protocol 3.5.29 will be applicable in settings of *instance-dependent* cryptographic primitives, as later considered in Section 3.6.

Our plan for the remaining of this section is as follows: (i) we present the definition of a universal one-way hash family due to Naor and Yung [NY]; (ii) we separate the properties of a universal one-way hash family into two parts; and finally, (iii) we prove the hiding and binding properties of Protocol 3.5.29 based on these two separate properties.

Universal one-way hash family. In order to define a universal one-way hash family, we need to understand what it means for a family of functions to be *polynomial-time computable.*

DEFINITION 3.5.30

A family of functions $\mathcal{F} = \bigcup_n \mathcal{F}_n = \{f \colon \{0,1\}^n \to \{0,1\}^m\}$ is ***polynomial-time computable*** if

- for every n, the description of a function $f \in \mathcal{F}_n$ is bounded by a polynomial in n, and
- there exists a deterministic polynomial-time algorithm F such that for every n and every $f \in \mathcal{F}_n$, given the description of the function f and a string $x \in \{0,1\}^n$, F outputs the value of $f(x)$.

DEFINITION 3.5.31

A polynomial-time computable family of functions $\mathcal{F} = \bigcup_n \mathcal{F}_n = \{f\colon \{0,1\}^n \to \{0,1\}^m\}$ is a ***universal one-way hash family*** if $m < n$ and there exists a negligible function ε such that for all nonuniform PPT A the following holds for all values of n and all $x^* \in \{0,1\}^n$:

$$\Pr_{f \leftarrow \mathcal{F}_n}\left[A(1^n, f) = x \text{ such that } f(x) = f(x^*) \text{ and } x \neq x^*\right] \leq \varepsilon(n) \ .$$

REMARK 3.5.32

- The original definition by Naor and Yung [NY] required the PPT adversary A to output a string x^* before a random hash $f \leftarrow \mathcal{F}_n$ is chosen. The definition presented above, which considers all strings x^*, is equivalent to the Naor–Yung definition when considering *nonuniform* PPT adversaries. This is because this adversary can be given as advice the value of x^* that maximizes its chance to produce a collision.

- Although it is more natural for the security be parameterized in terms of the output length, namely m, our applications do not require hash functions that are shrinking by more than a polynomial factor. Hence for this reason, and in part for consistency, we keep n as our security parameter.

- Naor and Yung [NY] showed that starting with a universal one-way hash family that is compressing by only one bit, namely $m = n - 1$, more compression can be achieved, say $m \leq n/2$, by iterative application several hash functions chosen from the family. Hence, without loss of generality, we can assume that our universal one-way hash family will have the feature that $m \leq n/2$.

Two properties of a universal one-way hash family. A universal one-way hash family satisfying Definition 3.5.31 has the following two main properties.

Large preimages: most of the preimages have a large size. This follows from the compressing nature of hash functions: the output length m is much shorter than the input length n. (Recall that we can get a universal one-way hash family with $m \leq n/2$.) We formalize this in property in Definition 3.5.33.

Target collision resistance: it is hard to find collisions for a pre-specified value of x^*. We formalize this in property in Definition 3.5.34.

DEFINITION 3.5.33

A family of functions $\mathcal{F} = \bigcup_n \mathcal{F}_n = \{f\colon \{0,1\}^n \to \{0,1\}^m\}$ has the ***large preimages*** property if for every $f \in \mathcal{F}$, most elements in the range of f have large preimage sizes. Stated precisely, there exists a function $\alpha(n) = \omega(1)$ and a negligible function ε, such that for all values of n, the following holds:

$$\Pr_{x \leftarrow \{0,1\}^n}\left[\left|f^{-1}(f(x))\right| \geq n^{\alpha(n)}\right] \geq 1 - \varepsilon(n) \ ,$$

for every function $f \in \mathcal{F}_n$.

DEFINITION 3.5.34

A family of functions $\mathcal{F} = \bigcup_n \mathcal{F}_n = \{f\colon \{0,1\}^n \to \{0,1\}^m\}$ has the ***statistical [resp., computational] target collision resistance*** property if there exists a negligible function ε such that for every [resp., nonuniform PPT] A, the following holds for all values of n and every $x^* \in \{0,1\}^n$:

$$\Pr_{f \leftarrow \mathcal{F}_n}\left[A(1^n, f) = x \text{ such that } f(x) = f(x^*) \text{ and } x \neq x^*\right] \leq \varepsilon(n) \ .$$

Large preimages and target collision resistance are opposing properties. Specifically, it is impossible for a *single* family of functions to have large preimages and have *statistical* target collision resistance.[13] The power of a universal one-way hash family comes from the fact that it has the large preimages property and has *computational* target collision resistance.

LEMMA 3.5.35

If $\mathcal{F} = \bigcup_n \mathcal{F}_n = \{f\colon \{0,1\}^n \to \{0,1\}^m\}$, for $m \leq n/2$, is a universal one-way hash family, then $\mathcal{F}$ has both the large preimages and the *computational* target collision resistance properties.

Proof. The computational target collision resistance property follow directly from Definition 3.5.31. Hence, all we need to show is that the compressing nature of $\mathcal{F}$, when $m \leq n/2$, implies the large preimages property.

Group the elements with small preimages into a set $S = \{y \in \{0,1\}^m : \left|f^{-1}(y)\right| < 2^{\frac{3}{4}n-m}\}$. Since $m \leq n/2$, every element $y \notin S$ has a preimage of size $\left|f^{-1}(y)\right| \geq 2^{\frac{3}{4}n-m} \geq 2^{n/4} = n^{\omega(1)}$. To complete, we bound the probability of landing in S, which we do by a union bound over the elements in S (for which, there are at most 2^m):

$$\Pr_{x \leftarrow \{0,1\}^n}[f(x) \in S] = \Pr[\exists y \in S \text{ with } f(U_n) = y] < \frac{2^{\frac{3}{4}n-m}}{2^n} \cdot 2^m = 2^{-n/4} = \text{neg}(n) \ . \qquad \square$$

Hiding. Having separated the properties of a universal one-way hash family into having large preimages and having target collision resistance, we now show that the large

[13] As we will see later in Section 3.6, it is possible for an *instance-dependent* family of functions to have large preimages and have statistical target collision resistance, albeit each property holds on different set of instances.

preimages property of $\mathcal{F}$ translates to the hiding property of the commitment scheme $(\mathbb{S}, \mathbb{R}) = \mathsf{HR\text{-}Transform}((\mathsf{S}, \mathsf{R}), \mathcal{F})$ obtained from the Haitner & Reingold transformation.

LEMMA 3.5.36

If the family of functions $\mathcal{F}$ has the large preimages property, and the 2-phase commitment scheme (S, R) is statistically hiding, then scheme $(\mathbb{S}, \mathbb{R}) = \mathsf{HR\text{-}Transform}((\mathsf{S}, \mathsf{R}), \mathcal{F})$ is statistically hiding.

Proof. What we need to show is that for any adversarial receiver R^*, the views of R^* in $(\mathbb{S}(0), R^*)$ and $(\mathbb{S}(1), R^*)$ are statistically indistinguishable. (In this proof, we drop the security parametrization of 1^n because it is clear from context.) We can, without loss of generality, only consider deterministic R^* because we can fix the adversary's coin tosses to maximize its distinguishing advantage. In the rest of this proof, we use *indistinguishability* and *hiding* to mean those of the statistical variant.

Let P denote the value of *phase* sent by R^*, and we break our hiding analysis to cases when $P = 1$ and $P = 2$. To formalize this case analysis, we say that random variables X and Y are ***indistinguishable on event*** E if for all D, $|\Pr[D(X) = 1 \wedge E] - \Pr[D(X) = 0 \wedge E]|$ is negligible (in the security parameter n). What we will show is that the random variables $\mathrm{view}_{R^*}(\mathbb{S}(0), R^*)$ and $\mathrm{view}_{R^*}(\mathbb{S}(1), R^*)$ are indistinguishable on both events $P = 1$ and $P = 2$, thus allowing us to conclude that the scheme is hiding.

First, we analyze the case when $P = 2$. Let random variable Σ and F denote $\mathbb{S}$'s choice of σ and the value of f sent by R^*, respectively. Observe that P is a *deterministic* function of the random variables $V_1 = \mathrm{view}_{R^*}(\mathsf{S}^1_c(\Sigma), R^*)$ and $Y = F(\Sigma)$. In turn, V_1 and Y are deterministic functions of the first-phase transcript $\mathrm{T} = \mathrm{transcript}(\mathsf{S}^1(\Sigma), R^*)$, which includes both the commit and reveal stages. This is because we can compute the view of the receiver from the first-phase transcript, and the first-phase transcript also contains the value of σ, from which we can compute $y = f(\sigma)$. For bit $b \in \{0, 1\}$, let random variable $V_2(b) = \mathrm{view}_{R^*}(\mathsf{S}^2_c(b), R^*)(\mathrm{T})$, recalling that $\mathrm{T} = \mathrm{transcript}(\mathsf{S}^1(\Sigma), R^*)$. Because (S, R) is hiding, its 2-phase commitments is hiding even given the first-phase transcript: this means that $(V_2(0), \mathrm{T})$ is indistinguishable from $(V_2(1), \mathrm{T})$. Since P is a deterministic function of T, random variables $(V_2(0), \mathrm{T})$ and $(V_2(1), \mathrm{T})$ are indistinguishable on event $P = 2$. Since $\mathrm{view}_{R^*}(\mathbb{S}(b), R^*)|_{P=2}$ is a deterministic function of $(V_2(b), \mathrm{T})|_{P=2}$, for $b \in \{0, 1\}$, we have that $\mathrm{view}_{R^*}(\mathbb{S}(0), R^*)$ and $\mathrm{view}_{R^*}(\mathbb{S}(1), R^*)$ are indistinguishable on event $P = 2$.

Next, we analyze the case when $P = 1$. The hiding property of the first phase gives us

$$(V_1, \Sigma) \approx_s (V_1, U_n) \ ,$$

where U_n represent a uniform random variable over $\{0, 1\}^n$, and is independent from V_1 and Σ. Recall that random variable F denotes the function f sent by R^*. Since F is a deterministic function of V_1, we get

$$(V_1, F, F(\Sigma), \Sigma) \approx_s (V_1, F, F(U_n), U_n) \ .$$

Now, let random variable H represent the hash function h selected by $\mathbb{S}$ when $phase = 1$. Note that H is independent of V_1, F, Σ, and U_n, so

$$(V_1, F, Y, H, H(\Sigma)) \approx_s (V_1, F, F(U_n), H, H(U_n)) \ , \tag{3.9}$$

recalling that $Y = F(\Sigma)$.

What we need to establish is that $H(U_n)$ is close to uniform so that we have hiding. The next claim does this for us.

CLAIM 3.5.37

Suppose family of functions $\mathcal{F} = \bigcup_n \mathcal{F}_n$ has the large preimages property. Let random variable H denote a random hash function from a family of pairwise-independent hash functions with domain $\{0,1\}^n$ and range $\{0,1\}$, random variable U_n denote a uniform string in $\{0,1\}^n$, random variable U_1' denote a uniform string in $\{0,1\}$, and that H, U_n, and U_1' are all independent. For every $f \in \mathcal{F}_n$, $(f(U_n), H, H(U_n))$ is indistinguishable from $(f(U_n), H, U_1')$.

Proof of Claim. The large preimages property of $\mathcal{F}$ guarantees that with probability $1 - \text{neg}(n)$ over $y \leftarrow f(U_n)$, the min-entropy $\mathrm{H}_\infty(U_n|_{f(U_n)=y}) \geq \omega(\log n)$. For y satisfying this condition, we apply the Leftover Hash Lemma 3.3.1 to get that $(y, H, H(U_n|_{f(U_n)=y}))$ is indistinguishable from $(y, H, H(U_n|_{f(U_n)=y}))$. □

Because H and U_n are independent from the rest of the random variables (and are independent from each other), Claim 3.5.37 states that

$$(V_1, F, F(U_n), H, H(U_n)) \approx_s (V_1, F, F(U_n), H, U_1') \ , \tag{3.10}$$

where U_1' is an independent random variable representing a uniform random variable over $\{0,1\}$. Combining (3.9) and (3.10), we get

$$(V_1, F, Y, H, H(\Sigma)) \approx_s (V_1, F, F(U_n), H, U_1') \ ,$$

which leads to:

$$\begin{aligned}(V_1, F, Y, H, 0 \oplus H(\Sigma)) &\approx_s (V_1, F, F(U_n), H, 0 \oplus U_1') \\ &\equiv \ (V_1, F, F(U_n), H, 1 \oplus U_1') \\ &\approx_s (V_1, F, Y, H, 1 \oplus H(\Sigma)) \ .\end{aligned}$$

Since P is a deterministic function of V_1 and Y, random variables $(V_1, F, Y, H, 0 \oplus H(\Sigma))$ and $(V_1, F, Y, H, 1 \oplus H(\Sigma))$ are indistinguishable on event $P = 1$. Since $\mathrm{view}_{R^*}(\mathbb{S}(b), R^*)|_{P=1}$ is a deterministic function of $(V_1, F, Y, H, b \oplus H(\Sigma))|_{P=1}$, for $b \in \{0, 1\}$, we have that $\mathrm{view}_{R^*}(\mathbb{S}(0), R^*)$ and $\mathrm{view}_{R^*}(\mathbb{S}(1), R^*)$ are indistinguishable on event $P = 1$. □

Binding. We show that the target collision resistance property of $\mathcal{F}$ translates to the binding property of the commitment scheme $(\mathbb{S}, \mathbb{R}) = \mathsf{HR\text{-}Transform}((\mathsf{S}, \mathsf{R}), \mathcal{F})$ obtained from the Haitner & Reingold transformation. Because we will only be able to show that $(\mathbb{S}, \mathbb{R})$ is binding with probability close to 1/2, we first define what it means to for a scheme to be binding with probability δ, for some $\delta \in [0, 1]$.

DEFINITION 3.5.38

Commitment scheme (S, R) is ***statistically [resp., computationally]*** $\delta(n)$***-binding*** if for all [resp., nonuniform PPT] S^* and every large enough values of n, sender S^* succeeds in the following game with probability at most $\delta(n)$:

> On security parameter 1^n, S^* interacts with R in the commit stage obtaining commitment c. Then S^* outputs pairs $(0, d_0)$ and $(1, d_1)$, and *succeeds* if in the reveal stage, $R(0, d_0, c) = R(1, d_1, c) = \texttt{accept}$.

The standard notion of binding as given in Definition 2.4.4 corresponds to being $(1 - \mathrm{neg}(n))$-binding in the above definition.

LEMMA 3.5.39

If the family of functions $\mathcal{F}$ is statistically [resp., computationally] universal one-way, and the 2-phase commitment scheme (S, R) is statistically [resp., computationally] 1-out-of-2 binding, then scheme $(\mathbb{S}, \mathbb{R}) = \mathsf{HR\text{-}Transform}((\mathsf{S}, \mathsf{R}), \mathcal{F})$ is statistically [resp., computationally] $(1/2 - \mathrm{neg}(n))$-binding.

Proof. For this proof, we take probabilities over the entire interaction between S^* and $\mathbb{R}$ in both the commit and reveal stages, unless stated otherwise. Since S^* is nonuniform, we can assume without loss of generality that reveal stage is *noninteractive*, and the message sent by S^* in the reveal stage is a *deterministic* function of its view in the commit phase. So naturally, we say that S^* ***breaks*** commitment $\Upsilon = \mathrm{output}(S^*, \mathbb{R})$ if it is able to produce decommitments to two different messages for commitment Υ in the reveal phase. (In this proof, we drop the security parametrization of 1^n because it is clear from context.)

We need to upper bound the probability that S^* breaks the commitment Υ. To help us do so, we establish several useful random variables. Let random variable $C = \mathrm{output}(S^*, \mathsf{R}_c^1)$ denote the first-phase commitment of S^* interacting with R_c^1. For each first-phase commitment $c \in C$, we will need to define a value of σ^* that will be interpreted as the commitment of S^* in the first phase such that only if S^* reveals to σ^*, will S^* be able to cheat in the second phase. To do so, define the following measure for each first-phase commitment $c \in C$:

$$p_\sigma[c] = \Pr\left[\begin{array}{c} S^* \text{ produces an } \textit{accepting} \text{ full transcript } \lambda = (\tau, \kappa) \\ \text{such that } \tau \notin \mathcal{B} \text{ and } \tau \text{ contains } c \text{ and } \sigma \end{array}\right], \tag{3.11}$$

where we say full transcript λ is ***accepting*** if both R_r^1 and R_r^2 accept in λ. With this measure, we define $\sigma^*[c] = \operatorname{argmax}_\sigma p_\sigma[c]$, breaking ties arbitrarily (say, by choosing the lexicographic smallest σ). We define the random variables P, F, Σ^*, Σ_1, and Σ_2 as follows:

- P represents the value of *phase*;
- F represents the function f from the family $\mathcal{F}$;
- $\Sigma^* = \sigma^*[C]$. Note that Σ^* is a *deterministic* function of C;
- Σ_1 represents the value of $\sigma \neq \Sigma^*$ given by S^* when $P = 1$ (if no such value exists, then the value of Σ_1 defaults to Σ^*);
- Σ_2 represents the value of σ revealed when $P = 2$.

We use the notation $\Sigma_2 \notin \mathcal{B}$ to denote the event of $P = 2$ and S^* completing an *accepting* full transcript $\lambda = (\tau, \kappa)$ such that $\tau \notin \mathcal{B}$ and τ contains c and Σ_2. In addition, we use the notation $\Sigma_2 \in \mathcal{B}$ to denote the *complementary event* of $\Sigma_2 \notin \mathcal{B}$ not happening.

Having established the appropriate random variables, we turn our attention back to bounding the probability that S^* breaks the commitments as follows:

$$\begin{aligned}\Pr[S^* \text{ breaks } \Upsilon] &\le \Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) = Y] + \Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) \neq Y] \\ &\le \begin{array}{l} \quad\Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) = Y \wedge P = 1] + \Pr[F(\Sigma^*) = Y \wedge P = 2] \\ +\Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) \neq Y \wedge P = 2] + \Pr[F(\Sigma^*) \neq Y \wedge P = 1] \end{array} \\ &= \begin{array}{l} \quad\Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) = Y \wedge P = 1] \\ +\Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) \neq Y \wedge P = 2] + 1/2 \end{array}\end{aligned} \tag{3.12}$$

with the last equality following from the fact that P is independent of Y, F, and Σ^*. Therefore, all we need to do is to bound both $\Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) = Y \wedge P = 1]$ and $\Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) \neq Y \wedge P = 2]$ by negligible functions, and we are done.

To bound $\Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) = Y \wedge P = 1]$, we will use the property of $\mathcal{F}$ having target collision resistance. For that, let ε_{uow} be the negligible function for $\mathcal{F}$ given by Definition 3.5.34. Since F is chosen independent of Σ^*, we have

$$\Pr[F(\Sigma_1) = F(\Sigma^*) \wedge \Sigma_1 \neq \Sigma^*] \le \varepsilon_{\text{uow}}. \tag{3.13}$$

For S^* to break commitment Υ when $F(\Sigma^*) = Y$ and $P = 1$, S^* needs to produce a

$\sigma \neq \Sigma^*$, and hence it must be the case that $\Sigma_1 \neq \Sigma^*$. With this in mind, we bound:

$$\begin{aligned}
&\Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) = Y \wedge P = 1] \\
&\quad \leq \Pr[F(\Sigma_1) = Y \wedge \Sigma_1 \neq \Sigma^* \wedge F(\Sigma^*) = Y] \\
&\quad = \Pr[F(\Sigma_1) = F(\Sigma^*) \wedge \Sigma_1 \neq \Sigma^*] \\
&\quad \leq \varepsilon_{\text{uow}} && \text{(from 3.13).}
\end{aligned}$$

To bound $\Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) \neq Y \wedge P = 2]$, we will exploit the 1-out-of-2 binding property of (S, R). The way we do so is stated in the next claim.

CLAIM 3.5.40

The following holds:

$$\Pr[\Sigma_2 \neq \Sigma^* \wedge \Sigma_2 \notin \mathcal{B}] = O((\varepsilon_{\text{bind}})^{1/3}),$$

where $\varepsilon_{\text{bind}}$ is the negligible function for 1-out-of-2-binding scheme (S, R) given by Definition 3.4.4.

Proof of Claim. Let $\varepsilon = \Pr[\Sigma_2 \neq \Sigma^* \wedge \Sigma_2 \notin \mathcal{B}]$. Taking probability over $C = \text{output}(S^*, \mathsf{R}_c^1)$, we get

$$\Pr_{c \leftarrow C}\big[\Pr[\Sigma_2 \neq \sigma^*[c] \wedge \Sigma_2 \notin \mathcal{B} \mid C = c] \geq \varepsilon/2\big] \geq \varepsilon/2.$$

Call c ***good*** if $\Pr[\Sigma_2 \neq \sigma^*[c] \wedge \Sigma_2 \notin \mathcal{B} \mid C = c] \geq \varepsilon/2$. This means that the probability of choosing a good $c \leftarrow C$ is at least $\varepsilon/2$.

Now, for a good c, we claim that with nonnegligible probability, we can find two accepting full transcripts $\lambda = (\tau, \kappa)$ and $\widetilde{\lambda} = (\widetilde{\tau}, \widetilde{\kappa})$, such that both $\tau, \widetilde{\tau} \notin \mathcal{B}$ and τ and $\widetilde{\tau}$ contains σ and $\widetilde{\sigma} \neq \sigma$, respectively. Call this event ***break first phase binding***.

From (3.11), we have

$$p_\sigma[c] = \Pr\left[\begin{array}{c} S^* \text{ produces an } \textit{accepting} \text{ full transcript } \lambda = (\tau, \kappa) \\ \text{such that } \tau \notin \mathcal{B} \text{ and } \tau \text{ contains } c \text{ and } \sigma \end{array}\right],$$

so for a good c, it must be the case that

$$\sum_\sigma p_\sigma[c] = p_{\sigma^*}[c] + \Pr[\Sigma_2 \neq \sigma^*[c] \wedge \Sigma_2 \notin \mathcal{B} \mid C = c] > \varepsilon/2.$$

After obtaining first-phase commitment c, we run two independent executions of S^* that continues from c. Because σ^* is maximal in the sense that $p_{\sigma^*}[c] \geq p_\sigma[c]$ for all other σ's, the probability that we land in the event *break first phase* *binding* conditioned that c is good is at least $\Omega(\varepsilon^2)$. Consequently, we break the first-phase binding property of (S,R) with probability:

$$\Pr[c \text{ is good}] \cdot \Pr[\textit{break first phase binding} \mid c \text{ is good} \geq (\varepsilon/2) \cdot \Omega(\varepsilon^2) = \Omega(\varepsilon^3) \ ,$$

and hence, this forces $\varepsilon = O((\varepsilon_{\text{bind}})^{1/3})$. □

Having established the above claim, we bound:

$$\begin{array}{rl}
\multicolumn{2}{l}{\Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) \neq Y \wedge P = 2]} \\
= & \Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) \neq Y \wedge F(\Sigma_2) = Y] \\
= & \Pr[S^* \text{ breaks } \Upsilon \wedge \Sigma_2 \neq \Sigma^*] \\
= & \begin{array}{l} \ \ \Pr[S^* \text{ breaks } \Upsilon \wedge \Sigma_2 \neq \Sigma^* \wedge \Sigma_2 \in \mathcal{B}] \\ + \Pr[S^* \text{ breaks } \Upsilon \wedge \Sigma_2 \neq \Sigma^* \wedge \Sigma_2 \notin \mathcal{B}] \end{array} \\
\leq & \begin{array}{l} \ \ \Pr[S^* \text{ breaks second phase of } \mathsf{R}^2 \ \wedge \Sigma_2 \in \mathcal{B}] \\ + \Pr[\Sigma_2 \neq \Sigma^* \wedge \Sigma_2 \notin \mathcal{B}] \end{array} \\
\leq & \Pr[S^* \text{ breaks second phase of } \mathsf{R}^2 \ \wedge \Sigma_2 \in \mathcal{B}] + O((\varepsilon_{\text{bind}})^{1/3}) \quad \text{(by Claim 3.5.40)} \\
\leq & \varepsilon_{\text{bind}} + O((\varepsilon_{\text{bind}})^{1/3}) \\
= & O((\varepsilon_{\text{bind}})^{1/3}) \ .
\end{array}$$

Finally, we continue from where we left at (3.12) to bound the success probability of S^* breaking the binding property of the commitment.

$$\begin{array}{rll}
\Pr[S^* \text{ breaks } \Upsilon] = & \begin{array}{l} \ \ \Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) = Y \wedge P = 1] \\ + \Pr[S^* \text{ breaks } \Upsilon \wedge F(\Sigma^*) \neq Y \wedge P = 2] + 1/2 \end{array} & \text{(from 3.12)} \\
\leq & \varepsilon_{\text{uow}} + O((\varepsilon_{\text{bind}})^{1/3}) + 1/2 & \\
= & 1/2 + \varepsilon' \ , &
\end{array}$$

for a negligible function $\varepsilon' = \varepsilon_{\text{uow}} + O((\varepsilon_{\text{bind}})^{1/3})$. □

Boosting the binding. The commitment scheme $(\mathbb{S},\mathbb{R})$ from Lemma 3.5.39 is only $(1/2-\text{neg}(n))$-binding. Nonetheless, we can boost its binding probability to $1-\text{neg}(n)$ as follows: to commit to bit b, run n independent executions of $(\mathbb{S}(b),\mathbb{R})$ in parallel. Intuitively, this new scheme is $(1-\text{neg}(n))$-binding, because to cheat, one will need to cheat in all n independent executions, and the probability of succeeding in that is bounded by $(1/2{+}\text{neg}(n))^n = \text{neg}(n)$.

CLAIM 3.5.41

(Folklore, cf., [HR2, Prop. 2.9].) There exists an efficient procedure that converts a statistically [resp., computationally] $(1/2 - \text{neg}(n))$-binding commitment scheme $(\mathbb{S},\mathbb{R})$ into commitment scheme (S, R) that is statistically [resp., computationally] binding. Furthermore, if $(\mathbb{S}, \mathbb{R})$ is statistically hiding, so is (S, R).

Putting it all together

Having established the appropriate claims and lemmas, we now state what is achievable from the Haitner & Reingold transformation [HR2].

PROPOSITION 3.5.42

There exist an efficient procedure, call it HR-FullTransform, that takes as inputs a 2-phase commitment scheme (S, R) and a family of functions $\mathcal{F}$, and outputs a commitment scheme $(S, R) =$ HR-FullTransform$((\mathsf{S}, \mathsf{R}), \mathcal{F})$ satisfying the following properties:

- If (S, R) is statistically hiding and $\mathcal{F}$ has the large preimages property, then (S, R) is statistically hiding.
- If (S, R) is statistically [resp., computationally] 1-out-of-2 binding and $\mathcal{F}$ has statistical [resp., computational] target collision resistance, then (S, R) is statistically [resp., computationally] binding (in the standard sense of binding).
- If (S, R) is public coin, then (S, R) is also public coin.

Proof. We describe the HR-FullTransform algorithm, recapping what we have done thus far, as follows.

1. On input 2-phase commitment scheme (S, R) and family of functions $\mathcal{F}$, convert (S, R) to 2-phase commitment scheme $(\mathsf{S}', \mathsf{R}')$ that has appropriate message lengths $(k_1, k_2) = (n, 1)$. This step follows from Claim 3.5.27.
2. Next, apply Algorithm 3.5.28 on $(\mathsf{S}', \mathsf{R}')$ and $\mathcal{F}$ to obtain a (standard) commitment scheme $(\mathbb{S}, \mathbb{R})$. Lemmas 3.5.36 and 3.5.39 state that for the right properties of both $(\mathsf{S}', \mathsf{R}')$ and $\mathcal{F}$ (see the first two items in Proposition 3.5.42 above), $(\mathbb{S}, \mathbb{R})$ is hiding and $(1/2 - \text{neg}(n))$-binding.
3. Finally, using Claim 3.5.41, boost the binding of $(\mathbb{S}, \mathbb{R})$ to obtain a scheme (S, R) that is $(1 - \text{neg}(n))$-binding while not affecting the hiding property. Output (S, R) as our desired scheme.

As for the preservation of the public coin property, observe that the messages sent by $\mathbb{R}$ that are specific to the Haitner & Reingold transformation are choosing $f \leftarrow \mathcal{F}$ and selecting *phase* $\leftarrow \{0, 1\}$, both of which are public coin operations. □

With Proposition 3.5.42 in hand, let us see how we can construct statistically hiding and computationally binding commitments from any one way function; this is captured by Theorem 3.0.4, restated as follows.

RESTATEMENT OF THEOREM 3.0.4

(First appeared in [HR2, Thm. 1.1].) If one-way functions exist, then there exist commitment schemes that are statistically hiding and computationally binding. Moreover, the commitment schemes obtained are public coin.

Proof of Theorem 3.0.4. We start off by constructing a collection of 2-phase commitment schemes from any one-way function; this is given by Theorem 3.5.1 which states that if one-way functions exist, then on security parameter 1^n, we can construct in time polynomial in n a collection of public-coin 2-phase commitment schemes $\mathcal{COM} = \{\mathsf{Com}_1, \cdots, \mathsf{Com}_m\}$, where $m = \mathrm{poly}(n)$, such that:

- there exists an index $i \in \{1, 2, \ldots, m\}$ such that scheme Com_i is statistically hiding, and
- for every index $i \in \{1, 2, \ldots, m\}$, scheme Com_i is computationally 1-out-of-2 binding.

Now we apply Proposition 3.5.42 to each 2-phase commitment Com_i in the collection with a universal one-way hash function family $\mathcal{F}$, which can be constructed from any one-way function [Rom] (see also [KK]). Let the resulting (standard) commitment schemes be $\mathsf{Com}'_i = \mathsf{HR\text{-}FullTransform}(\mathsf{Com}_i, \mathcal{F})$. By Proposition 3.5.42 and Lemma 3.5.35, we know that:

- Com'_i is statistically hiding if Com_i is statistically hiding,
- Com'_i is computationally binding if Com_i is computationally 1-out-of-2 binding, and
- Com'_i is public coin if Com_i is public coin.

This means that we now have a collection of public-coin (standard) commitment schemes $\mathcal{COM}' = \left\{\mathsf{Com}'_1, \cdots, \mathsf{Com}'_m\right\}$, where $m = \mathrm{poly}(n)$, such that:

- there exists an index $i \in \{1, 2, \ldots, m\}$ such that scheme Com'_i is statistically hiding, and
- for every index $i \in \{1, 2, \ldots, m\}$, scheme Com'_i is computationally binding (in the standard sense of binding).

We are almost done, except that we are still left with a collection of commitments instead of a *single* commitment scheme. To obtain that, we combine these schemes using a technique of secret sharing the committed bit. That is, to commit to bit b, we first secret share b into m shares $b_1, \ldots, b_m$ such that each share b_i is uniform in $\{0, 1\}$, and $b = b_1 \oplus b_2 \oplus \cdots \oplus b_m$. We then use Com'_i to commit to each bit b_i, with the m independent executions of $\mathsf{Com}'_1, \ldots, \mathsf{Com}'_m$ done in parallel (to save on round complexity). Intuitively, this would be hiding since at least one scheme Com'_i will hide the value of b_i, and knowing even $m-1$ shares will not reveal the value of b. The binding property follows from the fact that all schemes Com'_i are binding. And this new scheme will be public coin if the all Com'_i are. Our above observation is captured by the following claim.

CLAIM 3.5.43

(Folklore, cf., [HR2, Prop. 2.8].) There is an efficient procedure that converts a polynomial collection of commitment schemes, at least one of which is statistically hiding and all are computationally binding, into a *single* commitment scheme that is statistically hiding and computationally binding. In addition, if we start off with public-coin schemes, we also end up with a public-coin scheme.

Our proof is now complete since we now have a *single* commitment scheme that is statistically hiding and computationally binding, and the only complexity assumption made is the existence of one-way functions. □

We end this section with the following observation: the statistical hiding property of the commitment scheme constructed in the above proof of Theorem 3.0.4 does not depend on the one-way security of the function that the scheme is based on. Thus, we can construct an instance-dependent commitment scheme from any instance-dependent function such that the scheme is always statistically hiding, but is guaranteed to be computationally binding only on the instances where the function is hard to invert. This is stated in the following proposition, which can be viewed as an instance-dependent formulation of Theorem 3.0.4.

PROPOSITION 3.5.44

For every set $K \subseteq \{0,1\}^*$, if there is an instance-dependent one-way function on K, then problem $(\overline{K}, K)$ has an instance-dependent commitment that is statistically hiding on the YES instances (namely, instances in $\overline{K}$), and computationally binding on the NO instances (namely, instances in K). Moreover, the instance-dependent commitment scheme obtained is public coin.

3.6 Instance-Dependent Variant

Having constructed statistically-hiding and computationally-binding commitments from any one-way function, our theme in this section is to eliminate unproven assumptions—like the existence of one-way functions—and construct *instance-dependent* commitments that are based on special properties of certain class of problems. Recall that instance-dependent commitments are commitment schemes where both sender and receiver strategies can depend on the specific instance x of a problem Π. (Refer back to Section 2.4.4 if needed.)

The results presented in this section will show that every problem $\Pi = (\Pi_Y, \Pi_N) \in$ SZKP having statistical zero-knowledge proofs yield instance-dependent commitments for Π that are *statistically hiding* on the YES instances (meaning, instances in Π_Y), and *statistically binding* on the NO instance (meaning, instances in Π_Y). Note that this statement is unconditional—in that it does not rely on any unproven assumptions—and the instance-dependent commitments constructed have statistical security for both hiding and binding, a feature that is impossible for standard commitments.

Prior works have constructed instance-dependent commitments for certain specific problems in SZKP or a limited subclass of SZKP, such as Graph Isomorphism [BMO, IOS], Quadratic Residuosity [IOS], approximate versions of *lattice* problems [MV], and *random self-reducible* problems [TW, DDPY1, KMS]. But it was not until the seminal work of Vadhan did we know how to construct instance-dependent commitments for all problems in SZKP, even allowing for certain relaxations in the notion of commitments. Vadhan [Vad3] showed that every problem $\Pi \in$ SZKP has instance-dependent commitments, albeit with an inefficient sender algorithm. These instance-dependent commitments suffice for some applications like converting honest-verifier zero knowledge proofs into zero knowledge proofs secure against any adversarial verifier, without relying on unproven assumptions, but the downside is that we are left with a protocol having an inefficient prover strategy.

To obtain efficient-prover zero-knowledge proofs for NP, we need commitment schemes whose both sender and receiver strategies are efficient. Nguyen and Vadhan [NV] overcame this problem by constructing instance-dependent commitments for SZKP whose sender and receiver algorithms are efficient, but they paid a price in that their commitments are only 1-out-of-2 binding. It turns out that 1-out-of-2-binding commitments suffice for obtaining efficient-prover zero-knowledge proofs for all languages in SZKP$\cap$NP, but applications of 1-out-of-2 binding are still limited to settings where only the prover does the commitments, not to mention the complications that arise when dealing with 1-out-of-2-binding commitments. (Look back at Section 3.5 if you're not convinced of the intricacies of 1-out-of-2-binding commitments!)

Our goal in this section is to extend the result of Nguyen and Vadhan to prove that every problem in SZKP has instance-dependent commitments with the standard binding property, and with efficient sender and receiver strategies. Because these are standard notions for commitments (with the exception of being instance dependent), they have a much wider applicability—for example, we can now let the verifier do commitments too—and they are used throughout to establish the results in Chapter 4. Two nice features of our commitments are that they are very round efficient and are public coin, in addition to being constructed without assuming any unproven assumptions.

3.6.1 Instance-dependent commitments for statistical zero-knowledge proofs

An initial attempt

Consider the SZKP-complete problem Statistical Difference [SV], defined as SD = $(\mathrm{SD_Y}, \mathrm{SD_N})$ with:

$$\mathrm{SD_Y} = \{(X, Y) : \Delta(X, Y) \leq 1/3\}\,;$$
$$\mathrm{SD_Y} = \{(X, Y) : \Delta(X, Y) \geq 2/3\}\,,$$

where X and Y are represented by *circuits encoding these random variables*, and recall that $\Delta(X, Y)$ is the *statistical difference* between X and Y (see Section 2.2.1). Here, and throughout this section, a ***circuit encoding a random variable*** X is a Boolean circuit whose output on a uniformly random input string is identically distributed to X. Without loss of generality, we can assume that the circuit encoding a random variable X have size at most n^2, where n is the number of inputs to that circuit. (This can be done by padding dummy input variables to the circuit.)

The first approach one might think of in constructing instance-dependent commitments for all of SZKP is to start with the Statistical Difference problem. This is because there is a natural and simple instance-dependent scheme for Statistical Difference as follows.

> To commit to bit $b = 0$, sender S sends a random sample from X to R, and to commit to bit $b = 1$, sender S send a random sample from Y to R. To decommit, sender S just reveals its random coins used in sampling from either X or Y.

This scheme is somewhat hiding considering that the statistical distance between X and Y is less than 1/3 for the YES instances, and also seems to be somewhat binding considering the statistical distance between X and Y is greater than 2/3 for the NO instances. It turns out that we can boost the bounds to be $\Delta(X, Y) \leq \mathrm{neg}(n)$ for YES instances, and $\Delta(X, Y) \geq 1 - \mathrm{neg}(n)$ for NO instances; this seems to further suggest that this technique might work. On further inspection, however, a fundamental problem of binding arises with our simple commitment scheme when the NO instances have $\Delta(X, Y) < 1$. This is because a cheating sender S^* could possibly find a string $z \in \mathrm{Supp}(X) \cap \mathrm{Supp}(Y)$, and always send z as its commitments. If S^* knows how to find inverses of z under both X and Y, then it can decommit to both 0 and 1.

Nonetheless, as shown by Micciancio and Vadhan [MV], this simple commitment scheme is an instance-dependent commitment scheme for a restricted version of Statistical Difference, where the NO instances are such that X and Y have disjoint supports (i.e., $\Delta(X, Y) = 1$). We do still do not know if this restricted version of Statistical Difference is SZKP-complete, so we cannot depend on its instance-dependent commitment scheme to get instance-dependent commitments for all of SZKP. Vadhan [Vad3], however, managed to work a way around this problem of binding by designing an alternative scheme that works for (the non-restricted version of) Statistical Difference, albeit with an inefficient sender.

The Nguyen & Vadhan approach

To obtain efficient senders, Nguyen and Vadhan [NV] started off from a different SZKP-complete problem, namely the Entropy Difference [GV] problem $\mathrm{ED} = (\mathrm{ED_Y}, \mathrm{ED_N})$, defined as:

$$\begin{aligned}\mathrm{ED_Y} &= \{(X,Y) : \mathrm{H}(X) \geq \mathrm{H}(Y) + 1\};\\ \mathrm{ED_N} &= \{(X,Y) : \mathrm{H}(X) \leq \mathrm{H}(Y) - 1\},\end{aligned}$$

where X and Y are represented by circuits encoding these random variables, and $\mathrm{H}(\cdot)$ denotes the *entropy* measure (see Section 2.2.1). Their construction of instance-dependent schemes for ED is not a commitment scheme in the standard sense, but are commitments with the weaker *1-out-of-2 binding* property.[14] These commitments, even though with a weaker binding property, suffice for getting efficient-prover statistical zero-knowledge proofs for all of $\mathrm{SZKP} \cap \mathrm{NP}$ [NV].

Our construction of instance-dependent commitments for all of SZKP will follow closely the Nguyen & Vadhan approach, except at the part where they get stuck with 1-out-of-2-binding commitments, we convert them into commitments with the standard binding property using techniques from Section 3.5.5. Specifically, we use an instance-dependent variant of the Haitner & Reingold transformation to convert 1-out-of-2-binding commitments into commitments with the standard binding property.

Let us trace back the Nguyen & Vadhan approach. First, they did not construct their schemes directly from ED, but instead first established a Cook reduction from ED to a *restricted version* of the Entropy Approximation [GSV2] problem, denoted as EA' = $(\mathrm{EA'_Y}, \mathrm{EA'_N})$, and defined below:

$$\begin{aligned}\mathrm{EA'_Y} &= \{(X,t) : \mathrm{H}(X) \geq t + 1\};\\ \mathrm{EA'_N} &= \{(X,t) : t - 1/n^{14} \leq \mathrm{H}(X) \leq t\},\end{aligned}$$

where the circuit encoding the random variable X has input length n. The problem EA' is considered a restricted version of Entropy Approximation because (unrestricted version of) the Entropy Approximation problem $\mathrm{EA} = (\mathrm{EA_Y}, \mathrm{EA_N})$ does not lower-bound the

[14] 2-phase commitments scheme (Definition 3.4.1), and its corresponding hiding and 1-out-of-2 binding properties (Definitions 3.4.3 and 3.4.4, respectively), can be extended to instance-dependent analogues in a similar fashion as done for standard commitments in Section 2.4.4.

entropy in the case of the NO instances. EA is defined as follows:

$$\begin{aligned}\mathrm{EA}_{\mathrm{Y}} &= \{(X,t) : \mathrm{H}(X) \geq t+1\};\\ \mathrm{EA}_{\mathrm{N}} &= \{(X,t) : \mathrm{H}(X) \leq t\}.\end{aligned}$$

The Cook reduction from ED to EA' is established by the following proposition.

PROPOSITION 3.6.1

(Cook Reduction from ED to EA'; from [NV, Lem. 4.9], which builds on [GSV2].) Let (X, Y) be an instance of the Entropy Difference problem $\mathrm{ED} = (\mathrm{ED}_{\mathrm{Y}}, \mathrm{ED}_{\mathrm{N}})$, where the circuits encoding the random variables X and Y both have input length n. The Cook reduction from ED to EA' is as follows:

$$(X,Y) \in \mathrm{ED}_{\mathrm{Y}} \Rightarrow \bigvee_{i=0}^{n\cdot k} \left((Y, i/k) \in \overline{\mathrm{EA'}}_{\mathrm{Y}} \wedge \bigwedge_{j=0}^{i} (X, j/k) \in \mathrm{EA'}_{\mathrm{Y}} \right) \quad ;$$

$$(X,Y) \in \mathrm{ED}_{\mathrm{N}} \Rightarrow \bigwedge_{i=0}^{n\cdot k} \left((Y, i/k) \in \overline{\mathrm{EA'}}_{\mathrm{N}} \vee \bigvee_{j=0}^{i} (X, j/k) \in \mathrm{EA'}_{\mathrm{N}} \right) \quad ,$$

where $k = n^{14}$.

With this proposition, Nguyen and Vadhan noted that it suffices to construct instance-dependent commitments for both EA' and its complement $\overline{\mathrm{EA'}}$ in order to obtain instance-dependent commitments for ED (and hence, all of SZKP).

LEMMA 3.6.2

If both the special case of the Entropy Approximation problem EA' and its complement $\overline{\mathrm{EA'}}$ have instance-dependent commitments, namely:

- there exists instance-dependent commitments that are statistically hiding on instances in $\mathrm{EA'}_{\mathrm{Y}}$ and statistically binding on instances in $\mathrm{EA'}_{\mathrm{N}}$, and
- there exists instance-dependent commitments that are statistically hiding on instances in $\mathrm{EA'}_{\mathrm{N}}$ and statistically binding on instances in $\mathrm{EA'}_{\mathrm{Y}}$.

Then the Entropy Difference problem ED (and hence, every problem in SZKP) has an instance-dependent commitment scheme that is *statistically hiding* on the YES instances and *statistically binding* on the NO instances.

Proof. Let Com_x and Com'_x be the instance-dependent commitments for EA' and $\overline{\mathrm{EA'}}$, respectively. The instance-dependent commitment scheme for ED is as follows.

> On instance (X, Y) and an input bit b, first secret share b into $n \cdot k = n^{15}$ secrets shares $b_1, \ldots, b_{nk}$ with $b = b_1 \oplus \cdots \oplus b_{nk}$. Then for each share b_i, commit to b_i a total of $(i+2)$ times as follows: run $\mathsf{Com}'_{(Y,i/k)}(b_i)$, and for each $j = 0, 1, \ldots, i$, run $\mathsf{Com}_{(X,j/k)}(b_i)$. The executions are done in *parallel* to save on round complexity.

The hiding and binding properties of the above scheme follows from Proposition 3.6.1. □

Indeed, Nguyen and Vadhan [NV] constructed instance-dependent schemes for both EA' and $\overline{\text{EA'}}$. Their scheme for EA' is a commitment with the standard binding property, but for $\overline{\text{EA'}}$, they only managed to only get 1-out-of-2-binding commitments. This deficiency is what that makes their overall scheme only 1-out-of-2 binding.

LEMMA 3.6.3

(From [NV, Thm. 4.4].) The special case of the entropy approximation problem EA' has an instance-dependent commitment that is statistically hiding on YES instances (namely, instances in EA'_Y) and statistically binding on NO instances (namely, instances in EA'_N). Moreover, their scheme is public coin and constant round.

LEMMA 3.6.4

(From [NV, Thm. 4.5].) The *complement* of the special case of the entropy approximation problem $\overline{\text{EA'}} = (\overline{\text{EA'}}_\text{Y}, \overline{\text{EA'}}_\text{N}) = (\text{EA'}_\text{N}, \text{EA'}_\text{Y})$ has an instance-dependent 2-phase commitment that is statistically hiding on the YES instances (namely, instances in EA'_N) and statistically 1-out-of-2 binding on NO instances (namely, instances in EA'_Y). Moreover, their scheme is public coin and constant round.

Instance-dependent commitments for $\overline{\text{EA'}}$

To obtain instance-dependent commitments (with the standard binding property) for $\overline{\text{EA'}}$, we use an instance-dependent variant of the Haitner & Reingold transformation [HR2] (from Section 3.5.5). The main difference here is that we are now dealing with instance-dependent cryptographic primitives, but nonetheless the techniques from Section 3.5.5 will still apply. With instance-dependent primitives, the security is measured in terms of the length of the instance. In the case of $\overline{\text{EA'}}$, the instance is (X, t), and the length of the instance is the size of the circuit encoding X, which in turn is bounded by n^2, where n is the input length to that circuit. Therefore, we can view the security parameter in terms of n.

With this in mind, we generalize the notion of large preimages property in Definition 3.5.33 and the target collision resistance property in Definition 3.5.34 to accommodate instance-dependent family of functions $\mathcal{F} = \bigcup_x \mathcal{F}_x = \{f \colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)|}\}$, where $n(\cdot)$ and $m(\cdot)$ are polynomials. We present our definition of an *instance-dependent universal one-way hash family*, which will be used in the Haitner & Reingold transformation in order to convert instance-dependent 1-out-of-2-binding commitments into corresponding commitments with the standard binding property.

DEFINITION 3.6.5

Problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$ has an ***instance-dependent universal one-way hash family*** if there exists a polynomial-time computable family $\mathcal{F} = \bigcup_x \mathcal{F}_x = \{f\colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}\}$, where $n(\cdot)$ and $m(\cdot)$ are polynomials, such that the following two conditions hold.

- The family $\mathcal{F}_{\mathrm{Y}} = \bigcup_{x\in\Pi_{\mathrm{Y}}} \mathcal{F}_x$ has the large preimages property in the sense of Definition 3.5.33. That is, there exists a function $\alpha(\cdot) = \omega(1)$ and a negligible function ε, such that the following holds for all $x \in \Pi_{\mathrm{Y}}$ and every function $f \in \mathcal{F}_x$:

$$\Pr_{y\leftarrow\{0,1\}^{n(|x|)}} \left[\left|f^{-1}(f(y))\right| \geq |x|^{\alpha(|x|)}\right] \geq 1 - \varepsilon(|x|) \ .$$

- The family $\mathcal{F}_{\mathrm{N}} = \bigcup_{x\in\Pi_{\mathrm{N}}} \mathcal{F}_x$ has *statistical* target collision resistance in the sense of Definition 3.5.34. That is, there exists a negligible function ε such that for every A, the following holds for all $x \in \Pi_{\mathrm{Y}}$ and every $y^* \in \{0,1\}^{n(|x|)}$:

$$\Pr_{f\leftarrow\mathcal{F}_x} \left[A(1^{|x|}, f) = y \text{ such that } f(y) = f(y^*) \text{ and } y \neq y^*\right] \leq \varepsilon(|x|) \ .$$

REMARK 3.6.6

In the above definition of an instance-dependent universal one-way hash family, we allow $m(|x|) > n(|x|)$, and only insist that the family has the large preimages property on the YES instances. In fact, our construction of an instance-dependent universal one-way hash family for $\overline{\mathrm{EA}}$' will be such that $m(|x|)$ is much larger than $n(|x|)$.

In addition, we insist that $\mathcal{F}_{\mathrm{N}}$ to have *statistical* target collision resistance because we want to achieve instance-dependent commitments that are *statistically binding* (in addition to being statistically hiding). As mentioned previously in Section 3.5.5, it is impossible for a single family of functions $\mathcal{F}$ (i.e., one that is not instance-dependent) to achieve both the the large preimages and the statistical target collision resistance properties.

Proposition 3.5.42 from Section 3.5.5 can be extended to account for instance-dependent primitives as follows.

PROPOSITION 3.6.7

Let HR-FullTransform be the algorithm stated in Proposition 3.5.42. For any problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}})$, if the following two conditions hold:

- Family of functions $\mathcal{F} = \bigcup_x \mathcal{F}_x$ is an instance-dependent universal one-way hash family for Π;

- Scheme $(\mathsf{S}_x, \mathsf{R}_x)$ is an instance-dependent 2-phase commitment scheme for Π that is statistically hiding on the YES instances, and statistically 1-out-of-2 binding on NO instances.

Then, scheme $(S_x, R_x) = \text{HR-FullTransform}((\mathsf{S}_x, \mathsf{R}_x), \mathcal{F}_x)$ is an instance-dependent commitment scheme for Π that is statistically hiding on the YES instances, and statistically binding on NO instances. Moreover, (S_x, R_x) is public coin if $(\mathsf{S}_x, \mathsf{R}_x)$ is.

Based on the above proposition, it suffices to construct an instance-dependent universal one-way hash family for $\overline{\text{EA'}}$ in order to get instance-dependent commitments for $\overline{\text{EA'}}$.

Instance-dependent universal one-way hash family for $\overline{\textbf{EA}}$. Although we just need an instance-dependent universal one-way hash family for $\overline{\text{EA'}}$, we will construct one for the slightly more general problem of $\overline{\text{EA}}$ (note that $\overline{\text{EA'}}$ is polynomial-time reducible to $\overline{\text{EA}}$).

Working directly with $\overline{\text{EA}}$ is difficult since we do not know any structure of the random variable X, other than its entropy bound. So to get more structure out of the random variable, we will *flatten* random variable X by taking multiple copies of X and outputting all of them—in other words, we are taking a direct product of multiple independent copies of X. Let X' denote this new random variable. Doing this would make the probability mass of X' concentrated around $2^{-H(X')}$, and this is why we call it flattening the random variable. (This is also known as the Asymptotic Equipartition Property in the information theory literature; see [CT].) Following Goldreich and Vadhan [GV], we give a definition of *flatness* as follows:

DEFINITION 3.6.8

Random variable X is ***δ-flat*** if for every $t \geq 1$,

$$\Pr_{x \leftarrow X}\left[2^{-t\cdot\delta} < \frac{\Pr[X = x]}{2^{H(X)}} < 2^{t\cdot\delta}\right] > 1 - 2^{-t^2} \quad .$$

Consider the flattened version of the Entropy Approximation problem, denoted as $\textsc{Flat}\text{EA} = (\textsc{Flat}\text{EA}_\text{Y}, \textsc{Flat}\text{EA}_\text{N})$, and defined as follows:

$$\textsc{Flat}\text{EA}_\text{Y} = \{(X,t) : \text{H}(X) \geq t + n^{14/15} \text{ and } X \text{ is } n^{8/15}\text{-flat}\}$$
$$\textsc{Flat}\text{EA}_\text{N} = \{(X,t) : \text{H}(X) \leq t \text{ and } X \text{ is } n^{8/15}\text{-flat}\}$$

It is clear that $\textsc{Flat}$EA is polynomial time reducible to EA, and the reverse reduction from EA to $\textsc{Flat}$EA follows from the Flattening Lemma of Goldreich and Vadhan [GV, Lem. 3.5]. Hence, constructing an instance-dependent universal one-way hash family for $\overline{\text{EA}}$ is equivalent to constructing one for $\overline{\textsc{Flat}\text{EA}}$, and we do this next.

THEOREM 3.6.9

The complement of the flattened version of the $\textsc{Entropy Approximation}$ problem, namely $\overline{\textsc{FlatEA}} = (\overline{\textsc{FlatEA}}_{\mathrm{Y}}, \overline{\textsc{FlatEA}}_{\mathrm{N}}) = (\textsc{FlatEA}_{\mathrm{N}}, \textsc{FlatEA}_{\mathrm{Y}})$ has an instance-dependent universal one-way hash family.

In the remaining of this section, we abuse notation by denoting the circuit encoding random variable X as $X\colon \{0,1\}^n \to \{0,1\}^m$.

Proof Idea of Theorem 3.6.9

For problem $\textsc{FlatEA}$, we will need to construct an instance-dependent (family of) functions that have target collision resistance on the YES instances and large preimages property on the NO instances. These are reversed properties because we want to prove that the complement $\overline{\textsc{FlatEA}}$ has an instance-dependent universal one-way hash family.

For the YES instances of $\textsc{FlatEA}$, X has entropy at least $t + \gamma$, where $\gamma = n^{14/15}$. Since X is a *nearly-flat* random variable, most of its preimages are small, i.e., their sizes are $\lesssim 2^{n-t-\gamma}$. So with high probability over a random $y \leftarrow \{0,1\}^n$, the preimage size of $X(y)$ is $\lesssim 2^{n-t-\gamma}$. By applying a pairwise-independent hash $h\colon \{0,1\}^n \to \{0,1\}^\beta$ to y, for $\beta \gtrsim n - t - \gamma$, it would make the function $g_h(y) = (X(y), h(y))$ ***almost injective***, in that for almost every element in the range has a unique preimage. (An injective function is, by definition, collision resistant.)

The adversary, however, need not choose y uniformly at random; in particular, it could choose an element y such that $X^{-1}(X(y))$ is large, making $f(y) = (X(y), h(y))$ no longer injective. To prevent the adversary from gaining, we add a *shift* $s \in \{0,1\}^n$ to the circuit X. Specifically, let the new function be $f_{s,h}(y) = (X(y \oplus s), h(y))$. Since y is now randomly shifted by s, the preimage size of $X(y \oplus s)$ is small with high probability over a random $s \leftarrow \{0,1\}^n$. Thus, we can conclude that $f_{s,h}(y)$ is *almost injective* even for an adversarially chosen y. This will give us the desired target collision resistance property for $\beta \gtrsim n - t - \gamma$.

For the NO instances of $\textsc{FlatEA}$, X has entropy at most t. Since X is a *nearly-flat* random variable, most of its preimages are large, i.e., their sizes are $\gtrsim 2^{n-t}$. Restricting to a hash $h\colon \{0,1\}^n \to \{0,1\}^\beta$ will shrink the size of the preimages by a factor of approximately $2^{-\beta}$. So if $\beta \lesssim n - t$, the size of the preimages will still be large enough to satisfy the large preimages property.

The entropy gap of $\gamma = n^{14/15}$ between the YES and NO instances allows us to find an appropriate value of β between $n - t - \gamma$ and $n - t$ that satisfies both cases.

Proof of Theorem 3.6.9

Based on the proof idea presented above, our instance-dependent universal one-way hash family for $\overline{\text{FLATEA}}$ is as follows:

$$\mathcal{F} = \bigcup_{(X,t)} \mathcal{F}_{(X,t)} = \left\{ f_{s,h} \colon \{0,1\}^n \to \{0,1\}^{m+\beta} \right\},$$

where $s \in \{0,1\}^n$, $\beta = n - t - 3n^{9/15}$, h is from a pairwise-independent hash family $\mathcal{H} = \{h \colon \{0,1\}^n \to \{0,1\}^\beta\}$, and $f_{s,h}(y) = (X(y \oplus s), h(y))$ for all $y \in \{0,1\}^n$.

We divide the proof of Theorem 3.6.9 into Lemmas 3.6.10 and 3.6.12 that establish the large preimages and the statistical target collision resistance properties of $\mathcal{F}$, respectively.

LEMMA 3.6.10

For every $(X,t) \in \text{FLATEA}_\text{N}$, the family $\mathcal{F}_{(X,t)}$ has the large preimages property in the sense of Definition 3.5.33.

Proof. For the NO instances of FLATEA, that is when $(X,t) \in \text{FLATEA}_\text{N}$, we know that $\text{H}(X) \le t$ and X is $n^{8/15}$-flat. Define a new circuit $X_s(y) = X(y \oplus s)$, and since $X_s(U_n)$ and $X(U_n)$ are identically distributed, X_s inherits all its statistical properties from X. Therefore, for any fixed $s \in \{0,1\}^n$, we have that

$$\Pr_{y \leftarrow \{0,1\}^n}\left[\left|X_s^{-1}(X_s(y))\right| \ge 2^{n-t-n^{9/15}}\right] \ge 1 - 2^{-n^{2/15}} = 1 - \text{neg}(n). \tag{3.14}$$

Bounding the size of $\left|X_s^{-1}(X_s(y))\right|$ is not sufficient since $f_{s,h}$ leaks the value of $h(y)$ too. Thus what we need is to bound the size of $h^{-1}(h(y)) \cap X_s^{-1}(X_s(y)) = f_{s,h}^{-1}(f_{s,h}(y))$, and the following claim provides us a way to do so.

CLAIM 3.6.11

For any set $S \subseteq \{0,1\}^n$, any function $h \colon \{0,1\}^n \to \{0,1\}^\beta$, and any $c > 0$,

$$\Pr_{y \leftarrow S}\left[\frac{\left|h^{-1}(h(y)) \cap S\right|}{|S|} \le 2^{-\beta-c}\right] \le 2^{-c}.$$

Proof of Claim. The above probability can be written as:

$$\sum_{\gamma \in \{0,1\}^\beta} \Pr_{y \leftarrow S}\left[h(y) = \gamma \text{ and } \left|h^{-1}(\gamma) \cap S\right| \le 2^{-\beta-c}|S|\right] \le 2^\beta \cdot 2^{-\beta-c} = 2^{-c}. \qquad \square$$

To apply the above claim, observe that instead of picking $y \leftarrow \{0,1\}^n$, an equivalent way would be to select an output x of X weighted according to the size of its preimage, and then pick a random preimage of x (namely, sample a uniform element from $X^{-1}(x)$). Now we can apply Claim 3.6.11 to (3.14) and obtain

$$\Pr_{y \leftarrow \{0,1\}^n}\left[\left|h^{-1}(h(y)) \cap X^{-1}(X(y \oplus s))\right| \geq 2^{n-t-n^{9/15}-\beta-c}\right] \geq 1 - \mathrm{neg}(n) - 2^{-c},$$

for any fixed $s \in \{0,1\}^n$ and $h \in \mathcal{H}$. Since $\beta = n - t - 3n^{9/15}$, and we set $c = n^{9/15}$, the size of $f_{s,h}^{-1}(f_{s,h}(y)) = h^{-1}(h(y)) \cap X^{-1}(X(y \oplus s))$ is at least $2^{n^{9/15}}$ with probability $1 - \mathrm{neg}(n)$. □

LEMMA 3.6.12

For every $(X, t) \in \textsc{FlatEA}_{\mathrm{Y}}$, the family $\mathcal{F}_{(X,t)}$ has the statistical target collision resistance property in the sense of Definition 3.5.34.

Proof. For the YES instances of FLATEA, that is when $(X, t) \in \textsc{FlatEA}_{\mathrm{Y}}$, we know that $\mathrm{H}(X) \geq t + n^{14/15}$ and X is $n^{8/15}$-flat. Therefore, for any $y \in \{0,1\}^n$ chosen in advance,

$$\Pr_{s \leftarrow \{0,1\}^n}\left[\left|X^{-1}(X(y \oplus s))\right| \leq 2^{n-t-n^{14/15}+n^{9/15}}\right] \geq 1 - 2^{-n^{2/15}} = 1 - \mathrm{neg}(n). \tag{3.15}$$

The value of $X(y \oplus s)$ alone does not bind y, but since the number of its preimages is small (with high probability), a pairwise independent hash would uniquely determine y. The next claim captures this fact.

CLAIM 3.6.13

Let random variable H denote a random hash function from a family of pairwise-independent hash functions $\mathcal{H}$ mapping $\{0,1\}^n$ to $\{0,1\}^\beta$. For any subset $S \subseteq \{0,1\}^n$ and any element $y \in \{0,1\}^n$, we have:

$$\Pr\left[\exists y' \in S \setminus \{y\} \text{ such that } H(y') = H(y)\right] \leq |S| \cdot 2^{-\beta}.$$

Proof of Claim. The pairwise independent property of $\mathcal{H}$ guarantees that for any $y' \neq y$, we have $\Pr[H(y') = H(y)] \leq 2^{-\beta}$. Taking a union bound over all possible values of $y' \in S \setminus \{y\}$ yields our claim. □

We say $X^{-1}(X(y \oplus s))$ is ***large*** if $\left|X^{-1}(X(y \oplus s))\right| > 2^{n-t-n^{14/15}+n^{9/15}}$, and ***small*** otherwise. Because $f_{s,h}(y) = f_{s,h}(y')$ if only if both $X(y \oplus s) = X(y' \oplus s)$ and $h(y) = h(y')$, applying Claim 3.6.13 to (3.15), we have that for any fixed y,

$$\Pr\left[X^{-1}(X(y \oplus U_n)) \text{ is } \textit{large}\right] = \mathrm{neg}(n) \ . \tag{3.16}$$

Thus, we can bound the probability of finding a collision with any fixed y as follows:

$$\begin{aligned}
&\Pr[\exists y' \neq y \text{ such that } f_{U_n,H}(y) = f_{U_n,H}(y')] \\
&\quad \leq \begin{array}{l} \quad\Pr\left[X^{-1}(X(y \oplus U_n)) \text{ is large}\right] \\ + \Pr\left[\exists y' \neq y \text{ such that } f_{U_n,H}(y) = f_{U_n,H}(y') \mid X^{-1}(X(y \oplus U_n) \text{ is small}\right] \end{array} \\
&\quad \leq \quad \text{neg}(n) + 2^{n-t-n^{14/15}+n^{9/15}} \cdot 2^{-\beta} \qquad\qquad \text{(by 3.16)} \\
&\quad = \quad \text{neg}(n) + 2^{-n^{14/15}+4n^{9/15}} \ ,
\end{aligned}$$

with the final equality following from setting $\beta = n - t - 3n^{9/15}$. Since $2^{-n^{14/15}+4n^{9/15}}$ is negligible, our proof is complete. □

Because $\overline{\text{EA'}}$ polynomial-time reduces to $\overline{\text{EA}}$, which in turn is polynomial-time equivalent to $\overline{\text{FlatEA}}$, we arrive at the following corollary of Theorem 3.6.9.

COROLLARY 3.6.14

Both problems $\overline{\text{EA}}$ and $\overline{\text{EA'}}$ have instance-dependent universal one-way hash families.

Putting it all together

We restate Theorem 3.0.5 and prove it.

RESTATEMENT OF THEOREM 3.0.5

For every problem $\Pi \in$ SZKP, problem Π has an instance-dependent commitment scheme that is *statistically hiding* on the YES instances and *statistically binding* on the NO instances. Moreover, the instance-dependent commitment scheme obtained is public coin and is constant round.

Proof of Theorem 3.0.5. Proposition 3.6.1 informs us that to construct instance-dependent commitment schemes for all problems in SZKP, it suffices to construct instance-dependent commitment schemes for EA' and its complement $\overline{\text{EA'}}$, recalling that EA' is the restricted version of Entropy Approximation.

Lemma 3.6.3 and 3.6.4 provides instance-dependent commitments for EA' and instance-dependent *1-out-of-2-binding* commitment schemes for $\overline{\text{EA'}}$, respectively. (Both these schemes are due to Nguyen and Vadhan [NV].) We also know that $\overline{\text{EA'}}$ has an instance-dependent universal one-way hash family, given by Corollary 3.6.14. Therefore, Proposition 3.6.7 tells us that we can use the Haitner & Reingold transformation [HR2] to convert the instance-dependent 1-out-of-2-binding commitments for $\overline{\text{EA'}}$ into instance-dependent commitments (with the standard binding property) for $\overline{\text{EA'}}$.

The public coin property of our final scheme follows from the fact that the both schemes for EA' and $\overline{\text{EA'}}$ are public-coin schemes (refer to Lemmas 3.6.3 and 3.6.4, respectively).

Finally, observe that we obtain a constant-round scheme because both schemes for EA' and $\overline{\text{EA'}}$ are constant round (refer to Lemmas 3.6.3 and 3.6.4, respectively), and the added-on round complexity due to the Haitner & Reingold transformation is only a constant. □

Conclusion. We have shown how to construct instance-dependent commitment schemes for all of SZKP that are statistically hiding on the YES instances and statistically binding on the NO instances. In the next chapter, we extend our result to the other three variants of zero knowledge—namely, computational zero-knowledge proofs (CZKP), statistical zero-knowledge arguments (SZKA), and computational zero-knowledge arguments (CZKA)—showing that all of them have instance-dependent commitments with corresponding security properties. This means that we can obtain instance-dependent commitments from zero-knowledge protocols, and having this ability allows us to establish various unconditional relationships between the different formulations of zero knowledge, as done in Chapter 4.

4

UNCONDITIONAL CHARACTERIZATIONS OF ZERO KNOWLEDGE

In this chapter, we give *unconditional* characterizations of classes of problems having zero-knowledge argument systems using the Vadhan condition. These characterizations would, among other things, allow us to establish the main unconditional results of this dissertation, which are equivalences between instance-dependent commitments and zero-knowledge protocols (Theorem 1.2.4), symmetry between computational zero knowledge and computational soundness (Theorem 1.2.2), and a method of transforming any honest-verifier zero-knowledge argument system into a malicious-verifier zero-knowledge argument system for the same problem. Recall that all these results were highlighted in Section 1.2 of Chapter 1.

For context, we restate the Vadhan condition from Section 1.2 in the terms of instance-dependent one-way functions (Definition 2.4.6).

RESTATEMENT OF DEFINITION 1.2.3
A promise problem $\Pi = (\Pi_\mathrm{Y}, \Pi_\mathrm{N})$ satisfies the ***Vadhan condition*** if there exists a set of instances $I \subseteq \Pi_\mathrm{Y} \cup \Pi_\mathrm{N}$ such that:

- the promise problem $(\Pi_\mathrm{Y} \setminus I, \Pi_\mathrm{N} \setminus I)$ is in SZKP, and
- there exists an instance-dependent one-way function on I.

We call I the set of ***OWF instances***, $I \cap \Pi_\mathrm{Y}$ the set of ***OWF YES instances***, and $I \cap \Pi_\mathrm{N}$ the set of ***OWF NO instances***.

Chapter organization. In the next section, we present our main characterization theorems, which expands upon Theorem 1.2.4. The steps involved in proving these characterization theorems are outlined in the beginning of Section 4.2, and lemmas needed to establish these theorems are given in Sections 4.2.1, 4.2.2, and 4.2.3. Finally, in Section 4.3, we prove our Symmetry Theorem (Theorem 1.2.2) between computational zero knowledge and computational soundness, and show an interesting consequence of our Symmetry Theorem.

4.1 The Characterization Theorems

In this section, we expand upon the characterizations of zero-knowledge protocols in terms of the Vadhan condition given by Theorem 1.2.4, and the equivalences between instance-dependent commitments and zero-knowledge protocols given by Theorem 1.2.1. Specifically, we state four theorems giving a variety of equivalent characterizations of the zero-knowledge complexity classes SZKP, CZKP, CZKA, and SZKA. The ones for zero-knowledge arguments, namely CZKA and SZKA, are new; the other for zero-knowledge proofs, namely CZKP and SZKP, contain results from previous work, but are given for comparison. In addition to establishing Theorems 1.2.1 1.2.4, these theorems show an equivalence between problems having only honest-verifier zero-knowledge protocols, problems satisfying the Vadhan condition, and problems with (malicious-verifier) zero-knowledge protocols having desirable properties like an efficient prover, perfect completeness, public coins, and black-box simulation. It should be noted that these characterizations refer only to the classes of problems, and do not necessarily preserve other efficiency measures like round complexity, unless explicitly mentioned.

The following two theorems give unconditional characterizations of zero-knowledge proofs.

THEOREM 4.1.1

(SZKP Characterization Theorem, containing results from [Oka, GSV1, NV].) For every problem $\Pi \in \mathrm{IP}$, the following conditions are equivalent.

1. $\Pi \in$ HV-SZKP.

2. Π satisfies the Vadhan condition without OWF instances.

3. Π has an instance-dependent commitment scheme that is statistically hiding on the YES instances and statistically binding on the NO instances. Moreover, the scheme is public coin and constant round.

4. $\Pi \in$ SZKP, and the statistical zero-knowledge proof system for Π has a black-box simulator, is public coin, and has perfect completeness. Furthermore, if $\Pi \in \mathrm{NP}$, the proof system has an efficient prover and is constant round with a polynomially-small soundness error.[1]

[1] To get negligible soundness error with $\omega(1)$ rounds, we repeat the protocol $\omega(1)$ times sequentially.

THEOREM 4.1.2

(CZKP Characterization Theorem, containing results from [Vad3, NV].) For every problem $\Pi \in \mathrm{IP}$, the following conditions are equivalent.

1. $\Pi \in$ HV-CZKP.
2. Π satisfies the Vadhan condition without OWF NO instances.
3. Π has an instance-dependent commitment scheme that is computationally hiding on the YES instances and statistically binding on the NO instances. Moreover, the scheme is public coin and constant round.
4. $\Pi \in$ CZKP, and the computational zero-knowledge proof system for Π has a black-box simulator, is public coin, and has perfect completeness. Furthermore, if $\Pi \in \mathrm{NP}$, the proof system has an efficient prover and is constant round with a polynomially-small soundness error.[1]

REMARK 4.1.3

The new result in both Theorems 4.1.1 and 4.1.2 is the (2) ⇒ (3) direction, establishing instance-dependent commitments (with an efficient sender strategy and a regular binding property) from the Vadhan condition. The previous works of Vadhan [Vad3], and Nguyen and Vadhan [NV] result in an inefficient sender strategy and a weaker *1-out-of-2* binding property, respectively.

The (1) ⇒ (2) direction for the SZKP case (Theorem 4.1.1) and the CZKP case (Theorem 4.1.2) follows from [Oka, GSV1] and [Vad3], respectively. The (2) ⇒ (4) direction, in both the SZKP and CZKP cases, was established by [NV] using instance-dependent commitments with the weaker 1-out-2-binding property. Because of this weaker binding property, their zero-knowledge proof systems have polynomial number of rounds. We obtain constant-round instance-dependent commitments with the standard binding property, based on the (2) ⇒ (3) direction, and hence we are able to achieve constant-round zero-knowledge proofs with a polynomially-small soundness error.

We give analogous characterizations for zero-knowledge *arguments*.

THEOREM 4.1.4

(SZKA Characterization Theorem.) For every problem $\Pi \in \mathrm{NP}$, the following conditions are equivalent.

1. $\Pi \in$ HV-SZKA.
2. Π satisfies the Vadhan condition without OWF YES instances.

3. Π has an instance-dependent commitment scheme that is statistically hiding on the YES instances and computationally binding on the NO instances. Moreover, the scheme is public coin.
4. $\Pi \in \mathrm{SZKA}$, and the statistical zero-knowledge argument system for Π has a black-box simulator, is public coin, has perfect completeness, and an efficient prover.

THEOREM 4.1.5

(CZKA Characterization Theorem.) For every problem $\Pi \in \mathrm{NP}$, the following conditions are equivalent.

1. $\Pi \in$ HV-CZKA.
2. Π satisfies the Vadhan condition.
3. Π has an instance-dependent commitment scheme that is computationally hiding on the YES instances and computationally binding on the NO instances. Moreover, the scheme is public coin.
4. $\Pi \in \mathrm{CZKA}$, and the computational zero-knowledge proof system for Π has a black-box simulator, is public coin, has perfect completeness, and an efficient prover.

We prove Theorems 4.1.1, 4.1.2, 4.1.4, and 4.1.5 using lemmas established in Sections 4.2.1, 4.2.2, and 4.2.3. Notice that in the theorems involving zero knowledge arguments, we have restricted the problem Π to be in NP in contrast to the theorems involving zero-knowledge proofs (Theorems 4.1.1 and 4.1.2), which are naturally restricted to IP. The reason for this is that argument systems are mainly interesting when the honest prover runs in polynomial time given a witness for membership (otherwise the protocol would not even be sound against prover strategies with the same resources as the honest prover), and such efficient provers only make sense for problems in NP (or actually, MA, to which our results generalize easily). In fact our theorems above show that for problems in NP, a zero-knowledge protocol without an efficient prover can be converted into one with an efficient prover (by the equivalence of Items 1 and 4 in Theorems 4.1.1 to 4.1.4 above).

4.2 Proof of the Characterization Theorems

We provide an outline of the steps involved in proving the characterization theorems stated in the previous section.

1. In Section 4.2.1, we show that every problem Π possessing a (honest-verifier) zero-knowledge protocol satisfies the Vadhan condition. Depending on the zero knowledge and soundness guarantee, the types of Vadhan condition that Π satisfies will differ (in whether the sets of OWF YES instances and OWF NO instances are empty

or nonempty). This result extends the unconditional characterization work of Vadhan [Vad3] for zero-knowledge proof systems to the more general zero-knowledge argument systems.

2. Next, in Section 4.2.2, we show that every problem Π satisfying the Vadhan condition yields an instance-dependent commitment scheme for Π.

3. Finally, in Section 4.2.3, we show that every problem $\Pi \in \mathrm{NP}$ having instance-dependent commitments allow us to construct zero-knowledge argument systems for Π with desirable properties like perfect completeness, black-box zero knowledge, public coins, and an efficient prover. This is achieved by substituting instance-dependent commitments for standard commitments used in existing zero-knowledge protocols. (This technique of substituting instance-dependent commitments for standard commitments is detailed in Section 2.5.)

A summary of the steps involved in establishing our characterization theorems, together with their corresponding lemmas, is given in Figure 4.1.

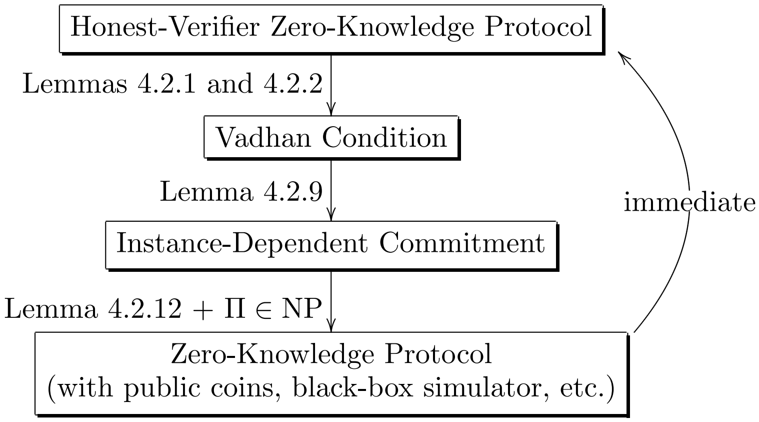
Honest-Verifier Zero-Knowledge Protocol
Lemmas 4.2.1 and 4.2.2
Vadhan Condition
Lemma 4.2.9
Instance-Dependent Commitment
Lemma 4.2.12 + Π ∈ NP
Zero-Knowledge Protocol
(with public coins, black-box simulator, etc.)
immediate

Figure 4.1: Steps of our proof.

4.2.1 From zero-knowledge protocols to the Vadhan condition

In this subsection, we show that problems possessing (honest verifier) zero-knowledge arguments satisfy the Vadhan condition. Specifically, we prove that for every problem Π having a zero-knowledge argument also satisfies the Vadhan condition. This involving establishing a set of instances $I \subseteq \Pi_{\mathrm{Y}} \cup \Pi_{\mathrm{N}}$ such that $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus I) \in \mathrm{SZKP}$, and from which instance-dependent one-way functions can be constructed. The main difference from Vadhan [Vad3] is that [Vad3] characterizes only zero-knowledge proofs and it without OWF NO instances, namely $I \cap \Pi_{\mathrm{N}} = \emptyset$. In other words, the characterizations of [Vad3] satisfy the Vadhan condition without OWF NO instances.

We state a lemma establishing characterizations of (honest verifier) zero-knowledge *proofs* in terms of the Vadhan condition. This lemma follows from the works of [Oka, GSV1, Vad3], but is given for comparison.

LEMMA 4.2.1

(Follows from [Oka, GSV1, Vad3].) If problem $\Pi \in$ HV-CZKP, then Π satisfies the Vadhan condition without OWF NO instances, namely $I \cap \Pi_{\mathrm{N}} = \emptyset$. In addition, if $\Pi \in$ HV-SZKP, then Π satisfies the Vadhan condition without OWF instances, namely $I = \emptyset$.

Next, we give analogous characterizations for (honest verifier) zero-knowledge *arguments.*

LEMMA 4.2.2

If problem $\Pi \in$ HV-CZKA, then Π satisfies the Vadhan condition. In addition, if $\Pi \in$ HV-SZKA, then Π satisfies the Vadhan condition without OWF YES instances, namely $I \cap \Pi_{\mathrm{Y}} = \emptyset$.

Proof Idea of Lemma 4.2.2

Proving that $\Pi \in$ HV-CZKA satisfies the Vadhan condition involves establishing a set I with an instance-dependent one-way on I and $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus I) \in$ SZKP. To do so, we provide a separate analysis for the YES and NO instances; namely, we show that there exist sets $I_{\mathrm{Y}} \subseteq \Pi_{\mathrm{Y}}$ and $I_{\mathrm{N}} \subseteq \Pi_{\mathrm{N}}$ such that instance-dependent one-way functions can be constructed on these sets, and that $(\Pi_{\mathrm{Y}} \setminus I_{\mathrm{Y}}, \Pi_{\mathrm{N}} \setminus I_{\mathrm{N}}) \in$ SZKP. These instance-dependent one-way functions f_x and g_x on I_{Y} and I_{N}, respectively, can be combined into a single instance-dependent one-way function on $I \stackrel{\mathrm{def}}{=} I_{\mathrm{Y}} \cup I_{\mathrm{N}}$ by concatenating the functions f_x and g_x.

The sets I_{Y} and I_{N} are defined based on the simulator S for the zero-knowledge protocol of $\Pi \in$ HV-CZKA. Following Fortnow [For], we consider a *simulation-based prover* P_S and corresponding *simulation-based verifier* V_S. Informally, P_S replies with the same conditional probability as the prover in the output of S, and V_S sends its messages with the same conditional probability as the verifier in the output of S. We make the following observations.

1. The interaction between P_S and V_S is identical to the output of the simulator S, on every x.
2. By the zero-knowledge condition, we have that $\langle P_S, V_S \rangle$ is computationally indistinguishable from $\langle P, V \rangle$, when $x \in \Pi_{\mathrm{Y}}$.
3. By assuming, without loss of generality, that the simulator always outputs accepting transcripts, it holds that P_S makes V_S accepts with probability 1, on every x.

We consider a statistical measure of how similar V_S is to V (on instance x, when interacting with simulation-based prover P_S). Using this statistical measure (given in the full proof below), we define sets I_{Y} and I_{N} as follows:

- I_{Y} contains instances $x \in \Pi_{\text{Y}}$ for which V_S is *statistically different* from V, and
- I_{N} contains instances $x \in \Pi_{\text{N}}$ for which V_S is *statistically similar* to V.

Now the proof that this gives a Vadhan condition proceed as follows:

1. On I_{Y}, we have that V_S is statistically different from V. Nevertheless, by the zero-knowledge condition (as noted above), V_S is computationally similar to V. This enables us to construct one-way functions for instances in I_{Y}, as shown in [Vad3].

2. On I_{N}, we have that V_S is statistically similar to V. Combining this with the fact that P_S will always convince V_S to accept (as noted above), we conclude that P_S convinces V to accept with high probability. By the computational soundness of (P, V), it must be the case that P_S is not PPT. Using techniques from Ostrovsky [Ost], this allows us to convert the simulator S into an instance-dependent distributional one-way function g_x.[2] Then by Proposition 2.4.8, due to Impagliazzo and Luby [IL], we can obtain an instance-dependent one-way function from g_x.

3. To see that $(\Pi_{\text{Y}} \setminus I_{\text{Y}}, \Pi_{\text{N}} \setminus I_{\text{N}}) \in$ SZKP, we observe the following: for those YES instances not in I_{Y} (i.e., instances in $\Pi_{\text{Y}} \setminus I_{\text{Y}}$), the simulated verifier V_S is statistically similar to V. And for those NO instances not in I_{N} (i.e., instances in $\Pi_{\text{N}} \setminus I_{\text{N}}$), the simulated verifier V_S is statistically different from V. This gap in the statistical properties allows us to reduce promise problem $(\Pi_{\text{Y}} \setminus I_{\text{Y}}, \Pi_{\text{N}} \setminus I_{\text{N}})$ to one of the complete problems for SZKP [SV, GV, Vad3].

Proof of Lemma 4.2.2

Let (P, V) be a zero-knowledge argument system for Π, with simulator S. Following Vadhan [Vad3], we modify our interactive protocol (P, V) to satisfy the following additional properties.

- The completeness error $c(|x|)$ and soundness error $s(|x|)$ are both negligible. This can be achieved by a standard error reduction via sequential repetition.
- On every input x, the two parties exchange $2\ell(|x|)$ messages for some polynomial ℓ, with the verifier sending even-numbered messages and sending all of its $r(|x|)$ random

[2] If g_x is not distributionally one-way, then P_S can be made to be efficient, hence contradicting the computational soundness of (P, V). Interestingly, Ostrovsky [Ost] uses the assumption that g_x is not distributionally one-way to invert the simulator S on the YES instances, and conclude that Π is not hard-on-average. Although we use similar techniques as [Ost], we instead invert S on the NO instances to contradict the computational soundness of (P, V).

coin tosses in the last message. (Without loss of generality, we may assume that $r(|x|) \geq |x|$.) Having the verifier send its coin tosses at the end does not affect soundness because it is after the prover's last message, and does not affect honest-verifier zero knowledge because the simulator is anyhow required to simulate the verifier's coin tosses.

- On every input x, the simulator S always outputs ***accepting transcripts***, where a simulator output τ is an accepting transcript on x if all of the verifier's messages in τ are consistent with its coin tosses (as specified in the last message), and the verifier would accept in such an interaction.

For a transcript τ, we denote by τ_i the *prefix* of τ consisting of the first i messages. For readability, we often drop the input x from the notation, for instance using $\ell = \ell(|x|)$, $\langle P, V\rangle = \langle P, V\rangle(x)$, $r = r(|x|)$, and so forth. Thus, in what follows, $\langle P, V\rangle_i$ and S_i are random variables representing prefixes of transcripts generated by the real interaction and simulator, respectively, on a specified input x.

Using the simulator S, we define the simulation-based prover P_S as follows: On input x and execution prefix τ_{2i}, for $i = 1, 2, \ldots, \ell - 1$, do the following:

1. If simulator $S(x)$ outputs a transcript that begins with τ_{2i} with probability 0, then P_S replies with a dummy message.

2. Otherwise, P_S replies according with the same conditional probability as the prover in the output of the simulator. That is, it replies with a string α with probability $p_\alpha = \Pr\left[S(x)_{2i} = \tau_{2i-1} \circ \alpha | S(x)_{2i-1} = \tau_{2i-1}\right]$.

The simulation-based verifier V_S can be defined analogously as follows: On input x and execution prefix τ_{2i-1}, for $i = 1, 2, \ldots, \ell$, do the following:

1. If simulator $S(x)$ outputs a transcript that begins with τ_{2i-1} with probability 0, then V_S replies with a dummy message.

2. Otherwise, V_S replies according with the same conditional probability as the verifier in the output of the simulator. That is, it replies with a string β with probability $p_\beta = \Pr\left[S(x)_{2i+1} = \tau_{2i} \circ \beta | S(x)_{2i} = \tau_{2i}\right]$.

Observe that $\langle P_S, V_S\rangle(x)$ is identically distributed to $S(x)$, for every x. Following [AH, PT, GV, Vad3], we consider the following quantity:

$$h(x) = \sum_{i=1}^{\ell} \left[\mathrm{H}(S(x)_{2i}) - \mathrm{H}(S(x)_{2i-1})\right] = \sum_{i=1}^{\ell} \left[\mathrm{H}(\langle P_S, V_S\rangle(x)_{2i}) - \mathrm{H}(\langle P_S, V_S\rangle(x)_{2i-1})\right] \quad , \tag{4.1}$$

recalling that $\mathrm{H}(\cdot)$ is the *(Shannon) entropy* measure.

From [AH, PT, GV], we know that for every $x \in \{0,1\}^*$, and every prover strategy P',

$$r(|x|) = \sum_{i=1}^{\ell} \left[\mathrm{H}(\langle P', V\rangle(x)_{2i}) - \mathrm{H}(\langle P', V\rangle(x)_{2i-1}) \right] \quad . \tag{4.2}$$

The above sum in (4.2) measures the total entropy contributed by the honest verifier's messages, and hence it is natural that this should equal $r(|x|)$, the number of coin tosses of the honest verifier. This is because the honest verifier reveals all its coin tosses at the end.

From (4.1) and (4.2), we observe that how close the value of $h(x)$ gets to $r(|x|)$ is a measure of how close the simulation-based verifier V_S is from the honest verifier V (when interacting with P_S). Following our intuition in the proof sketch above, we let I_{Y} be the set of instances $x \in \Pi_{\mathrm{Y}}$ for which the V_S is *far* from the honest verifier V, and we let I_{N} be the set of instances $x \in \Pi_{\mathrm{N}}$ for which the V_S is *close* to V. Formally, we define:

$$I_{\mathrm{Y}} = \{x \in \Pi_{\mathrm{Y}} : h(x) < r(|x|) - 1/q(|x|)\} \ ;$$
$$I_{\mathrm{N}} = \{x \in \Pi_{\mathrm{N}} : h(x) > r(|x|) - 2/q(|x|)\} \ ,$$

where the polynomial $q(|x|) = 256 \cdot \ell(|x|)$.

Having defined sets I_{Y} and I_{N}, Lemma 4.2.2 is established by the following claims. The first three are proven in the same way as in [Vad3], and hence we defer their proofs to Appendix A.3.

CLAIM 4.2.3

Problem $(\Pi_{\mathrm{Y}} \setminus I_{\mathrm{Y}}, \Pi_{\mathrm{N}} \setminus I_{\mathrm{N}}) \in \mathrm{SZKP}$.

CLAIM 4.2.4

There exists an instance-dependent one-way function on I_{Y}.

CLAIM 4.2.5

For $\Pi \in \mathrm{HV\text{-}SZKA}$, we can take $I_{\mathrm{Y}} = \emptyset$.

The main novelty in our analysis is the following claim.

CLAIM 4.2.6

There exists an instance-dependent one-way function on I_{N}.

Proof of Claim. To get an instance-dependent one-way function on I_{N}, we use the following idea of Ostrovsky [Ost]: if we can invert the simulator, then P_S's replies can be approximated efficiently. By the computational soundness of (P, V), this is impossible, so the simulator must be a one-way function. More precisely, we define the function g_x, whose purpose is to output the messages of the simulator, as follows:

$$g_x(i, \omega) = (x, i, S(x; \omega)_{2i}) \ . \tag{4.3}$$

Note that g_x is polynomial-time computable because the simulator S runs in polynomial time. If g_x is *not* distributionally one-way (in the sense of Definition 2.4.7), then we can devise an efficient cheating prover strategy, call it $\widetilde{P}$, that *efficiently approximates* our simulation-based prover P_S upto negligible statistical error. The way to do this is to feed a given transcript prefix τ_{2i} after the verifier has responded in round $2i$, into the inversion algorithm of g_x to obtain the simulation-based prover response for round $2i+1$. In doing so, we contradict the computational soundness property of (P, V). This argument is captured by following proposition, whose proof is given in Appendix A.3.

PROPOSITION 4.2.7

(Based on [Ost, Lem. 1].) Let g_x be as in (4.3). For every set $K \subseteq \{0,1\}^*$, if g_x is *not* an instance-dependent distributionally one-way function on K, then for every polynomial p, there exists a nonuniform PPT prover $\widetilde{P}$ such that

$$\Delta(\langle \widetilde{P}, V\rangle(x), S(x)) \leq \ell(|x|) \cdot \left(\frac{1}{p(|x|)} + 2 \cdot \Delta(\langle P_S, V\rangle(x), S(x)) \right) \ ,$$

for infinitely many $x \in K$.

Our main goal is to upper bound $\Delta(\langle \widetilde{P}, V\rangle, S)$ because doing so would contradict the computational soundness of V: by virtue of the fact that S always outputs accepting transcripts, if $\langle \widetilde{P}, V\rangle$ is close to S, then the nonuniform PPT $\widetilde{P}$ will convince V to accept with noticeable probability. The above proposition tells us that in order to obtain an upper bound on $\Delta(\langle \widetilde{P}, V\rangle, S)$, we just need to upper bound $\Delta(\langle P_S, V\rangle, S)$, which we do next.

Recall that for every $x \in I_{\mathrm{N}}$, we have $h > r - 2/q$. From [AH, PT, GV], we know that $h = r - \mathrm{KL}(\langle P_S, V\rangle, S)$, where KL is the ***Kullback-Leibler*** distance defined as $\mathrm{KL}(X, Y) = \mathrm{E}_{\alpha \leftarrow X}\big[\log(\Pr[X = \alpha]) - \log(\Pr[Y = \alpha])\big]$. (See [GV, Lem. 2.2].) Hence, we get $\mathrm{KL}(\langle P_S, V\rangle, S) < 2/q$. Using the fact that for any random variables X and Y, $\mathrm{KL}(X, Y) \geq (1/2) \cdot (\Delta(X, Y))^2$ [CT, Lem. 12.6.1], we get that for all $x \in I_{\mathrm{N}}$,

$$\Delta(\langle P_S, V\rangle, S) < 2/\sqrt{q} = 1/(8 \cdot \ell) \ , \tag{4.4}$$

since $q = 256 \cdot \ell$.

Now by Proposition 4.2.7, if g_x is not distributionally one-way on I_{N}, we can take $I_{\mathrm{N}} = K$

and choose $p(|x|) = 4 \cdot \ell(|x|)$, to get a nonuniform PPT $\widetilde{P}$ such that

$$\begin{aligned} \Delta(\langle \widetilde{P}, V \rangle, S) &\leq \ell \cdot (1/p + 2 \cdot \Delta(\langle P_S, V \rangle, S)) \\ &= 1/4 + 2 \cdot \ell \cdot \Delta(\langle P_S, V \rangle, S) \\ &< 1/2 \ . && \text{(by 4.4)} \end{aligned}$$

And since the simulator S always produce accepting transcripts, we have

$$\Pr[(\widetilde{P}, V)(x) = \texttt{accept}] \geq 1/2 \ ,$$

for infinitely many $x \in I_{\mathrm{N}}$. This contradicts the computational soundness of (P, V). Therefore, g_x must be a distributionally one-way function on I_{N}. By Proposition 2.4.8 (due to Impagliazzo and Luby [IL]), g_x can be converted into an instance-dependent (standard) one-way function on I_{N}, as desired. □

Let us see how the above five claims establish Lemma 4.2.2. Define set $I = I_{\mathrm{Y}} \cup I_{\mathrm{N}}$. This means that the promise problem $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus I) = (\Pi_{\mathrm{Y}} \setminus I_{\mathrm{Y}}, \Pi_{\mathrm{N}} \setminus I_{\mathrm{N}})$, and Claim 4.2.3 places this problem in SZKP. Claims 4.2.4 and 4.2.6 give us instance-dependent one-way functions on I_{Y} and I_{N}, respectively; to obtain a single instance-dependent one-way function on $I = I_{\mathrm{Y}} \cup I_{\mathrm{N}}$, we use the following claim.

CLAIM 4.2.8

For any sets $J, K \subseteq \{0,1\}^*$, if there exist instance-dependent one-way functions on J and there exist instance-dependent one-way functions on K, then there exist instance-dependent one-way functions on $J \cup K$.

Proof of Claim. Let f_x and g_x be any instance-dependent one-way function on J and K, respectively. Then, $h_x(y,z) = (f_x(y), g_x(z))$ is an instance-dependent one-way function on $J \cup K$. This is because inverting h_x involves inverting both f_x and g_x, at least one of which is hard to invert on $J \cup K$. □

Therefore, by Claim 4.2.8 above, we know that $\Pi \in$ HV-CZKA satisfies the Vadhan condition. Furthermore, if $\Pi \in$ HV-SZKA, Claim 4.2.5 tells us that $I_{\mathrm{Y}} = \emptyset$, and hence $I \cap \Pi_{\mathrm{Y}} = I_{\mathrm{Y}} = \emptyset$, giving us that Π satisfies the Vadhan condition without OWF YES instances. This completes our proof of Lemma 4.2.2.

4.2.2 From the Vadhan condition to instance-dependent commitments

In this subsection, we show that every problem Π satisfying the Vadhan condition yields an instance-dependent commitment scheme for Π. This is obtained by combining the statistically-binding commitments from one-way functions of [Nao, HILL], the statistically-hiding commitments from one-way functions of Theorem 3.0.4, and the instance-dependent commitments for SZKP of Theorem 3.0.5.

LEMMA 4.2.9

The following conditions hold for problems Π satisfying the Vadhan condition.

- SZKP case: if Π satisfies the Vadhan condition without OWF instances, then it has an instance-dependent commitment scheme that is statistically hiding on the YES instances and statistically binding on the NO instances. Moreover, this scheme is *constant round*.
- CZKP case: if Π satisfies the Vadhan condition without OWF NO instances, then it has an instance-dependent commitment scheme that is computationally hiding on the YES instances and statistically binding on the NO instances. Moreover, this scheme is *constant round*.
- SZKA case: if Π satisfies the Vadhan condition without OWF YES instances, then it has an instance-dependent commitment scheme that is statistically hiding on the YES instances and computationally binding on the NO instances.
- CZKA case: if Π satisfies the Vadhan condition, then it has an instance-dependent commitment scheme that is computationally hiding on the YES instances and computationally binding on the NO instances.

Furthermore, all the above instance-dependent commitment schemes are public coin.

The proof of Lemma 4.2.9, tying together all the following propositions and claims, is given at the end of this subsection. But first, we provide an outline of the steps of our construction in the next paragraph.

For a problem Π that satisfies the Vadhan condition, let I_Y and I_N be the set of OWF YES and OWF NO instances, respectively. We break the task of constructing an instance-dependent commitment scheme for a Π into following four steps: (i) construct a scheme that is hiding on $\Pi_\mathrm{Y} \setminus I_\mathrm{Y}$ and binding on $\Pi_\mathrm{N} \setminus I_\mathrm{N}$, (ii) construct a scheme that is hiding on I_Y and binding everywhere, (iii) construct a scheme that is hiding everywhere and binding on I_N, and (iv) combine all these three schemes into a single instance-dependent commitment scheme for Π. We will explain why these four steps yield an instance-dependent commitment scheme for Π in the proof of Lemma 4.2.9, given at the end of this subsection.

Step 1. Obtain a statistically-hiding and statistically-binding instance-dependent commitment scheme for problem $(\Pi_\mathrm{Y} \setminus I_\mathrm{Y}, \Pi_\mathrm{N} \setminus I_\mathrm{N}) \in \mathrm{SZKP}$ from Theorem 3.0.5, restated below.

RESTATEMENT OF THEOREM 3.0.5

Every problem in SZKP has an instance-dependent commitment scheme that is *statistically* *hiding* on the YES instances and *statistically binding* on the NO instances. Moreover, this instance-dependent commitment scheme is public coin and is constant round.

Step 2. From an instance-dependent one-way function on I_{Y}, apply Proposition 2.4.14 to get an instance-dependent commitment scheme that is computationally hiding on I_{Y} and statistically binding elsewhere.

RESTATEMENT OF PROPOSITION 2.4.14

(Follows from [Nao, HILL].) For every set $K \subseteq \{0,1\}^*$, if there is an instance-dependent one-way function on K, then problem $(K, \overline{K})$ has an instance-dependent commitment scheme that is computationally hiding on the YES instances (namely, instances in K), and statistically binding on the NO instances (namely, instances in $\overline{K}$). Moreover, the instance-dependent commitment scheme obtained is public coin and constant round.

Step 3. From an instance-dependent one-way function on I_{N}, apply Proposition 3.5.44 to get an instance-dependent commitment scheme that is computationally binding on I_{N} and statistically hiding elsewhere.

RESTATEMENT OF PROPOSITION 3.5.44

For every set $K \subseteq \{0,1\}^*$, if there is an instance-dependent one-way function on K, then problem $(\overline{K}, K)$ has an instance-dependent commitment that is statistically hiding on the YES instances (namely, instances in $\overline{K}$), and computationally binding on the NO instances (namely, instances in K). Moreover, the instance-dependent commitment scheme obtained is public coin.

Step 4. Finally, we use standard methods to combine the three instance-dependent commitment schemes that we have constructed into a single instance-dependent commitment scheme for Π. The first method gives a combined scheme for the intersection of two problems.

CLAIM 4.2.10

Suppose problems $\Gamma' = (\Gamma'_{\mathrm{Y}}, \Gamma'_{\mathrm{N}})$ and $\Gamma'' = (\Gamma''_{\mathrm{Y}}, \Gamma''_{\mathrm{N}})$ have instance-dependent commitment schemes Com'_x and Com''_x, respectively. Then problem $\Gamma \stackrel{\text{def}}{=} \Gamma' \cap \Gamma'' = (\Gamma'_{\mathrm{Y}} \cap \Gamma''_{\mathrm{Y}}, \Gamma'_{\mathrm{N}} \cup \Gamma''_{\mathrm{N}})$ has an instance-dependent commitment scheme Com_x with the following properties.

- Com_x is statistically [resp., computationally] hiding if both Com'_x and Com''_x are statistically [resp., computationally] hiding.
- Com_x is statistically [resp., computationally] binding if either of Com'_x or Com''_x is statistically [resp., computationally] binding.
- Com_x is public coin if both Com'_x and Com''_x are public coin.

- The round complexity of Com_x equals the larger of the round complexities of Com'_x and Com''_x.

Proof. In commitment scheme Com_x, the sender commits to b by committing to b in both schemes Com'_x and Com''_x, with the execution of both schemes done in parallel. The claimed properties of Com_x follow by inspection. □

The second method provides a combined scheme for the union of two problems.

CLAIM 4.2.11

Suppose problems $\Gamma' = (\Gamma'_{\mathrm{Y}}, \Gamma'_{\mathrm{N}})$ and $\Gamma'' = (\Gamma''_{\mathrm{Y}}, \Gamma''_{\mathrm{N}})$ have instance-dependent commitment schemes Com'_x and Com''_x, respectively. Then problem $\Gamma \stackrel{\mathrm{def}}{=} \Gamma' \cup \Gamma'' = (\Gamma'_{\mathrm{Y}} \cap \Gamma''_{\mathrm{Y}}, \Gamma'_{\mathrm{N}} \cup \Gamma''_{\mathrm{N}})$ has an instance-dependent commitment scheme Com_x with the following properties.

- Com_x is statistically [resp., computationally] hiding if either of Com'_x or Com''_x is statistically [resp., computationally] hiding.
- Com_x is statistically [resp., computationally] binding if both Com'_x and Com''_x are statistically [resp., computationally] binding.
- Com_x is public coin if both Com'_x and Com''_x are public coin.
- The round complexity of Com_x equals the larger of the round complexities of Com'_x and Com''_x.

Proof. In commitment scheme Com_x, the sender on input bit b, first secret shares b into two shares, b' and b'', with the property that $b' \oplus b'' = b$ and both b' and b'' are uniform in $\{0,1\}$. (This can be done by choosing a random $b' \leftarrow \{0,1\}$, and setting $b'' = b' \oplus b$.) The sender then commits to b by committing to bits b' and b'' in schemes Com'_x and Com''_x, respectively. The execution of schemes Com'_x and Com''_x is done in parallel.

The hiding property follows from the fact that bit b remains hidden as long as one of the bits b' or b'' remains hidden. Then binding property follows from the fact that $b = b' \oplus b''$, and hence b is bounded to a fixed value if both b' and b'' are bounded to fixed values. The public coin property and round complexity of Com_x follow by inspection. □

Having established the propositions and claims that we need, we now prove Lemma 4.2.9.

Proof of Lemma 4.2.9. Given that problem Π satisfies the Vadhan condition, let I be the set of OWF instances. We will partition I into the OWF YES instances $I_{\mathrm{Y}} = I \cap \Pi_{\mathrm{Y}}$ and the OWF NO instances $I_{\mathrm{N}} = I \cap \Pi_{\mathrm{N}}$. By Theorem 3.0.5, and Propositions 2.4.14 and 3.5.44, we have three instance-dependent commitment schemes, call them $\mathsf{Com}^{(1)}_x$, $\mathsf{Com}^{(2)}_x$, and $\mathsf{Com}^{(3)}_x$, for the problems $(\Pi_{\mathrm{Y}} \setminus I, \Pi_{\mathrm{N}} \setminus I) \in \mathrm{SZKP}$, $(I_{\mathrm{Y}}, \overline{I_{\mathrm{Y}}})$, and $(\overline{I_{\mathrm{N}}}, I_{\mathrm{N}})$, respectively. Moreover, all three schemes are public coin, and the first two are constant round.

If Π satisfies the Vadhan condition without OWF instances, then set $I = \emptyset$, and hence $\mathsf{Com}_x^{(1)}$ suffices to be our instance-dependent commitment scheme for Π. If Π satisfies the Vadhan condition without OWF NO instances, then $I_{\mathrm{N}} = I \cap \Pi_{\mathrm{N}} = \emptyset$. Consequently, we do not need scheme $\mathsf{Com}_x^{(3)}$, and can just combine schemes $\mathsf{Com}_x^{(1)}$ and $\mathsf{Com}_x^{(2)}$ in a manner prescribed by Claim 4.2.11 to get a constant-round instance-dependent commitment scheme for Π.

Analogously, if Π satisfies the Vadhan condition without OWF YES instances, then $I_{\mathrm{Y}} = I \cap \Pi_{\mathrm{Y}} = \emptyset$. Consequently, we do not need scheme $\mathsf{Com}_x^{(2)}$, and can just combine schemes $\mathsf{Com}_x^{(1)}$ and $\mathsf{Com}_x^{(3)}$ in a manner prescribed by Claim 4.2.10 to get an instance-dependent commitment scheme for Π. Finally, if Π satisfies the Vadhan condition, we first combine schemes $\mathsf{Com}_x^{(1)}$ and $\mathsf{Com}_x^{(2)}$ in a manner prescribed by Claim 4.2.11 to get an instance-dependent commitment scheme for $(\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}} \backslash I_{\mathrm{N}})$, and then combine this scheme with $\mathsf{Com}_x^{(3)}$ in a manner prescribed by Claim 4.2.10 to get an instance-dependent commitment scheme for Π.

The hiding, binding, and public coin properties of the instance-dependent commitment scheme for Π follow by inspection. □

4.2.3 From instance-dependent commitments to zero-knowledge protocols

The instance-dependent commitment scheme for a problem $\Pi \in \mathrm{NP}$ obtained in the previous subsection can be used to *unconditionally* construct a zero-knowledge protocol for Π. Recall that we did this in Section 2.5 where we substituted instance-dependent commitments for standard commitments in the Blum protocol [Blu]. In the following lemma, we expand upon Proposition 2.5.2 from Section 2.5.

LEMMA 4.2.12

(Expanded version of Proposition 2.5.2, which is based on [Blu].) If problem $\Pi \in \mathrm{NP}$ has an instance-dependent commitment scheme Com_x, then it has an *efficient-prover* protocol (P, V) with *perfect completeness* and the following additional properties.

- (P, V) is statistical [resp., computational] zero knowledge if Com_x is statistically [resp., computationally] hiding on the YES instances. Moreover, (P, V) has a black-box simulator.
- (P, V) is a proof [resp., argument] system if Com_x is statistically [resp., computationally] binding on the NO instances.
- (P, V) is public coin if Com_x is public coin.
- (P, V) has polynomially-small soundness error.
- The round complexity of (P, V) equals that of Com_x plus an additive constant.

REMARK 4.2.13

Protocol (P, V) in Lemma 4.2.12 above is obtained by repeating the Blum protocol [Blu], given by Protocol 2.5.1, a total of $O(\log n)$ times in *parallel*, which still maintains the zero knowledge property (cf., [BL, BLV]).

4.2.4 Putting it all together

We now show how our lemmas in Sections 4.2.1, 4.2.2, and 4.2.3 imply our main characterization theorems in Section 4.1.

Proof of Theorems 4.1.1, 4.1.2, 4.1.4, and 4.1.5. The implications for these four theorems are captured by the same lemmas, so we can conveniently state them together.

(1) ⇒ **(2)** is established by Lemma 4.2.1 for the SZKP and CZKP cases, and by Lemma 4.2.2 for SZKA and CZKA cases.

(2) ⇒ **(3)** is established by Lemma 4.2.9.

(3) ⇒ **(4)** is established by Lemma 4.2.12. This is the only step that requires the problem Π to be in NP. For problems $\Pi \in$ IP having zero-knowledge *proofs*, this direction was established in [Vad3, Sect. 4.2] based on techniques from [IY, BGG$^+$, IOS].[3]

(4) ⇒ **(1)** follows directly from definition. □

4.3 Symmetry between Zero Knowledge and Soundness

The characterization theorems in Section 4.1 yield the Symmetry Theorem (Theorem 1.2.2) presented in Section 1.2. Here we restate our Symmetry Theorem in a slightly more general form that captures promise problems. For a discussion on why we regard this theorem as establishing an unconditional symmetry between computational zero knowledge and computational soundness, refer back to page 8.

RESTATEMENT OF THEOREM 1.2.2

(Symmetry Theorem.)

1. CZKA versus co-CZKA: a problem $\Pi \in \text{NP} \cap \text{co-NP}$ has a computational zero-knowledge argument system if and only if its complement $\overline{\Pi}$ has a computational zero-knowledge argument system.

[3] Vadhan [Vad3] constructed public-coin honest-verifier zero-knowledge proofs for $\Pi \in$ IP from instance-dependent commitment schemes for Π with weaker properties, such as having an inefficient sender, and used a result of Goldreich, Sahai, and Vadhan [GSV1] to convert them into general, malicious-verifier zero-knowledge proofs. Since we now have instance-dependent commitments with standard properties, the construction presented in [Vad3, Sect. 4.2] would directly yield general, malicious-verifier zero-knowledge proofs (without using the [GSV1] conversion).

2. SZKA versus CZKP: a problem $\Pi \in$ NP has a statistical zero-knowledge argument system if and only if its complement $\overline{\Pi}$ has a computational zero-knowledge proof system.

Observe how the quality of the zero-knowledge condition for Π translates to the quality of the soundness condition for $\overline{\Pi}$ and vice-versa.

Proof of Theorem 1.2.2. Using the fact that SZKP = co-SZKP [Oka, GSV1], and the symmetric role played by the set of OWF instances I in the Vadhan condition (Definition 1.2.3), we can derive the following claim.

CLAIM 4.3.1

(Symmetry of the Vadhan condition.)

1. A problem Π satisfies the Vadhan condition if and only if its complement $\overline{\Pi}$ satisfies the Vadhan condition.
2. A problem Π satisfies the Vadhan condition without OWF YES instances with if and only if its complement $\overline{\Pi}$ satisfies the Vadhan condition without OWF NO instances.

For our first result, by Theorem 4.1.5, we know that $\Pi \in \mathrm{NP} \cap \text{co-NP}$ is in CZKA if and only if Π satisfies the Vadhan condition. From Item 1 of Claim 4.3.1 above, this is equivalent to $\overline{\Pi} \in \mathrm{NP} \cap \text{co-NP}$ satisfying the Vadhan condition. Applying Theorem 4.1.5 again makes this equivalent to $\overline{\Pi} \in$ CZKA.

For our second result, by Theorem 4.1.4, we know that $\Pi \in$ NP is in SZKA if and only if Π satisfies the Vadhan condition without OWF YES instances. From Item 2 of Claim 4.3.1 above, this is equivalent to $\overline{\Pi} \in \text{co-NP} \subseteq \mathrm{IP}$ satisfying the Vadhan condition without OWF NO instances. Applying Theorem 4.1.2 again makes this equivalent to $\overline{\Pi} \in$ CZKP. □

While our Symmetry Theorem establishes that the class CZKA is closed under complement for problems in $\mathrm{NP} \cap \text{co-NP}$, we still do not know if the classes CZKP or SZKA are closed under complement (even for problems in $\mathrm{NP} \cap \text{co-NP}$). Nevertheless, the Symmetry Theorem allows us to derive the following relationships.

COROLLARY 4.3.2

The following three statements are equivalent.

1. The class CZKP is closed under complement for problems in $\mathrm{NP} \cap \text{co-NP}$. In other words, $\mathrm{CZKP} \cap (\mathrm{NP} \cap \text{co-NP}) = \text{co-CZKP} \cap (\mathrm{NP} \cap \text{co-NP})$.
2. The class SZKA is closed under complement for problems in $\mathrm{NP} \cap \text{co-NP}$. In other words, $\mathrm{SZKA} \cap (\mathrm{NP} \cap \text{co-NP}) = \text{co-SZKA} \cap (\mathrm{NP} \cap \text{co-NP})$.

3. The class CZKP equals SZKA for problems in $\mathrm{NP} \cap \text{co-NP}$. In other words, $\mathrm{CZKP} \cap (\mathrm{NP} \cap \text{co-NP}) = \mathrm{SZKA} \cap (\mathrm{NP} \cap \text{co-NP})$.

Note that none of the above statements are known to be true unconditionally (though they all hold under the assumption that one-way functions exist), but if any one is true, then so are the others.

Proof of Corollary 4.3.2. We prove the equivalences of the three statements as follows.

(1) ⇒ (2) is established by:

$$\begin{aligned}
\mathrm{SZKA} \cap (\mathrm{NP} \cap \text{co-NP}) &= \text{co-CZKP} \cap (\mathrm{NP} \cap \text{co-NP}) && \text{(by the Symmetry Theorem)} \\
&= \mathrm{CZKP} \cap (\mathrm{NP} \cap \text{co-NP}) && \text{(by 1)} \\
&= \text{co-SZKA} \cap (\mathrm{NP} \cap \text{co-NP}) && \text{(by the Symmetry Theorem).}
\end{aligned}$$

(2) ⇒ (3) is established by:

$$\begin{aligned}
\mathrm{SZKA} \cap (\mathrm{NP} \cap \text{co-NP}) &= \text{co-SZKA} \cap (\mathrm{NP} \cap \text{co-NP}) && \text{(by 2)} \\
&= \mathrm{CZKP} \cap (\mathrm{NP} \cap \text{co-NP}) && \text{(by the Symmetry Theorem).}
\end{aligned}$$

(3) ⇒ (1) is established by:

$$\begin{aligned}
\mathrm{CZKP} \cap (\mathrm{NP} \cap \text{co-NP}) &= \mathrm{SZKA} \cap (\mathrm{NP} \cap \text{co-NP}) && \text{(by 3)} \\
&= \text{co-CZKP} \cap (\mathrm{NP} \cap \text{co-NP}) && \text{(by the Symmetry Theorem).}
\end{aligned}$$ □

We have now established all the results contained in this dissertation. In the next chapter, we explore a direction for future research.

5

FUTURE RESEARCH

In this dissertation, we constructed instance-dependent commitment schemes for a problem Π based on *any*—even an honest verifier—zero-knowledge protocol for Π. We then established our various unconditional results by substituting instance-dependent commitments for standard commitments in existing zero-knowledge protocols. In this final chapter, we explore a future research direction motivated by the following question: can we obtain *constant-round* statistical [resp., computational] zero-knowledge arguments for every problem in $\mathrm{SZKA} \cap \mathrm{NP}$ [resp., $\mathrm{CZKA} \cap \mathrm{NP}$]?

For comparison, our characterization theorems for zero-knowledge *proofs*, as stated by Theorems 4.1.1 and 4.1.2, yield constant-round statistical [resp., computational] zero-knowledge proofs (with perfect completeness and a polynomially small soundness error) for every problem in $\mathrm{SZKP} \cap \mathrm{NP}$ [resp., $\mathrm{CZKP} \cap \mathrm{NP}$]. The bottleneck in extending this to argument systems turns out to be that the known construction of statistically-hiding commitments based on any one-way function, as presented in this dissertation, has polynomial number of rounds (cf., [HR2]). Moreover, any *fully-black-box* construction of statistically-hiding commitments even from any one-way permutation requires $\Omega(n/\log n)$ rounds [HHRS], and indeed ours is a fully-black-box construction.[1] Recall that we needed to base our construction of statistically-hiding commitments on one-way functions because in Theorems 4.1.4 and 4.1.5, we characterized every problem having zero-knowledge arguments in terms of instances with a statistical zero-knowledge proofs plus a set of instances I_{N} from which we can construct a *one-way function* (i.e., the OWF NO instances).

Nevertheless, constant-round statistically-hiding commitment schemes can based on any

[1] See [RTV, Def. 2.3] and [HHRS, Def. 2.6] for the definition of a *fully-black-box construction.*

collision-resistant hash family [NY, DPP].[2] Thus, if we can characterize every problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}) \in \mathrm{SZKA}$ in terms of instances with a statistical zero-knowledge proof system plus a set of instances $I_{\mathrm{N}} \subseteq \Pi_{\mathrm{N}}$ from which we can construct a *collision-resistant hash family*, then we should expect to resolve the above question in the affirmative in the SZKA case: proving that every problem in $\mathrm{SZKA} \cap \mathrm{NP}$ has a *constant-round* statistical zero-knowledge argument system.

And although not immediately obvious, the CZKA case does follow from the SZKA case if our hypothesis turns out to be true. This is because, based on our main characterization theorems, every problem $\Pi \in \mathrm{CZKA} \cap \mathrm{NP}$ can be characterized in terms of instances with a statistical zero-knowledge *argument* system plus a set of instances $I_{\mathrm{Y}} \subseteq \Pi_{\mathrm{Y}}$ from which we can construct a one-way function (i.e., the OWF YES instances). And we can construct *constant-round* computationally-hiding and statistically-binding commitments from this set of OWF YES instances (see Proposition 2.4.14).

To formalize our hypothesis as an open problem, we first give a proposed definition of a hash family that is collision resistant on a set I, following the spirit of an *instance-dependent one-way function* in Definition 2.4.6.

DEFINITION 5.0.3

A polynomial-time computable family $\mathcal{H} = \bigcup_x \mathcal{H}_x = \{h\colon \{0,1\}^{n(|x|)} \to \{0,1\}^{m(|x|)}\}$, where $n(\cdot) > m(\cdot)$ are polynomials, is an ***instance-dependent collision-resistant hash family on*** I if there exists a negligible function ε such that for every nonuniform PPT A, the following holds for every $x \in I$:

$$\Pr_{h \leftarrow \mathcal{H}_x}[A(x,h) = (\alpha, \alpha') \text{ such that } h(\alpha) = h(\alpha')] \leq \varepsilon(|x|) \enspace .$$

OPEN PROBLEM 5.0.4

For a problem $\Pi = (\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}}) \in \mathrm{SZKA} \cap \mathrm{NP}$, does there exists a set $I_{\mathrm{N}} \subseteq \Pi_{\mathrm{N}}$ such that:

- the promise problem $(\Pi_{\mathrm{Y}}, \Pi_{\mathrm{N}} \setminus I_{\mathrm{N}})$ is in SZKP, and
- there exists an instance-dependent collision-resistant hash family on I_{N} ?

As a first step, we could even ask whether the above two conditions hold for problems $\Pi \in \mathrm{NP}$ having *constant-round* statistical zero-knowledge arguments (instead of considering all problems in $\mathrm{SZKA} \cap \mathrm{NP}$).

More ambitiously, it would be interesting to see if the techniques presented in this dissertation can be extended to conduct an unconditional study on other cryptographic constructs, like *multiparty cryptographic protocols* (cf., [Yao, GMW1]).

[2] A ***collision-resistant hash family*** is a family of hash functions $\mathcal{H} = \{h\colon \{0,1\}^n \to \{0,1\}^m\}$ where $n > m$, and given a random hash $h \leftarrow \mathcal{H}$, it is computationally infeasible to find a pair y and y' such that $h(y) = h(y')$. See Definition 5.0.3 for an instance-dependent variant.

A

DEFERRED PROOFS

We present proofs that have been deferred from the main text. Appendices A.1, A.2, and A.3 contain proofs deferred from Sections 3.2.2, 3.5.2, and 4.2.1, respectively.

A.1 Interactive Hashing with Multiple Outputs

This section is devoted to prove Theorem 3.2.4 from Section 3.2.2, restated below.

RESTATEMENT OF THEOREM 3.2.4

There exists an interactive hashing with multiple outputs protocol, namely Protocol 3.2.3.

The correctness of Protocol 3.2.3 is easy to see. Hence, we divide the proof of this theorem into lemmas establishing the hiding and binding properties of Protocol 3.2.3. The proofs presented are very similar in nature to those in [NOVY], with additional analysis needed to handle interactive hashing for multiple outputs.

LEMMA A.1.1

Protocol 3.2.3 satisfies the hiding property of Definition 3.2.1. In other words, letting interactive hashing $(S_{\rm IH}, R_{\rm IH})$ be as in Protocol 3.2.3, we have for all R^*, (V, Z) is distributed identically to (V, U_k), where $V = \text{view}_{R^*}(S_{\rm IH}(U_q), R^*)$ is the view of receiver R^*, and $Z = \text{output}_{S_{\rm IH}}(S_{\rm IH}(U_q), R^*)$ is the private output of $S_{\rm IH}$.

Proof. The view of any R^* will be the hash functions $h_0, h_1, \cdots, h_{q-k-1}$ together with $S_{\rm IH}$'s responses $c_0, c_1, \ldots, c_{q-k-1}$. Given queries $h_0, h_1, \cdots, h_{q-k-1}$ from R^*, we show that there are 2^{q-k} possible y's that would make $S_{\rm IH}(y)$ respond to $c_0, c_1, \ldots, c_{q-k-1}$.

Consider the matrix $H = (h_0, h_1, \cdots, h_{q-k-1})$ whose rows are the h_i's, vector $c = (c_0, c_1, \ldots, c_{q-k-1})$, and the equation $Hy = c$. Since h_i is of the form $0^i1\{0,1\}^{q-i-1}$, the first $q-k$ columns of the matrix are linearly independent. Hence, any setting of the last k bits of y, will fully determine the first $q-k$ bits of it. Since the output of S_{IH}, denoted as z, is the last k bits of its private input y, any $z \in \{0,1\}^k$ is equally as likely given the view of R^*. □

LEMMA A.1.2

Protocol 3.2.3 satisfies the binding property of Definition 3.2.1. That is, letting interactive hashing $(S_{\mathrm{IH}}, R_{\mathrm{IH}})$ be as in Protocol 3.2.3, there exists a oracle PPT algorithm A such that:

> For every S^* and any relation W, denoting the common output as $C = (S^*, R_{\mathrm{IH}})(1^q, 1^k)$, and private outputs of S^* as $((x_0, z_0), (x_1, z_1)) = \mathrm{output}_{S^*}(S^*, R_{\mathrm{IH}})$, if it is the case that
>
> $$\Pr[x_0 \in W_{C(z_0)} \wedge x_1 \in W_{C(z_1)} \wedge z_0 \neq z_1] > \varepsilon \ ,$$
>
> where the above probability is over the coin tosses of R_{IH} and S^*, then it is also the case that
>
> $$\Pr_{y \leftarrow \{0,1\}^q}[A^{S^*}(y, 1^q, 1^k, \varepsilon) \in W_y] = \Omega(\varepsilon^3 q^{-6} 2^{-k}) \ .$$

REMARK A.1.3

Independent of our work, Haitner and Reingold [HR1] gave an improved bound that brings the success probability of A^{S^*} to $\Omega(\varepsilon^2 q^{-8} 2^{-k})$. Their witness-finding algorithm A^{S^*} is also different from ours and is more efficient in terms of its running time. We will, however, not need these improved bounds for our applications.

We prove Lemma A.1.2 by providing an algorithm A that finds a valid witness (according to relation W) for a random string $y \leftarrow \{0,1\}^q$ with nonnegligible probability. Before describing algorithm A, we provide the following definitions.

Definitions. In the enumerated definitions below, h_i is of the form $0^i1\{0,1\}^{q-i-1}$, and $h_i(y) = \langle h_i, y \rangle$. Without loss of generality, we can assume that S^* is deterministic because every probabilistic S^* can be converted to a (nonuniform) deterministic one with the same success probability and running time by fixing its random coins to maximize its success probability.

1. For $0 \leq i < q$, let $\mathcal{H}_i$ denote the family of hash functions of the form $0^i1\{0,1\}^{q-i-1}$, i.e., $\mathcal{H}_i = \{0^i1w : w \in \{0,1\}^{q-i-1}\}$.

2. A ***node*** N at level i is defined by a series of hash functions $(h_0, h_1, \ldots, h_{i-1})$, where

each $h_j \in \mathcal{H}_j$. (Since S^* is deterministic, this determines $c_0, \ldots, c_{i-1}$ where $c_j = S^*(h_0, \ldots, h_j)$.) Let L_i denote the set of nodes at level i.

3. The set of compatible hash functions at node $N \in L_i$ is denoted as

$$\mathrm{Comp}(N, y) = \{h_i \in \mathcal{H}_i : S^*(N, h_i) = h_i(y)\} \ ,$$

where $S^*(N, h_i)$, with $N = (h_0, \ldots, h_{i-1})$, denotes $S^*(h_0, \ldots, h_i)$.

4. A string y is γ-***balanced*** at $N \in L_i$ if

$$\frac{1-\gamma}{2} \le \frac{\mathrm{Comp}(N, y)}{|\mathcal{H}_i|} \le \frac{1+\gamma}{2} \ .$$

A string y is γ-***fully-balanced*** at $N \in L_i$ if it is γ-balanced at all its parental nodes. That is, letting $N = (h_0, \ldots, h_{i-1})$, y is required to be γ-balanced at all $N_0 = (h_0), N_1 = (h_0, h_1), \ldots, N = N_{i-1} = (h_0, \ldots, h_{i-1})$.

5. A string y is said to be ***compatible*** with a node $N = (h_0, \ldots, h_{i-1})$ if $h_j(y) = S^*(h_0, \ldots, h_j)$ for all $0 \le j < i$. Let $U(N)$ denote the set of compatible y's with node N. Note that for every $N \in L_i$, we have $|U(N)| = 2^{q-i}$.

6. Let $B(N)$ and $F(N)$ denote the set of γ-balanced strings and γ-fully-balanced strings at node N respectively. Moreover, let $G(N) = U(N) \setminus F(N)$ be the set of strings that are not fully-balanced. Note that for every node N, we have $F(N) \subseteq B(N) \subseteq U(N)$.

7. At every node $N \in L_{q-k}$, we can assume without loss of generality that $S^*(N)$ outputs a pair of strings (x_0, z_0) and (x_1, z_1), but it is not necessarily the case that any of $x_b \in W_{C(z_b)}$.

The description of our witness-finding algorithm is presented next.

ALGORITHM A.1.4

Algorithm A^{S^*}: on input $y \in \{0,1\}^q, 1^q, 1^k$ and ε, do the following.

1. Set parameters $\gamma = 1/q$, $\beta = \log(1/\varepsilon) + 2\log(q) + 4\log(1/\gamma) + 4$, and $\alpha = q - \beta - k$.

2. Repeat the following for $i = 0, 1, \ldots, \alpha - 1$:

   When A is at node $N \in L_i$, explore along a random $h_i \leftarrow \mathrm{Comp}(N, y)$ to get to a new node $N' = (N, h_i) \in L_{i+1}$. (This can be done efficiently by choosing a random $h_i \leftarrow \mathcal{H}_i$ and querying S^* to make sure that $h_i \in \mathrm{Comp}(N, y)$, and repeat up to $8q$ times if not. If after $8q$ repetitive tries and fail to encounter any $h_i \in \mathrm{Comp}(N, y)$, then output `fail`.)

3. At node $N \in L_\alpha$, choose random $h_\alpha \leftarrow \mathcal{H}_\alpha, \ldots, h_{\alpha+\beta-1} \leftarrow \mathcal{H}_{\alpha+\beta-1}$, to arrive at node $\widetilde{N} = (N, h_\alpha, h_{\alpha+1}, \ldots, h_{\alpha+\beta-1}) \in L_{\alpha+\beta}$. (Note that $q - k = \alpha + \beta$, and hence $\widetilde{N} \in L_{\alpha+\beta} = L_{q-k}$.)

4. Query $S^*(\widetilde{N})$ to get (x_0, z_0) and (x_1, z_1). If either of $C(z_b) = y$, then output x_b. Else, output `fail`.

It is clear that algorithm A^{S^*} stated in Algorithm A.1.4 runs in polynomial time (with oracle queries to S^*). The remainder of the proof is broken down into the following claims.

CLAIM A.1.5

For every node $N \in L_i$, the set of unbalanced strings, $U(N) \setminus B(N) \leq 2/\gamma^2$.

Proof. Let $X \subseteq U(N)$ be a set of size 2^d, where $d = 2\log(1/\gamma)$. We interpret X as a random variable that has equal weights on each of its 2^d elements. Let $\mathcal{H}_i$ be the family of hash functions after node N of the form $0^i 1\{0,1\}^{q-i-1}$. Observe that for every $x \neq x'$, $\Pr_{h_i \leftarrow \mathcal{H}_i}[h_i(x) = h_i(x')] \leq 1/2$. Also, note that h_i requires exactly $q - i - 1$ bits to describe.

Recall the definition of collision probability, denoted as CP, from Section 3.5.2. Computing the collision probabilities (using the notation H_i to denote a random hash function from $\mathcal{H}_i$), we get

$$\begin{aligned}
\mathrm{CP}((H_i, H_i(X))) &\leq \mathrm{CP}(H_i)(\mathrm{CP}(X) + \Pr[H_i(X) = H_i(X') : X \neq X']) \\
&\leq \mathrm{CP}(H_i) \cdot (1/2^d + 1/2) \\
&= 2^{-(q-i-1)}(1/2^d + 1/2) \qquad \text{whereas,}
\end{aligned}$$

$$\begin{aligned}
\mathrm{CP}((H_i, U_1)) &= \mathrm{CP}(H_i) \cdot 1/2 \\
&= 2^{-(q-i-1)} \cdot (1/2)\ .
\end{aligned}$$

Thus,

$$\begin{aligned}
\Delta((H_i, H_i(X)), (H_i, U_1)) &= 1/2\,|(H_i, H_i(X)) - (H_i, U_1)|_1 \\
&\leq 1/2 \cdot \sqrt{2^{q-i-1}} \cdot \sqrt{\mathrm{CP}((H_i, H_i(X))) - \mathrm{CP}((H_i, U_1))} \\
&\leq 1/2\sqrt{1/2^d} \\
&= 2^{-d/2-1} \\
&\leq \gamma/2\ ,
\end{aligned}$$

with the last inequality following from $d = 2\log(1/\gamma)$.

Having establish the above bound, assume for sake of contradiction that $U(N) \setminus B(N) > 2^{d+1} = 2/\gamma^2$. Then we will have a set $T \subseteq U(N) \setminus B(N)$ of size greater then 2^d with elements that are unbalanced in one direction (i.e. all $> 1/2 + \gamma$, or all $< 1/2 - \gamma$). But this contradicts the requirement that $\Delta((H_i, H_i(T)), (H_i, U_1)) \leq \gamma/2$ (since $|T| > 2^d$). □

CLAIM A.1.6

For every node $N \in L_i$, the set of strings that are not fully balanced, $G(N) = U(N) \setminus F(N) \leq 2i/\gamma^2$. In particular, for $\gamma = 1/q$, $|F(N)| \geq |U(N)|/2$ for $i \leq q - 4\log q$.

Proof. Follows from Claim A.1.5 by taking a union bound over all unbalanced elements at levels i and smaller. □

CLAIM A.1.7

For every node $N \in L_\alpha$, the fraction of children nodes $N_{\alpha+\beta}$ with greater than one element from $G(N)$ is at most $\varepsilon/4$.

Proof. Consider any fixed node $N \in L_\alpha$. The number of non-fully-balanced (aka bad) elements in that node is $G(N)$. Hence, the number of pairs of these bad elements is at most $|G(N)|^2$. Since for each $x \neq y \in U(N)$, $\Pr[h_i(x) = h_i(y)] \leq 1/2$ for all $\alpha \leq i < \alpha + \beta$, the fraction of children nodes $N' \in L_{\alpha+\beta}$ with greater than one element from $G(N)$ is at most $|G(N)|^2 / 2^\beta$.

Since $\beta = \log(1/\varepsilon) + 2\log(q) + 4\log(1/\gamma) + 4$, we can bound $|G(N)|^2 / 2^\beta$ as follows:

$$|G(N)|^2 \cdot 2^{-\beta} \leq (2\alpha\gamma^{-2})^2 2^{-\beta} \leq 4q^2\gamma^{-4}2^{-\beta} < \varepsilon/4 \ . \qquad \square$$

A node $N \in L_{\alpha+\beta} = L_{q-k}$ is ***witness revealing*** if both of $S^*(N)$'s outputs, namely (x_0, z_0) and (x_1, z_1), satisfy $C(z_b) \in U(N)$ and $x_b \in W_{C(z_b)}$, for $b \in \{0,1\}$. A node $N \in L_\alpha$ is said to be ***good*** if greater than $\varepsilon/2$ of its children at level $q-k$ are witness revealing.

CLAIM A.1.8

The fraction of good nodes at level α is at least $\varepsilon/2$.

Proof. By the assumption that

$$\Pr\left[x_0 \in W_{C(z_0)} \wedge x_1 \in W_{C(z_1)} \middle| \begin{array}{c} (S^*, R)(1^q, 1^k) = C \ \text{, and} \\ \text{output}_{S^*}(S^*, R) = ((x_0, z_0), (x_1, z_1)) \end{array} \right] > \varepsilon \ ,$$

we know that at least ε fraction of all the nodes at level $q-k$ are nonbinding. And, by a Markov bound, we have that $\varepsilon/2$ fraction of nodes at level α are good. □

CLAIM A.1.9

For any fixed $N \in L_\alpha$ and $y' \in F(N)$, we have

$$\frac{1}{2^{q-\alpha}} \cdot \frac{1}{(1+\gamma)^\alpha} \cdot \frac{1}{|L_\alpha|} \leq \Pr[A \text{ reaches } N \wedge y = y'] \leq \frac{1}{2^{q-\alpha}} \cdot \frac{1}{(1-\gamma)^\alpha} \cdot \frac{1}{|L_\alpha|} \ ,$$

where the probability is taken over $y \in \{0,1\}^q$ and the random coins of A.

Proof. Let $N = (h_0, h_2, \ldots, h_{\alpha-1})$, and for $1 \le j \le \alpha$, define $N_j = (h_0, \ldots, h_{j-1})$. To get the upper bound,

$$\begin{aligned}
\Pr[A \text{ reaches } N \wedge y = y'] &= \Pr[y = y'] \cdot \Pr[A \text{ reaches } N] \\
&= 2^{-q} \prod_{j=0}^{\alpha-1} \frac{1}{\mathrm{Comp}(N_j, y)} \\
&\le 2^{-q} \prod_{j=0}^{\alpha-1} \frac{2}{1-\gamma} \cdot \frac{1}{|\mathcal{H}_j|} \\
&= \frac{1}{2^{q-\alpha}} \cdot \frac{1}{(1-\gamma)^{\alpha}} \cdot \frac{1}{|L_\alpha|} \quad .
\end{aligned}$$

To get the lower bound, we use very similar techniques.

$$\begin{aligned}
\Pr[A \text{ reaches } N \wedge y = y'] &= 2^{-q} \prod_{j=0}^{\alpha-1} \frac{1}{\mathrm{Comp}(N_j, y)} \\
&\ge 2^{-q} \prod_{j=0}^{\alpha-1} \frac{2}{1+\gamma} \cdot \frac{1}{|\mathcal{H}_j|} \\
&= \frac{1}{2^{q-\alpha}} \cdot \frac{1}{(1+\gamma)^{\alpha}} \cdot \frac{1}{|L_\alpha|} \quad .
\end{aligned}$$

□

CLAIM A.1.10

$$\Pr[\text{The node } N \text{ reached by } A \text{ is good } \wedge y \in F(N)] \ge \frac{\varepsilon}{4(1+\gamma)^{\alpha}} \quad ,$$

where the probability is taken over $y \in \{0,1\}^q$ and the random coins of A.

Proof. Let $N \in L_\alpha$ be any good node at level α. Then,

$$\begin{aligned}
\Pr[A \text{ reaches } N \wedge y \in F(N)] &= \sum_{y' \in F(N)} \Pr[A \text{ reaches } N \wedge y = y'] \\
&\ge \sum_{y' \in F(N)} \frac{1}{|L_\alpha|} \cdot \frac{1}{2^{q-\alpha}} \cdot \frac{1}{(1+\gamma)^{\alpha}} \\
&= \frac{|F(N)|}{2^{q-\alpha}} \cdot \frac{1}{|L_\alpha|} \cdot \frac{1}{(1+\gamma)^{\alpha}} \\
&= \frac{|F(N)|}{|U(N)|} \cdot \frac{1}{|L_\alpha|} \cdot \frac{1}{(1+\gamma)^{\alpha}} \\
&\ge \frac{1}{2} \cdot \frac{1}{|L_\alpha|} \cdot \frac{1}{(1+\gamma)^{\alpha}} \quad ,
\end{aligned}$$

with the last inequality following from the fact that $|F(N)| / |U(N)| \geq 1/2$, noting $\alpha \leq q - 3\log q$ (refer to Claim A.1.6).

There are $|L_\alpha|$ nodes at level α, and at least $\varepsilon/2$ fraction of them are good. Hence, we multiply the above probability by $(\varepsilon/2)\,|L_\alpha|$ to get our stated result. □

CLAIM A.1.11

In any good node $N \in L_\alpha$, the fraction of nonbinding children of N at level $\alpha+\beta$ that has one or less image in $G(N)$ is at least $\varepsilon/4$.

Proof. The fraction of nonbinding children is greater than $\varepsilon/2$, and by Claim A.1.7, the fraction of children nodes of N with greater than one element from $G(N)$ is at most $\varepsilon/4$. □

CLAIM A.1.12

For any fixed $N \in L_\alpha$ and $y' \in F(N)$, we have

$$\Pr[y = y' | A \text{ reaches } N \wedge y \in F(N)] \geq \frac{1}{|F(N)|}\left(\frac{1-\gamma}{1+\gamma}\right)^\alpha \quad ,$$

where the probability is taken over $y \in \{0,1\}^q$ and the random coins of A.

Proof. For any fixed $N \in L_\alpha$ and $y' \in F(N)$,

$$\Pr[y = y' | A \text{ reaches } N \wedge y \in F(N)] = \frac{\Pr[A \text{ reaches } N \wedge y = y']}{\Pr[A \text{ reaches } N \wedge y \in F(N)]} \quad .$$

For the numerator, by Claim A.1.9,

$$\Pr[A \text{ reaches } N \wedge y = y'] \geq \frac{1}{|L_\alpha|} \cdot \frac{1}{2^{q-\alpha}} \cdot \frac{1}{(1+\gamma)^\alpha} \quad .$$

For the denominator, also using Claim A.1.9,

$$\begin{aligned}\Pr[A \text{ reaches } N \wedge y \in F(N)] &= \sum_{y' \in F(N)} \Pr[A \text{ reaches } N \wedge y = y'] \\ &\leq \sum_{y' \in F(N)} \frac{1}{|L_\alpha|} \cdot \frac{1}{2^{q-\alpha}} \cdot \frac{1}{(1-\gamma)^\alpha} \\ &= |F(N)| \cdot \frac{1}{|L_\alpha|} \cdot \frac{1}{2^{q-\alpha}} \cdot \frac{1}{(1-\gamma)^\alpha} \quad .\end{aligned}$$

Combining the two, we have our result. □

We are now ready to prove Lemma A.1.2.

Proof of Lemma A.1.2. Observe how algorithm A in Algorithm A.1.4 operates. On input y, it follows a random compatible (with y) hash functions h_i out of node $N \in L_i$, for $1 \leq i < \alpha$, and then takes random h_i's (not necessarily compatible with y) when $\alpha \leq i < \alpha + \beta$. (For now, we can ignore failure to obtain compatible hash functions.) Algorithm A will find a valid witness for y if the all following conditions hold.

1. Algorithm A reaches a good node $N \in L_\alpha$ such that $y \in F(N)$. This happens with probability at least $\varepsilon/(4(1+\gamma)^\alpha)$ (Claim A.1.10).

2. Algorithm A reaches a witness revealing child with at most one element in $G(N)$. Given that Item 1 occurs, this happens with probability at least $\varepsilon/4$ (Claim A.1.11).

   When this is the case, S^* outputs (x_0, z_0) and (x_1, z_1), such that at least one (x_b, z_b) will have $x_b \in W_{C(z_b)}$ and $C(z_b) \in U(N) \setminus G(N) = F(N)$. Let $y' = C(z_b)$.

3. The string y equals $y' = C(z_b)$. The conditional probability of this happening, by Claim A.1.12, is:

   $$\Pr[y = y' | A \text{ reaches } N \wedge y' \in F(N)] \geq \frac{1}{|F(N)|}\left(\frac{1-\gamma}{1+\gamma}\right)^\alpha .$$

   When this happens, A will output $x_b \in W_y$, a valid witness for y.

Combining all the probabilities, we have

$$\begin{aligned}\Pr_{y \leftarrow \{0,1\}^q}[A(y) \in W_y] &\geq \frac{\varepsilon}{4(1+\gamma)^\alpha} \cdot \frac{\varepsilon}{4} \cdot \frac{1}{|F(N)|}\left(\frac{1-\gamma}{1+\gamma}\right)^\alpha \\ &\geq \frac{1}{2^{\beta+k}} \cdot \frac{\varepsilon^2}{32} \cdot \left(\frac{1-\gamma}{(1+\gamma)^2}\right)^q .\end{aligned}$$

With settings of parameters $\gamma = 1/q$ and $\beta = \log(1/\varepsilon) + 2\log(q) + 4\log(1/\gamma)) + 4$, we have the probability of finding a witness to be greater than $c \cdot (\varepsilon^3 q^{-6} 2^{-k})$, for some constant $c > 0$.

Finally, we need to account for the case when we fail to find compatible hash functions h_i out of node $N \in L_i$, for $1 \leq i < \alpha$. Nevertheless, because our analysis has only focused on fully balanced y, and we repeat $8q$ times to find a compatible hash, the probability of failure, by a Chernoff bound, is exponentially small. Therefore, the overall success probability is greater than $c \cdot (\varepsilon^3 q^{-6} 2^{-k}) - \exp(q) = \Omega(\varepsilon^3 q^{-6} 2^{-k})$. □

A.2 Collision Probability Lemmas

We prove the lemmas presented in Section 3.5.2.

RESTATEMENT OF LEMMA 3.5.4

For independent pairs of random variables $(X_1, Y_1), \ldots, (X_m, Y_m)$,

$$\mathrm{CP}^{1/2}((X_1, \ldots, X_m)|(Y_1, \ldots, Y_m)) = \prod_{i=1}^{m} \mathrm{CP}^{1/2}(X_i|Y_i) \ .$$

Note that X_i and Y_i can be correlated, it is only required that the pair (X_i, Y_i) be independent from the other tuples.

Proof. Since the X_i's are independent, for all $y_1, \ldots, y_m$, we have

$$\mathrm{CP}((X_1, \ldots, X_m)|_{Y_1=y_1,\ldots,Y_m=y_m}) = \prod_{i=1}^{m} \mathrm{CP}(X_i|_{Y_i=y_i}) \ . \tag{A.1}$$

This gives us

$$\begin{aligned}
&\mathrm{CP}^{1/2}((X_1, \ldots, X_m)|(Y_1, \ldots, Y_m)) \\
&= \mathop{\mathrm{E}}_{(Y_1,\ldots,Y_m)} \left[\mathrm{CP}^{1/2}((X_1, \ldots, X_m)|_{Y_1,\ldots,Y_m})\right] \\
&= \mathop{\mathrm{E}}_{(Y_1,\ldots,Y_m)} \left[\prod_{i=1}^{m} \mathrm{CP}^{1/2}(X_i|_{Y_i})\right] && \text{(by A.1)} \\
&= \prod_{i=1}^{m} \mathop{\mathrm{E}}_{Y_i} \left[\mathrm{CP}^{1/2}(X_i|_{Y_i})\right] && \text{(by independence of } Y_i\text{'s)} \\
&= \prod_{i=1}^{m} \mathrm{CP}^{1/2}(X_i|Y_i) \ . && \square
\end{aligned}$$

RESTATEMENT OF LEMMA 3.5.5

Suppose random variables $(X_1, Y_1), \ldots, (X_m, Y_m)$ satisfy the following conditions for some values of $\alpha_1, \ldots, \alpha_m \in \mathbb{R}^+$ and all $i = 1, 2, \ldots, m$:

1. For every $(y_1, \ldots, y_{i-1}) \in \mathrm{Supp}(Y_1, Y_2, \ldots, Y_{i-1})$,

$$\mathrm{CP}^{1/2}(X_i|_{Y_1=y_1,\ldots,Y_{i-1}=y_{i-1}} \mid Y_i|_{Y_1=y_1,\ldots,Y_{i-1}=y_{i-1}}) \leq \alpha_i \ .$$

2. For every $(y_1, \ldots, y_i) \in \mathrm{Supp}(Y_1, Y_2, \ldots, Y_i)$, the $i+1$ random variables $X_1, X_2, \ldots, X_i$, and Y_{i+1} are independent, even if we condition on $Y_1 = y_1, \ldots, Y_i = y_i$.

Then,

$$\mathrm{CP}^{1/2}((X_1,\ldots,X_m)|(Y_1,\ldots,Y_m)) \le \prod_{i=1}^{m} \alpha_i \ .$$

Proof. By induction, it suffices to prove

$$\mathrm{CP}^{1/2}((X_1,\ldots,X_m)|(Y_1,\ldots,Y_m)) \le \alpha_m \cdot \mathrm{CP}^{1/2}\left((X_1,\ldots,X_{m-1})|(Y_1,\ldots,Y_{m-1})\right) \ , \tag{A.2}$$

and then by iteratively expanding $\mathrm{CP}^{1/2}\left((X_1,\ldots,X_{m-1})|(Y_1,\ldots,Y_{m-1})\right)$ in terms of α_j's, we get our result. To simplify notation, we write $X'_m = X_m|_{Y_1=y_1,\ldots,Y_{m-1}=y_{m-1}}$ and $Y'_m = Y_m|_{Y_1=y_1,\ldots,Y_{m-1}=y_{m-1}}$ when $y_1,\ldots,y_{m-1}$ are clear from context. We prove (A.2) as follows:

$$
\begin{align}
&\mathrm{CP}^{1/2}((X_1,\ldots,X_m)|(Y_1,\ldots,Y_m)) \tag{A.3}\\
&= \mathop{\mathrm{E}}_{(Y_1,\ldots,Y_m)}\left[\mathrm{CP}^{1/2}((X_1,\ldots,X_m)|_{Y_1,\ldots,Y_m})\right] \tag{A.4}\\
&= \mathop{\mathrm{E}}_{(Y_1,\ldots,Y_{m-1})}\left[\mathop{\mathrm{E}}_{Y'_m}\left[\mathrm{CP}^{1/2}((X_1,\ldots,X_m)|_{Y_1,\ldots,Y'_m})\right]\right] \tag{A.5}\\
&= \mathop{\mathrm{E}}_{(Y_1,\ldots,Y_{m-1})}\left[\mathop{\mathrm{E}}_{Y'_m}\left[\mathrm{CP}^{1/2}((X_1,\ldots,X_{m-1})|_{Y_1,\ldots,Y'_m}) \cdot \mathrm{CP}^{1/2}(X_m|_{Y_1,\ldots,Y'_m})\right]\right] \tag{A.6}\\
&= \mathop{\mathrm{E}}_{(Y_1,\ldots,Y_{m-1})}\left[\mathrm{CP}^{1/2}((X_1,\ldots,X_{m-1})|_{Y_1,\ldots,Y_{m-1}}) \cdot \mathop{\mathrm{E}}_{Y'_m}\left[\mathrm{CP}^{1/2}(X_m|_{Y_1,\ldots,Y'_m})\right]\right] \tag{A.7}\\
&= \mathop{\mathrm{E}}_{(Y_1,\ldots,Y_{m-1})}\left[\mathrm{CP}^{1/2}((X_1,\ldots,X_{m-1})|_{Y_1,\ldots,Y_{m-1}}) \cdot \mathrm{CP}^{1/2}(X'_m|Y'_m)\right] \tag{A.8}\\
&\le \alpha_m \cdot \mathop{\mathrm{E}}_{(Y_1,\ldots,Y_{m-1}))}\left[\mathrm{CP}^{1/2}((X_1,\ldots,X_{m-1})|_{Y_1,\ldots,Y_{m-1}})\right] \tag{A.9}\\
&\le \alpha_m \cdot \mathrm{CP}^{1/2}\left((X_1,\ldots,X_{m-1})|(Y_1,\ldots,Y_{m-1}))\right) \ . \tag{A.10}
\end{align}
$$

Equation (A.6) follows because $X_1,\ldots,X_m$ conditioned on $Y_1 = y_1,\ldots,Y_m = y_m$ are independent. Equation (A.7) follows because $X_1,\ldots,X_{m-1}$, and Y_m conditioned on $Y_1 = y_1,\ldots,Y_{m-1} = y_{m-1}$ are independent. Finally, (A.9) follows from the assumption that for all $(y_1,\ldots,y_{i-1}) \in \mathrm{Supp}(Y_1,Y_2,\ldots,Y_{m-1})$,

$$\mathrm{CP}^{1/2}(X'_m|Y'_m) = \mathrm{CP}^{1/2}(X_m|_{Y_1=y_1,\ldots,Y_{m-1}=y_{m-1}} \mid Y_m|_{Y_1=y_1,\ldots,Y_{m-1}=y_{m-1}}) \le \alpha_m \ . \qquad \square$$

RESTATEMENT OF LEMMA 3.5.6

(Randomness Extraction Lemma.) Let (X,Y) be any (possibly correlated) pair of random variables, and let random variable H denote a random hash function from a family of pairwise-independent hash functions $\mathcal{H}$ with range $\{0,1\}^{\alpha}$. Suppose the hash functions from $\mathcal{H}$ are represented by $(q-\alpha)$-bit strings and $\mathrm{CP}^{1/2}(X|Y) \le \sqrt{2^{-(\alpha+3)}}$. If H is independent from (X,Y), then

$$\mathrm{CP}^{1/2}((H,H(X))|Y) \le \sqrt{2^{-(q-1)}} \ .$$

Proof. We bound the value of $\mathrm{CP}^{1/2}((H, H(X))|Y)$ as follows:

$$\begin{aligned}
&\mathrm{CP}^{1/2}(H, H(X)|Y) \\
&= \mathop{\mathrm{E}}_{y \leftarrow Y}\left[\mathrm{CP}^{1/2}(H, H(X)|_{Y=y})\right] \\
&\le \mathop{\mathrm{E}}_{y \leftarrow Y}\left[\mathrm{CP}^{1/2}(H) \cdot \sqrt{\mathrm{CP}(X|_{Y=y}) + 2^{-\alpha}}\right] && \left(\begin{array}{l}\text{since } \mathrm{CP}(H, H(Z)) \le \\ \mathrm{CP}(H) \cdot (\mathrm{CP}(Z) + 2^{-\alpha})\end{array}\right) \\
&\le \mathop{\mathrm{E}}_{y \leftarrow Y}\left[\mathrm{CP}^{1/2}(H) \cdot \left(\mathrm{CP}^{1/2}(X|_{Y=y}) + \sqrt{2^{-\alpha}}\right)\right] && \text{(Cauchy-Schwartz/Jensen)} \\
&= \mathrm{CP}^{1/2}(H) \cdot \left(\left(\mathop{\mathrm{E}}_{y \leftarrow Y}\left[\mathrm{CP}^{1/2}(X|_{Y=y})\right]\right) + \sqrt{2^{-\alpha}}\right) \\
&= \mathrm{CP}^{1/2}(H) \cdot (\mathrm{CP}^{1/2}(X|Y) + \sqrt{2^{-\alpha}}) \\
&\le \sqrt{2^{-(q-\alpha)}} \cdot (\mathrm{CP}^{1/2}(X|Y) + \sqrt{2^{-\alpha}}) && (\text{since } |h| = q - \alpha) \\
&\le \sqrt{2^{-(q-\alpha)}} \cdot \left(\sqrt{\frac{2^{-\alpha}}{8}} + \sqrt{2^{-\alpha}}\right) \\
&< \sqrt{2^{-(q-\alpha)}} \cdot \left(\sqrt{2^{-\alpha}} \cdot \sqrt{2}\right) \\
&= \sqrt{2^{-(q-1)}} \ .
\end{aligned}$$

□

RESTATEMENT OF LEMMA 3.5.7

For any triple of (possibly correlated) random variables X, Y and Z,

$$\mathrm{CP}^{1/2}(X|Y) \le \mathrm{CP}^{1/2}(X|(Y,Z)) \le \sqrt{|\mathrm{Supp}(Z)|} \cdot \mathrm{CP}^{1/2}(X|Y) \ .$$

Proof. For each $y \in \mathrm{Supp}(Y)$ and $z \in \mathrm{Supp}(Z)$, let $v_{y,z}$ be the vector $(\Pr[X = x \wedge Z = z|Y = y])_{x \in \mathrm{Supp}(X)}$. With this, we compute:

$$\begin{aligned}
\left\|\sum_z v_{y,z}\right\|_2 &\le \sum_z \|v_{y,z}\|_2 && \text{(triangle inequality)} \\
&\le \sqrt{\mathrm{Supp}(Z|_{Y=y})} \cdot \left\|\sum_z v_{y,z}\right\|_2 && \text{(Cauchy-Schwartz/Jensen)} \\
&\le \sqrt{\mathrm{Supp}(Z)} \cdot \left\|\sum_z v_{y,z}\right\|_2 \ .
\end{aligned}$$

Since $\mathrm{CP}^{1/2}(X|_{Y=y}) = \|\sum_z v_{y,z}\|_2$ and $\mathrm{CP}^{1/2}\left((X|_{Y=y})\,|\,(Z|_{Y=y})\right) = \sum_z \|v_{y,z}\|_2$, taking expectations over Y for both sides yield our result. □

RESTATEMENT OF LEMMA 3.5.8

Let random variable H denote a random hash function from a family of pairwise-independent hash functions $\mathcal{H}$ with range $\{0,1\}^{\alpha}$. For any $\varepsilon > 0$, if $\mathrm{CP}(X) \le \varepsilon^2 \cdot 2^{-\alpha}$ and H is independent from X, then random variable $(H, H(X))$ is ε-close in statistical distance to uniform.

Proof. Let $D = 2^{q-\alpha}$ and $L = 2^{\alpha}$. We bound the statistical distance of $(H, H(X))$ from uniform as follows:

$$\begin{aligned}
\frac{1}{2}\left|(H,H(X)) - U_q\right|_1 &\leq \frac{\sqrt{DL}}{2}\left\|(H,H(X)) - U_q\right\|_2 \\
&= \frac{\sqrt{DL}}{2} \cdot \sqrt{\mathrm{CP}(H,H(X)) - 2^{-q}} \\
&\leq \frac{\sqrt{DL}}{2} \cdot \sqrt{\frac{1}{D}\left(\mathrm{CP}(X) + \frac{1}{L}\right) - \frac{1}{DL}} \\
&= \frac{\sqrt{\mathrm{CP}(X) \cdot L}}{2} \\
&\leq \frac{\varepsilon}{2} \ .
\end{aligned}$$

□

A.3 Establishing the Vadhan Condition

We restate and prove Claims 4.2.3, 4.2.4, and 4.2.5, and Proposition 4.2.7 from Section 4.2.1. Doing so would give a complete proof of Lemma 4.2.2, which states that every problem Π having a zero-knowledge argument system also satisfies the Vadhan condition. The three claims are proven using techniques from Vadhan [Vad3], and Proposition 4.2.7 is based on ideas from Ostrovsky [Ost]. Recall that (P, V) is the zero-knowledge argument system for Π, with simulator S.

Before proving the above claims and proposition, we first define the *conditional entropy* of two jointly distributed random variables as follows: For jointly distributed random variables X and Y, we define the *conditional entropy of X given Y* to be

$$\mathrm{H}(X|Y) \stackrel{\mathrm{def}}{=} \mathop{\mathrm{E}}_{y \leftarrow Y}\left[\mathrm{H}(X|_{Y=y})\right] = \mathop{\mathrm{E}}_{(x,y)\leftarrow(X,Y)}\left[\log \frac{1}{\Pr[X = x|Y = y]}\right] = \mathrm{H}(X,Y) - \mathrm{H}(Y) \ .$$

Next, recall the definition of $h(x)$ as stated by (4.1) in Section 4.2.1:

$$h(x) = \sum_{i=1}^{\ell}\left[\mathrm{H}(S(x)_{2i}) - \mathrm{H}(S(x)_{2i-1})\right] = \sum_{i=1}^{\ell} \mathrm{H}(S(x)_{2i}|S(x)_{2i-1}) \ , \qquad \text{(A.11)}$$

recalling that $\mathrm{H}(\cdot)$ is the *entropy* measure. The second equality in (A.11) follows the fact that the output of S_{2i} contains S_{2i-1}, and hence $\mathrm{H}(S_{2i}, S_{2i-1}) = \mathrm{H}(S_{2i})$.

Finally, recall that from (4.2) in Section 4.2.1, we have that for every $x \in \{0,1\}^*$, and every prover strategy P', the number of coins used by the honest verifier, denoted by $r(|x|)$, is:

$$r(|x|) = \sum_{i=1}^{\ell} \left[\mathrm{H}(\langle P', V\rangle(x)_{2i}) - \mathrm{H}(\langle P', V\rangle(x)_{2i-1})\right] = \sum_{i=1}^{\ell} \mathrm{H}(\langle P', V\rangle(x)_{2i} | \langle P', V\rangle(x)_{2i-1}) \ , \tag{A.12}$$

with the second equality following from the fact that the output of $\langle P', V\rangle_{2i}$ contains $\langle P', V\rangle_{2i-1}$, and hence $\mathrm{H}(\langle P', V\rangle_{2i}, \langle P', V\rangle_{2i-1}) = \mathrm{H}(\langle P', V\rangle_{2i})$.

RESTATEMENT OF CLAIM 4.2.3

Problem $(\Pi_\mathrm{Y} \setminus I_\mathrm{Y}, \Pi_\mathrm{N} \setminus I_\mathrm{N}) \in \mathrm{SZKP}$.

Proof. The following proposition is from [Vad3].

> **PROPOSITION A.3.1**
>
> (Based on [Vad3, Prop. 3.2].) Consider the problem CONDITIONAL ENTROPY APPROXIMATION = $(\mathrm{CEA}_\mathrm{Y}, \mathrm{CEA}_\mathrm{N})$, where $\mathrm{CEA}_\mathrm{Y} = \{((X,Y), r) : \mathrm{H}(X|Y) \geq r\}$ and $\mathrm{CEA}_\mathrm{N} = \{((X,Y), r) : \mathrm{H}(X|Y) \leq r-1\}$. Here (X, Y) is a *samplable joint distribution* specified by two circuits that use the same coin tosses. CONDITIONAL ENTROPY APPROXIMATION is complete for SZKP.

Given the above proposition, it suffices to show a reduction from $(\Pi_\mathrm{Y} \setminus I_\mathrm{Y}, \Pi_\mathrm{N} \setminus I_\mathrm{N})$ to CONDITIONAL ENTROPY APPROXIMATION. Our reduction is as follows: On input x, we construct circuits X and Y that sample from the following (joint) random variables.

(X, Y): Select $i \leftarrow \{1, \ldots, \ell(|x|)\}$, choose random coin tosses ω for the simulator, and output $(S_{2i}(x;\omega), S_{2i-1}(x;\omega))$.

When $x \in \Pi_\mathrm{Y} \setminus I_\mathrm{Y}$, we have $h(x) > r - 1/q$, and hence:

$$\mathrm{H}(X|Y) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathrm{H}(S_{2i}|S_{2i-1}) = \frac{h}{\ell} > \frac{r - 1/q}{\ell} = \frac{r}{\ell} - \frac{1}{q \cdot \ell} \ .$$

And when $x \in \Pi_\mathrm{N} \setminus I_\mathrm{N}$, we have have $h(x) < r - 2/q$, and hence:

$$\mathrm{H}(X|Y) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathrm{H}(S_{2i}|S_{2i-1}) = \frac{h}{\ell} < \frac{r - 2/q}{\ell} = \frac{r}{\ell} - \frac{2}{q \cdot \ell} \ .$$

This is what we need to prove, except the entropy gap is only $1/(q \cdot \ell)$. This can be increased to 1 by taking $q \cdot \ell$ independent samples from the joint distribution. That is, we define $(\overline{X}, \overline{Y}) = ((X_1, \ldots, X_{q \cdot \ell}), (Y_1, \ldots, Y_{q \cdot \ell}))$, where the (X_i, Y_i)'s are independent copies of (X, Y). Since $(\Pi_\mathrm{Y} \setminus I_\mathrm{Y}, \Pi_\mathrm{N} \setminus I_\mathrm{N})$ reduces to CONDITIONAL ENTROPY APPROXIMATION, Proposition A.3.1 gives us that $(\Pi_\mathrm{Y} \setminus I_\mathrm{Y}, \Pi_\mathrm{N} \setminus I_\mathrm{N}) \in \mathrm{SZKP}$. □

RESTATEMENT OF CLAIM 4.2.4

There exists an instance-dependent one-way function on I_{Y}.

Proof. The following proposition is from [Vad3].

> **PROPOSITION A.3.2**
>
> (From [Vad3, Lem. 3.10].) Let $K \subseteq \{0,1\}^*$ be any set. Assume that there exists a polynomial-time computable mapping that maps every $x \in K$ to samplable joint distributions (X, Y) and a parameter r such that $\mathrm{H}(X|Y) \leq r-1$, but $\mathrm{H}(X'|Y') \geq r$ for some (X', Y') indistinguishable from (X, Y). Then there exists an instance-dependent one-way function on K.

When $x \in \Pi_{\mathrm{Y}}$, then S is computationally indistinguishable from $\langle P, V\rangle$. So (X, Y), as defined in the proof of Claim 4.2.3 above, is indistinguishable from the joint distribution $(X', Y') = (\langle P, V\rangle_{2L}, \langle P, V\rangle_{2L-1})$, where random variable L denotes an independent uniform element of $\{1, \ldots, \ell\}$.

By (A.12), we have:

$$\mathrm{H}(X'|Y') = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathrm{H}(\langle P, V\rangle_{2i} | \langle P, V\rangle_{2i-1}) = \frac{r}{\ell} \ ,$$

for all $x \in \Pi_{\mathrm{Y}}$. And when $x \in I_{\mathrm{Y}} \subseteq \Pi_{\mathrm{Y}}$, we have have $h(x) < r - 1/q$ and hence:

$$\mathrm{H}(X|Y) = \frac{1}{\ell}\sum_{i=1}^{\ell} \mathrm{H}(S_{2i}|S_{2i-1}) = \frac{h}{\ell} < \frac{r - 1/q}{\ell} = \frac{r}{\ell} - \frac{1}{q \cdot \ell} \ .$$

Again, like in the proof of Claim 4.2.3, we can increase the entropy gap between $\mathrm{H}(X'|Y')$ and $\mathrm{H}(X|Y)$ to 1. Finally, we apply Proposition A.3.2 to establish our claim. □

RESTATEMENT OF CLAIM 4.2.5

For $\Pi \in \mathrm{HV\text{-}SZKA}$, we can take $I_{\mathrm{Y}} = \emptyset$.

Proof. For $\Pi \in \mathrm{HV\text{-}SZKA}$, the output of the simulator $S(x)$ is statistically close to $\langle P, V\rangle(x)$ for every $x \in \Pi_{\mathrm{Y}}$. This implies that $I_{\mathrm{Y}} = \emptyset$, since for every $x \in \Pi_{\mathrm{Y}}$, we have

$$h(x) > \sum_{i=1}^{\ell} [\mathrm{H}(\langle P, V\rangle_{2i}(x)) - \mathrm{H}(\langle P, V\rangle_{2i-1}(x))] - \mathrm{neg}(|x|) = r(|x|) - \mathrm{neg}(|x|) \ ,$$

with the last equality following from (A.12). □

Finally, we prove Proposition 4.2.7, restated below. Recall that the function $g_x(i, \omega) = (x, i, S(x; \omega)_{2i})$, as stated by (4.3) in Section 4.2.1.

RESTATEMENT OF PROPOSITION 4.2.7

Let g_x be as in (4.3) in Section 4.2.1. For every set $K \subseteq \{0,1\}^*$, if g_x is *not* an instance-dependent distributionally one-way function on K, then for every polynomial p, there exists a nonuniform PPT prover $\widetilde{P}$ such that

$$\Delta(\langle \widetilde{P}, V\rangle(x), S(x)) \leq \ell(|x|) \cdot \left(\frac{1}{p(|x|)} + 2 \cdot \Delta(\langle P_S, V\rangle(x), S(x))\right) \ ,$$

for infinitely many $x \in K$.

Proof. Let random variable $\mathcal{I}$ denote an independent uniform index $i \leftarrow \{1, 2, \ldots, \ell\}$, and let random variable Ω denote independent uniform coins ω for the simulator S. Recall the definition of instance-dependent distributionally one-way function as stated in Definition 2.4.7. If g_x is not an instance-dependent distributionally one-way function on K, then for any polynomial q, there exists a nonuniform PPT A such that the random variables $((\mathcal{I}, \Omega), S(x;\Omega)_{2\mathcal{I}})$ and $(A(S(x;\Omega)_{2\mathcal{I}}), S(x;\Omega)_{2\mathcal{I}})$ are $1/q(|x|)$-close for infinitely many $x \in K$. Let $K' \subseteq K$ be the infinite set of instances x for which the previously stated random variables are $1/q(|x|)$-close. Let the polynomial $p(|x|) = q(|x|) \cdot (1/\ell(|x|))$. For this point on, we will drop the mention of x and assume that $x \in K'$.

Since $\mathcal{I}$ is independent from the other random variables, we have that for all $i = 1, 2, \ldots, \ell$, the random variables

$$((i, \Omega), S(\Omega)_{2i}) \text{ and } (A(S(\Omega)_{2i}), S(\Omega)_{2i}) \text{ are } (1/p)\text{-close} \ , \tag{A.13}$$

since $\ell \cdot (1/q) = 1/p$.

For any interactive machine A and B, let random variable $\langle A, B\rangle[m_j]$ denote the transcript of messages exchanged between A and B conditioned on the first j messages being m_j. In other words, $\langle A, B\rangle[m_j] = \langle A, B\rangle|_{\langle A,B\rangle_j = m_j}$. It follows from definition that

$$\langle A, B\rangle[\langle A, B\rangle_j] \equiv \langle A, B\rangle \ , \tag{A.14}$$

for any index j.

By (A.13), and noting that P_S and $\widetilde{P}$ use (i, Ω) and $A(S(\Omega)_{2i})$ to produce their messages in round $2i+1$, respectively, we have that for every $i = 1, 2, \ldots, \ell$,

$$(\langle P_S, V\rangle[S_{2i}])_{2i+2} \text{ and } (\langle \widetilde{P}, V\rangle[S_{2i}])_{2i+2} \text{ are } (1/p)\text{-close} \ , \tag{A.15}$$

Using (A.14) and (A.15) above, we have that for every $i = 1, 2, \ldots, \ell$,

$$\begin{aligned}
&\Delta(\langle \widetilde{P}, V\rangle_{2i+2}, \langle P_S, V\rangle_{2i+2}) \\
&\quad = \Delta((\langle \widetilde{P}, V\rangle[\langle \widetilde{P}, V\rangle_{2i}])_{2i+2}, (\langle P_S, V\rangle[\langle P_S, V\rangle_{2i}])_{2i+2}) && \text{(by A.14)} \\
&\quad \le \Delta((\langle \widetilde{P}, V\rangle[S_{2i}])_{2i+2}, (\langle P_S, V\rangle[S_{2i}])_{2i+2}) \\
&\qquad + \Delta(\langle \widetilde{P}, V\rangle_{2i}, S_{2i}) + \Delta(\langle P_S, V\rangle_{2i}, S_{2i}) \\
&\quad \le (1/p) + \Delta(\langle \widetilde{P}, V\rangle_{2i}, S_{2i}) + \Delta(\langle P_S, V\rangle_{2i}, S_{2i}) && \text{(by A.15)} \\
&\quad \le (1/p) + \Delta(\langle \widetilde{P}, V\rangle_{2i}, S_{2i}) + \Delta(\langle P_S, V\rangle, S) \ . && \text{(A.16)}
\end{aligned}$$

We now prove the following by induction on $i = 0, 1, 2, \ldots, \ell$:

$$\Delta(\langle \widetilde{P}, V\rangle_{2i}, S_{2i}) \le i \cdot (1/p + 2 \cdot \Delta(\langle P_S, V\rangle, S)) \ . \tag{A.17}$$

Note that the case for $i = \ell$ establishes Proposition 4.2.7. The base case for $i = 0$ is trivial. We prove the inductive step as follows:

$$\begin{aligned}
&\Delta(\langle \widetilde{P}, V\rangle_{2i+2}, S_{2i+2}) \\
&\quad \le \Delta(\langle \widetilde{P}, V\rangle_{2i+2}, \langle P_S, V\rangle_{2i+2}) + \Delta(\langle P_S, V\rangle_{2i+2}, S_{2i+2}) \\
&\quad \le \Delta(\langle \widetilde{P}, V\rangle_{2i+2}, \langle P_S, V\rangle_{2i+2}) + \Delta(\langle P_S, V\rangle, S) \\
&\quad \le (1/p) + \Delta(\langle \widetilde{P}, V\rangle_{2i}, S_{2i}) + 2 \cdot \Delta(\langle P_S, V\rangle, S) && \text{(by A.16)} \\
&\quad \le (i+1) \cdot (1/p + 2 \cdot \Delta(\langle P_S, V\rangle, S)) && \text{(by induction on } i\text{)} \ .
\end{aligned}$$

This completes our proof of Proposition 4.2.7. □

BIBLIOGRAPHY

[AH] 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. Preliminary version in *FOCS'87*.

[ALM$^+$] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998. Preliminary version in *FOCS'92*.

[AS] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: a new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998. Preliminary version in *FOCS'92*.

[Bar] Boaz Barak. How to go beyond the black-box simulation barrier. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 106–115. IEEE Computer Society, 2001.

[BBR] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

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

[BD] Gilles Brassard and Ivan Damgård. "Practical IP" $\Leftarrow$ MA. In *Advances in Cryptology – CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 580–582. Springer, 1988.

[BDMP] Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM Journal on Computing*, 20(6):1084–1118, 1991.

[Bel] Mihir Bellare. A note on negligible functions. *Journal of Cryptology*, 15(4):271–284, 2002.

[BGG$^+$] Michael Ben-Or, Oded Goldreich, Shafi Goldwasser, Johan Håstad, Joe Kilian, Silvio Micali, and Phillip Rogaway. Everything provable is provable in zero-knowledge. In *Advances in Cryptology – CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 37–56. Springer, 1988.

[BHZ] Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Information Processing Letters*, 25:127–132, 1987.

[BKK] Joan F. Boyar, Stuart A. Kurtz, and Mark W. Krentel. A discrete logarithm implementation of perfect zero-knowledge blobs. *Journal of Cryptology*, 2(2):63–76, 1990.

[BL] Boaz Barak and Yehuda Lindell. Strict polynomial-time in simulation and extraction. *SIAM Journal on Computing*, 33(4):738–818, 2004. Preliminary version in *STOC'02*.

[Blu] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, pages 1444–1451. American Mathematical Society, 1987.

[BLV] Boaz Barak, Yehuda Lindell, and Salil Vadhan. Lower bounds for non-black-box zero knowledge. *Journal of Computer and System Sciences*, 72(2):321–391, 2006. Preliminary version in *FOCS'04*.

[BM] László Babai and Shlomo Moran. Arthur-Merlin games: A randomized proof system and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36:254–276, 1988.

[BMO] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. Perfect zero-knowledge in constant rounds. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 482–493. ACM Press, 1990.

[BOGKW] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: how to remove intractability assumptions. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 113–131. ACM Press, 1988.

[BP] Mihir Bellare and Erez Petrank. Making zero-knowledge provers efficient. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC)*, pages 711–722. ACM Press, 1992.

[BR] Mihir Bellare and Phillip Rogaway. Collision-resistant hashing: towards making UOWHFs practical. In *Advances in Cryptology – CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 470–484. Springer, 1997.

[CCM] Christian Cachin, Claude Crépeau, and Julien Marcil. Oblivious transfer with a memory-bounded receiver. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 493–502. IEEE Computer Society, 1998.

[CDG] David Chaum, Ivan Damgård, and Jeroen van de Graaf. Multiparty computations ensuring privacy of each party's input and correctness of the result. In *Advances in Cryptology – CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 87–119. Springer, 1987.

[CM] Christian Cachin and Ueli M. Maurer. Unconditional security against memory-bounded adversaries. In *Advances in Cryptology - CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 292–306. Springer, 1997.

[CS] Claude Crépeau and George Savvides. Optimal reductions between oblivious transfers using interactive hashing. In *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 201–221. Springer, 2006.

[CT] Thomas M. Cover and Joy A. Thomas. *Elements of information theory.* Wiley-Interscience, New York, NY, USA, second edition, 2006.

[Dam1] Ivan Damgård. Collision free hash functions and public key signature schemes. In *Advances in Cryptology – EUROCRYPT '87*, volume 304 of *Lecture Notes in Computer Science*, pages 203–216. Springer, 1987.

[Dam2] Ivan Damgård. On the existence of bit commitment schemes and zero-knowledge proofs. In *Advances in Cryptology - CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 17–27. Springer, 1989.

[Dam3] Ivan B. Damgård. Interactive hashing can simplify zero-knowledge protocol design without computational assumptions. In *Advances in Cryptology – CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 100–109. Springer, 1993.

[DC] Ivan Damgård and Ronald Cramer. On monotone function closure of perfect and statistical zero-knowledge. Technical Report CS-R9618, Centrum voor Wiskunde en Informatica (CWI), 1996. `http://www.cwi.nl/ftp/CWIreports/AA/CS-R9618.pdf`.

[DDPY1] Alfredo De Santis, Giovanni Di Crescenzo, Giuseppe Persiano, and Moti Yung. On monotone formula closure of SZK. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 454–465. IEEE Computer Society, 1994.

[DDPY2] Alfredo De Santis, Giovanni Di Crescenzo, Giuseppe Persiano, and Moti Yung. Image Density is complete for non-interactive-SZK. In *Automata, Languages and Programming, 25th International Colloquium, ICALP'98*, volume 1443 of *Lecture Notes in Computer Science*, pages 784–795. Springer, 1998.

[DHRS] Yan Zong Ding, Danny Harnik, Alon Rosen, and Ronen Shaltiel. Constant-round oblivious transfer in the bounded storage model. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 446–472. Springer, 2004.

[DOY] Giovanni Di Crescenzo, Tatsuaki Okamoto, and Moti Yung. Keeping the SZK-verifier honest unconditionally. In *Advances in Cryptology – CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 1997.

[DPP] Ivan B. Damgård, Torben P. Pedersen, and Birgit Pfitzmann. Statistical secrecy and multibit commitments. *IEEE Transactions on Information Theory*, 44(3):1143–1151, 1998.

[DR] Yan Zong Ding and Michael O. Rabin. Hyper-encryption and everlasting security. In *Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2002)*, volume 2285 of *Lecture Notes in Computer Science*, pages 1–26. Springer, 2002.

[DS] Irit Dinur and Shmuel Safra. The importance of being biased. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 33–42. ACM Press, 2002.

[ESY] Shimon Even, Alan L. Selman, and Yacov Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, 1984.

[FGL$^+$] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996. Preliminary version in *FOCS'91*.

[FGM$^+$] Martin Fürer, Oded Goldreich, Yishay Mansour, Michael Sipser, and Stathis Zachos. On completeness and soundness in interactive proof systems. *Advances in Computing Research*, 5:429–442, 1989. Preliminary version in *FOCS'87*.

[For] Lance Fortnow. The complexity of perfect zero-knowledge. *Advances in Computing Research: Randomness and Computation*, 5:327–343, 1989.

[FRS] Lance Fortnow, John Rompel, and Michael Sipser. On the power of multi-prover interactive protocols. *Theoretical Computer Science*, 134(2):545–557, 1994.

[FS] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 416–426. ACM Press, 1990.

[GGL] Oded Goldreich, Shafi Goldwasser, and Nathan Linial. Fault-tolerant computation in the full information model. *SIAM Journal on Computing*, 27(2):506–544, 1998.

[GK1] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, 1996.

[GK2] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996. Preliminary version in *ICALP'90*.

[GK3] Oded Goldreich and Eyal Kushilevitz. A perfect zero-knowledge proof system for a problem equivalent to the discrete logarithm. *Journal of Cryptology*, 6:97–116, 1993.

[GMR1] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. Preliminary version in *STOC'85*.

[GMR2] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, 1988. Preliminary version in *FOCS'84*.

[GMW1] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC)*, pages 218–229. ACM Press, 1987.

[GMW2] 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(1):691–729, 1991. Preliminary version in *FOCS'86*.

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

[Gol1] Oded Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6(1):21–53, 1993.

[Gol2] Oded Goldreich. *Foundations of Cryptography: Basic Tools.* Cambridge University Press, 2001.

[Gol3] Oded Goldreich. Zero-knowledge twenty years after its invention. `http://www.wisdom.weizmann.ac.il/~oded/zk-tut02.html`, March 2004.

[GS] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. *Advances in Computing Research: Randomness and Computation*, 5:73–90, 1989.

[GSV1] Oded Goldreich, Amit Sahai, and Salil Vadhan. Honest verifier statistical zero-knowledge equals general statistical zero-knowledge. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 399–408. ACM Press, 1998.

[GSV2] Oded Goldreich, Amit Sahai, and Salil Vadhan. Can statistical zero-knowledge be made non-interactive?, or On the relationship of SZK and NISZK. In *Advances in Cryptology – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 467–484. Springer, 1999.

[GV] Oded Goldreich and Salil P. Vadhan. Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In *IEEE Conference on Computational Complexity*, pages 54–73. IEEE Computer Society, 1999.

[HHK$^+$] Iftach Haitner, Omer Horvitz, Jonathan Katz, Chiu-Yuen Koo, Ruggero Morselli, and Ronen Shaltiel. Reducing complexity assumptions for statistically-hiding commitment. In *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 58–77. Springer, 2005.

[HHRS] 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. Technical Report TR07-038, Electronic Colloquium on Computational Complexity, 2007.

[HILL] 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. Preliminary versions in *STOC'89* and *STOC'90.*

[HR1] Iftach Haitner and Omer Reingold. A new interactive hashing theorem. Technical Report TR06–096, Electronic Colloquium on Computational Complexity, 2006.

[HR2] Iftach Haitner and Omer Reingold. Statistically-hiding commitment from any one-way function. Technical Report 2006/436, Cryptology ePrint Archive, 2006. To appear in *STOC'07*.

[IL] Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 230–235, 1989.

[ILL] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 12–24. ACM Press, 1989.

[Imp] Russell Impagliazzo. A personal view of average-case complexity. In *Proceedings of the Tenth Annual Structure in Complexity Theory Conference*, pages 134–147. IEEE Computer Society, 1995.

[IOS] Toshiya Itoh, Yuji Ohta, and Hiroki Shizuya. A language-dependent cryptographic primitive. *Journal of Cryptology*, 10(1):37–49, 1997.

[IY] Russell Impagliazzo and Moti Yung. Direct minimum-knowledge computations (extended abstract). In *Advances in Cryptology – CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 40–51. Springer, 1987.

[Kar] Richard M. Karp. Reducibility among combinatorial problems. In J. W. Thatcher and R. E. Miller, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, Inc., 1972.

[KK] Jonathan Katz and Chiu-Yuen Koo. On constructing universal one-way hash functions from arbitrary one-way functions. Technical Report 2005/328, Cryptology ePrint Archive, 2005.

[KMS] Bruce Kapron, Lior Malka, and Venkatesh Srinivasan. A characterization of non-interactive instance-dependent commitment-schemes (NIC). In *Automata, Languages and Programming, 34th International Colloquium, ICALP 2007*, Lecture Notes in Computer Science. Springer, 2007.

[KSS] Jeff Kahn, Michael Saks, and Cliff Smyth. A dual version of Reimer's inequality and a proof of Rudich's conjecture. In *15th Annual IEEE Conference on Computational Complexity*, pages 98–103, 2000.

[Lev] Leonid A. Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.

[Mau] Ueli M. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1):53–66, 1992.

[MV] Daniele Micciancio and Salil Vadhan. Statistical zero-knowledge proofs with efficient provers: lattice problems and more. In *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 282–298. Springer, 2003.

[Nao] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991. Preliminary version in *CRYPTO'89*.

[Ngu] Minh-Huyen Nguyen. *Zero knowledge and efficient provers.* PhD thesis, Harvard University, Cambridge, MA, USA, 2006.

[NOVY] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Perfect zero-knowledge arguments for NP using any one-way permutation. *Journal of Cryptology*, 11(2):87–108, 1998. Preliminary version in *CRYPTO'92*.

[NV] Minh-Huyen Nguyen and Salil Vadhan. Zero knowledge with efficient provers. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 287–295. ACM Press, 2006.

[NY] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 33–43. ACM Press, 1989.

[Oka] Tatsuaki Okamoto. On relationships between statistical zero-knowledge proofs. *Journal of Computer and System Sciences*, 60(1):47–108, 2000. Preliminary version in *STOC'96*.

[Ost] Rafail Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. In *Proceedings of the 6th Annual Structure in Complexity Theory Conference*, pages 133–138. IEEE Computer Society, 1991.

[OVY] Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Fair games against an all-powerful adversary. *AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 155–169, 1993. Preliminary version in *SEQUENCES'91*.

[OW] Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Proceedings of the 2nd Israel Symposium on Theory of Computing Systems*, pages 3–17. IEEE Computer Society, 1993.

[Ped] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology – CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1991.

[PS] Rafael Pass and Abhi Shelat. Unconditional characterizations of non-interactive zero-knowledge. In *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 118–134. Springer, 2005.

[PT] Erez Petrank and Gábor Tardos. On the knowledge complexity of NP. *Combinatorica*, 22(1):83–121, 2002. Preliminary version in *FOCS'96*.

[Rom] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 387–394, 1990.

[RTV] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2004.

[Rud] Steven Rudich. *Limits on the Provable Consequences of One-Way Functions.* PhD thesis, U.C. Berkeley, 1988.

[Sim] Daniel Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Advances in Cryptology – EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345. Springer, 1998.

[Sip] Michael Sipser. *Introduction to the Theory of Computation.* Thomson Course Technology, Boston, MA, USA, second edition, 2005.

[SV] Amit Sahai and Salil Vadhan. A complete problem for statistical zero knowledge. *Journal of the ACM*, 50(2):196–249, 2003. Preliminary version in *FOCS'97*.

[TW] Martin Tompa and Heather Woll. Random self-reducibility and zero knowledge interactive proofs of possession of information. In *Proceedings of the 28th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 472–482. IEEE Computer Society, 1987.

[Vad1] Salil P. Vadhan. *A Study of Statistical Zero-Knowledge Proofs.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.

[Vad2] Salil P. Vadhan. Interactive proofs & zero-knowledge proofs. `http://www.eecs.harvard.edu/~salil/papers/pcmi-abs.html`, 2000.

[Vad3] Salil P. Vadhan. An unconditional study of computational zero knowledge. *SIAM Journal on Computing*, 36(4):1160–1214, 2006. Preliminary version in *FOCS'04*.

[Yao] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 162–167. IEEE Computer Society, 1986.