

A UNIVERSAL FRAMEWORK FOR CONCURRENT SECURITY

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Muthuramakrishnan Venkitasubramaniam

August 2011

© 2010 Muthuramakrishnan Venkatasubramaniam
ALL RIGHTS RESERVED

A UNIVERSAL FRAMEWORK FOR CONCURRENT SECURITY

Muthuramakrishnan Venkitasubramaniam, Ph.D.

Cornell University 2011

Cryptography today has evolved far beyond its traditional goal of secure message transmission. Through the notion of secure computation, a set of mutually distrustful agents can collaborate to accomplish a common goal while preserving each agent's privacy to a maximal extent. In the seminal works of Yao and Goldreich, Micali and Wigderson, it was shown that any computational task can be securely implemented through a protocol. Traditionally, the rules governing privacy for these protocols have been designed to work only when a single execution running in isolation. However, with the advent of the Internet, many transactions occur simultaneously, and the protocols designed for the single execution setting fail to remain secure in a concurrent setting.

While both the need and definitions for concurrent security were realized in the early 90's, practical protocols that are concurrently secure are lacking. The protocols designed for concurrent security, thus far, have mostly relied on having a trusted setup or a relaxed definition of security.

In this thesis, we put forward a unified framework for the construction of concurrently secure protocols both with and without trusted set-up. This framework not only provides a conceptually simple solution for essentially all previous results, but also significantly improves efficiency and reduces the requirements on the trusted setup used in these works. Moreover, in several setup models, our constructions are tight with respect to computational assumptions and efficiency.

BIOGRAPHICAL SKETCH

The author was born in Tirunelveli, a small district located in the south of Tamil Nadu, India on August 8, 1982. His parents, Venkitasubramaniam and Kanthimathi, named him *Nagaraj* after the Hindu God worshipped in a temple at Mannarsalai, Kerala. As per family tradition, his official name was to be his maternal grandfather's, Muthuramakrishnan.

Muthu spent a great deal of his childhood learning tennis and computers from his father and a ton of mathematics from his mother. His childhood memories include coding in Basic on an Intel-8086 based IBM-PC and watching Tamil, Hindi and Malayalam movies. He attended high-school in Chennai, where he developed passion for mathematics and programming. He was exposed to a lot of non-routine problem-solving when he attended the training camp for the International Mathematical Olympiad in Mumbai, India. In the meanwhile, he spent several sleepless nights, diligently preparing for the Indian Institute Of Technology, Joint Entrance Examination. Securing 21st rank in the examination allowed him to pursue Computer Science at IIT-Madras where he gained interest in Theoretical Computer Science, a field that allowed him to combine his passions. Four years later, he graduated with a bachelors degree. Drawn to theoretical research, owing to his fascinating mentor at IIT, Pandu Rangan, he joined the Ph.D. program at Cornell University in fall 2004, where he spent six beautiful years working on theoretical research under the supervision of Rafael Pass.

This thesis is dedicated to my wife who has always been supportive and loving.

ACKNOWLEDGEMENTS

First, I wish to express my sincere gratitude to Rafael Pass, my advisor and mentor. With his enthusiasm, his inspiration, and his great efforts to explain things clearly and simply, he helped make cryptography fun for me. I am deeply grateful to him for his ability to understand me and help me harness my strengths and bolster my confidence, while at the same time, in critical moments, providing sincere and determined opinions, helping me making the right decisions. I am also very grateful to him for investing a great number of hours in teaching me the spirit of good research.

During my initial years at Cornell, I had the fortune of interacting with Johannes Gehrke and Dexter Kozen, from whom, I learned a great deal about critical thinking and understanding the principles of scientific research. I also benefited a lot from the fantastic courses taught by Jon Kleinberg, Robert Kleinberg, Dexter Kozen, David Shmoys, Eva Tardos, and, David Williamson.

I am indebted to my many student colleagues for providing a stimulating and fun environment in which to learn and grow. I am especially grateful to Huijia Lin and Wei-lung Dustin Tseng for our countless discussions and mutual exchange of ideas on all topics. Both of them have had a strong impact on the research in this thesis and are both dear friends.

Other researchers that have deeply influenced my research include Ran Canetti, Rosario Gennaro, Dov Gordon, Hugo Krawczyk, Tal Rabin, and, Vinod Vaikuntanathan. I am very grateful for my discussions with them.

I wish to thank my wife, Shravya Markandeya, for supporting, encouraging and most of all, being patient and accomodating, especially during the finishing stages of my Ph.D. I also wish to thank my best friends Bistra Dilkina, Tudor Marian, and, Anton Morozov, for helping me get through some difficult times, and for all

the entertainment they provided.

My understanding of the topics in this thesis have benefited from discussions with Thanh Nguyen, Ian Kash, Yogeshwer Sharma, Yeejiun Song, Smita Shankar, Linga Prakash, and, Daria Sorokina whose comments has had an important impact on this thesis. I am grateful to the secretaries in the computer science department, for helping the departments to run smoothly and for assisting me in many different ways. Becky Stewart and Stephanie Meik, deserve special mention.

Finally, my parents Venkatasubramaniam and Kanthimathi who have given me the opportunity of an education from the best institutions.

Please forgive me for any omissions.

CONTENTS

Biographical Sketch	iii
Dedication	iv
Acknowledgements	v
Contents	vii
List of Figures	ix
1 Introduction	1
1.1 A Unified Framework	3
1.2 Techniques	7
1.3 Overview	8
2 Preliminaries	10
2.1 Basic Notation	10
2.1.1 General Notation	10
2.1.2 Protocol Notation	12
2.2 Basic Notions	14
2.2.1 Basic Complexity Classes	14
2.2.2 Indistinguishability	15
2.2.3 Interactive Proofs and Arguments	16
2.2.4 Witness Indistinguishability	16
2.2.5 Zero-Knowledge	18
2.2.6 Commitment Schemes	19
3 General Framework for Secure Computation	22
3.1 Traditional UC	22
3.2 A Generalized Version of UC	26
3.3 Main Result	26
3.4 UC-puzzles	28
4 Strong Non-Malleable Witness Indistinguishable Proofs	30
4.1 Definition	30
4.2 \mathcal{SNMWI} from Simulation-Extractability	32
4.3 \mathcal{SNMWI} from any Non-Malleable Commitment	34
4.3.1 Non-malleable commitment schemes	35
4.3.2 \mathcal{SNMWI} Argument of Knowledge Protocol $\langle P_s, V_s \rangle$	36
4.4 Robust \mathcal{SNMWI} Arguments	45
4.5 Sequential composition of \mathcal{SNMWI} Arguments	47
5 Proof of the Main Theorem	54
5.1 IdealZK Functionality	56
5.2 Oblivious Transfer	57
5.3 The Puzzle Lemma	59
5.3.1 Step 1: Simulating puzzle interactions	63

5.3.2	Step 2: Simulating Zero-Knowledge	68
6	Applications of the General Framework	85
6.1	Non-Uniform UC	85
6.2	Quasi-Polynomial UC	88
6.3	UC in the Common Reference String model	90
6.4	UC in the Uniform Reference String model	91
6.5	UC in the Key Registration model	92
6.6	UC in the Sunspots model	94
6.7	UC in the Tamper-Proof Hardware Model	98
6.8	Stand-alone Model	101
7	Lower Bounds for Non-Uniform UC-Security	106
7.1	On Existence of Evasive Sets	112
8	Efficient Concurrent Zero-Knowledge	116
8.1	Results	118
8.2	Techniques	120
8.3	Open questions	121
8.4	Black-Box Concurrent Zero-Knowledge	122
8.5	Description of the protocol	122
8.6	Warm-up: Simulating Static Non-Aborting Adversaries	125
8.7	The Simulator	129
8.8	Analysis of the Simulator	131
8.9	Concurrent Computational \mathcal{ZK} Proof for \mathbf{NP}	137
8.10	Concurrent \mathcal{ZK} Arguments from OWFs	144
8.11	Concurrent Perfect \mathcal{ZK} Proofs for languages in \mathbf{NP}	144
8.12	Unconditional \mathcal{WI} Proof of Knowledge for Specific Languages	145
8.13	Concurrent Perfect \mathcal{ZK} Proof for Graph Non-Isomorphism	146
	Bibliography	147

LIST OF FIGURES

4.1	Strongly Non-Malleable \mathcal{WI} Argument of Knowledge for NP	37
4.2	The two cases in a man-in-the-middle execution of $\langle P_s, V_s \rangle$ with adversary A	38
4.3	The two cases E_1 and E_2 in a man-in-the-middle execution of $\langle P_s^2, V_s^2 \rangle$ with adversary A	48
6.1	Uniform Reference String functionality	91
6.2	Key Registration functionality	93
6.3	The Sunspots functionality	95
6.4	Wrap functionality	99
6.5	The Synchronous Communication functionality	102
7.1	Ideal Bit Commitment Functionality	106
8.1	Concurrent Perfect \mathcal{ZK} Argument for NP	124
8.2	Simulator for Perfect Concurrent Zero-Knowledge Argument $\langle P, V \rangle$	130
8.3	Computational \mathcal{ZK} Proof for NP	139
8.4	\mathcal{WI} Proof of Knowledge for 1OF2GRAPHISO	145
8.5	Perfect Concurrent \mathcal{ZK} Proof for GraphNonIso	146

CHAPTER 1

INTRODUCTION

The notion of *secure multi-party computation* allows m mutually distrustful parties to securely compute a functionality $f(\bar{x}) = (f_1(\bar{x}), \dots, f_m(\bar{x}))$ of their corresponding private inputs $\bar{x} = x_1, \dots, x_m$, such that party P_i receives the value $f_i(\bar{x})$. Loosely speaking, the security requirements are that the parties learn nothing more from the protocol than their prescribed output, and that the output of each party is distributed according to the prescribed functionality. This should hold even in the case that an arbitrary subset of the parties maliciously deviates from the protocol.

The above security guarantees are traditionally formalized using the *simulation paradigm* [37, 38]. The basic idea, which originates in [34], is to say that a protocol π securely realizes f if running π emulates an ideal process where all parties secretly provide inputs to an imaginary trusted party that computes f and returns the outputs to the parties; more precisely, any “harm” done by a polynomial-time adversary in the real execution of π , could have been done by a polynomial-time *simulator* in the ideal process.

Shortly after its conceptualization, strong results were established for secure multi-party computation. Specifically, it was shown that any probabilistic polynomial-time computable multi-party functionality can be securely computed, assuming the existence of enhanced trapdoor permutations [77, 34]. The original setting in which secure multi-party protocols were investigated, however, only allowed the execution of a single instance of the protocol at a time; this is the so called *stand-alone setting*. A more realistic setting, is one which allows the concurrent execution of protocols. In the *concurrent setting*, many protocols are executed at the same time. This setting presents the new risk of a coordinated attack in which an adversary interleaves many different executions of a protocol and chooses its messages in each instance based on other partial executions of the protocol. The strongest (but also most realistic) setting for concurrent security—called *Universally Composable* (UC) security [12, 69, 24], or environmental-security—considers the execution of an unbounded number of concurrent protocols, in an arbitrary, and adversarially controlled, network environment. Unfortunately, security in the stand-alone setting does not imply security in the concurrent setting. In fact,

without assuming some trusted set-up, the traditional simulation-based notion of concurrent security, and in particular UC security, cannot be achieved in general [15, 18, 52].

To circumvent the broad impossibility results, two distinct veins of research can be identified in the literature.

Trusted set-up models: A first vein of work initiated by Canetti and Fischlin [15] and Canetti, Lindell, Ostrovsky and Sahai [19] (see also e.g., [2, 14, 44, 20]) considers constructions of UC-secure protocol using various trusted set-up assumptions, where the parties have limited access to a trusted entity. (See [13] for a recent survey of various different set-up assumptions.) In many situations, however, trusted set-up is hard to come by (or at least expensive). An important question is to identify the weakest possible set-up that allows us to obtain general feasibility results for UC security.

Relaxed models of security: In some situations, trusted set-up is not only expensive, but might not even exist. It is thus imperative to have a notion of concurrent security that can be realized without trusted set-up. Another vein of work considers relaxed models of security such as *quasi-polynomial simulation* [63, 71, 6] or *input-indistinguishability* [57]. These works, however, only provide weak guarantees about the computational advantages gained by an adversary in a concurrent execution of the protocol. As such, currently, there are no known protocols—without trusted set-up—that can be used to securely compute “computationally-sensitive” functionality (such as e.g., private data-base queries, proof-of-work protocols [26, 28], or player bridge or poker on the Internet [34]) in a fully concurrent setting.¹

In this work we address both of the above research goals by presenting a *unified framework* for the construction of UC secure protocols—both with, and without, trusted set-up. This framework not only provides a conceptually simpler solution for essentially all general UC-feasibility results (e.g., [19, 14, 2, 44, 20, 45, 41]), but also allows us to (often significantly) improve the round-complexity and complexity theoretic assumptions. Interestingly, our new results even improve the round complexity of *stand-alone* secure computation. As far as we know this is the first

¹Yet another vein of work considers relaxed notions of concurrency, such as *bounded concurrency* [1, 66, 53, 64]. In this work, we, however, focus only on *full concurrency*, where no restrictions on the number of concurrent executions are made.

improvement to the original work of Goldreich, Micali and Wigderson [34], assuming only trapdoor permutations.) More importantly, this framework allows us to consider weaker trusted set-up models (e.g., the existence of a *single* imperfect reference string, or an “unrestricted” timing model) and new relaxed models of security. In particular, we present a new model of concurrent security, called *Non-Uniform UC*, which allows us to achieve—without any trusted set-up—the first “fully-concurrent” secure computation protocol that provides strong guarantees about the computational advantages gained by an adversary. We also complement our positive results with new lower bounds, showing that our results (both with and without trusted set-up) are essentially tight (often optimal)—both in terms of round complexity and in terms of complexity-theoretic assumptions. As such, our framework helps in characterizing models in which UC security is realizable, and also at what cost.

We start by outlining our framework, and then state our main result followed by some applications.

1.1 A Unified Framework

Earlier results on UC secure computation all rely on quite different techniques. Roughly speaking, to prove that a protocol is concurrently secure, one needs to show two different properties: 1) *concurrent simulation*, and 2) *concurrent non-malleability*. Intuitively, concurrent simulation amounts to providing the simulator with a “trapdoor” that allows it to emulate players without knowing their inputs. On the other hand, concurrent non-malleability, requires showing that an adversary cannot make use of messages received in one execution to “cheat” in another execution; this is often achieved by providing a technique which enables the simulator to have *different* trapdoors for each player (in a sense an “identity-based” trapdoor using the terminology of [14]), and showing that the trapdoor for one player does not reveal a trapdoor for another.

The simulation part is usually “easy” to achieve. Consider, for instance, the Common Reference String model—where the players have access to a public reference string that is ideally sampled from some distribution. In this model it is easy to provide the simulator with a single trapdoor; it could, for instance, be the

inverse of the CRS through a one-way function. However, achieving concurrent non-malleability is significantly harder. In this particular case, [19] solve the problem by embedding the public-key of a CCA-secure encryption scheme in the CRS, but in general, quite different techniques are employed in each model. Yet the same phenomena persists: concurrent simulation is easy to achieve, but concurrent non-malleability requires significantly more work, and often stronger set-up and/or stronger computational assumptions and/or larger round-complexity.

In this work, we provide a technique showing that concurrent simulation is sufficient—i.e., it is sufficient to provide the simulator with a *single* trapdoor. In a nutshell, once such a trapdoor is established, concurrent non-malleability (and thus full UC-security) can be achieved by further relying on a *stand-alone secure non-malleable commitment scheme*[25].

To formalize “concurrent simulation” we define the notion of a *UC-puzzle*—which, intuitively, is a protocol with the property that no adversary can successfully complete the puzzle and also obtain a trapdoor, but there exists a simulator who can generate (correctly distributed) puzzles together with trapdoors.

A commitment scheme, often described as the “digital” analogue of sealed envelopes, enables a sender to commit itself to a value while keeping it secret from the receiver. A commitment scheme, is said to be *stand-alone non-malleable*, if it is infeasible for an adversary to *maul* a commitment to a value v into a commitment to a related value \tilde{v} . To achieve concurrent non-malleability, we will in fact, rely on a non-malleable commitment that satisfies an additional property, referred to as *robustness*, which requires the commitment to also be non-malleable w.r.t arbitrary k -round protocols. However, as it turns out, essentially all non-malleable commitments satisfy this property.

Finally, we rely on *stand-alone secure semi-honest oblivious transfer*. This assumption, as we already know, is necessary even when considering stand-alone secure computation. Informally, oblivious transfer[72], enables a receiver to privately select and obtain one (and at most one) out of two values submitted by a sender. Semi-honest oblivious transfer considers adversarial senders (or receivers) that try to learn as much as possible from the messages while following the prescribed protocol.

The main result that we obtain in this work is the following:

Theorem 1 (Informally stated). Assume the existence of a $t_1(\cdot)$ -round UC-secure puzzle Σ using some set-up \mathcal{T} , the existence of $t_2(\cdot)$ -round robust non-malleable commitments and the existence of $t_3(\cdot)$ -round stand-alone secure semi-honest oblivious-transfer protocol. Then, for every m -ary functionality f , there exists a $O(t_1(\cdot) + t_2(\cdot) + t_3(\cdot))$ -round protocol Π —using the same set-up \mathcal{T} —that UC-realizes f .

Since, the best known construction of non-malleable commitment requires only $O(1)$ -rounds [50] and can be based on any one-way function which in turn are implied by the existence of semi-honest oblivious transfer protocols, we obtain the following theorem.

Theorem 2 (Informally stated). Assume the existence of a $t_1(\cdot)$ -round UC-secure puzzle Σ using some set-up \mathcal{T} , and the existence of $t_3(\cdot)$ -round stand-alone secure semi-honest oblivious-transfer protocol. Then, for every m -ary functionality f , there exists a $O(t_1(\cdot) + t_3(\cdot))$ -round protocol Π —using the same set-up \mathcal{T} —that UC-realizes f .

In fact, as we show in several set-ups, the UC-puzzles are not only sufficient, but are also necessary. Since oblivious-transfer is necessary to achieve secure computation, in models where we prove that our UC-puzzles are necessary, we obtain tight feasibility conditions.

As such, UC-puzzles *fully characterize* in what set-up models UC security is possible. This characterization not only gives insight into what is needed to establish UC security, as we shall see, in previously studied models, $O(1)$ -round UC-puzzles are “easy” to construct. As such, Theorem 1 provides a conceptually simple and unified proof of known results, while at the same time reducing the trusted set-up, the computational assumptions and/or the round-complexity. We briefly highlight some results obtained by instantiating our framework with known constructions of non-malleable commitments [25, 67, 51, 49, 50, 40].² In all the results below we focus only on *static* adversaries.

UC in the “imperfect” string model. Canetti, Pass and Shelat [20] consider UC security where parties have access to an “imperfect” reference string—

²Interesting, for many of our results, we get quite substantial improvements “already” by relying on the original DDN-construction [25].

called a “sunspot”—that is generated by any arbitrary efficient min-entropy source (obtained e.g., by measurement of some physical phenomenon). The CPS-protocol, however, requires m communicating parties to share m reference strings, each of them generated using fresh entropy.

Our results show that, somewhat surprisingly, a *single* imperfect reference string is sufficient for UC security. This stands in sharp contrast to the general study of randomness extraction, where single-source extraction from arbitrary *efficient* sources is impossible, but extraction from multiple sources is feasible!³

UC in the timing model. Dwork, Naor and Sahai [27] introduced the *timing model*, where all players are assumed to have access to clocks with a certain drift. In this model, they rely on *delays* and *time-outs* to obtain a $O(1)$ -round concurrent zero-knowledge protocol. Kalai, Lindell and Prabhakaran [44] subsequently presented a concurrent secure computation protocol in the timing model; whereas the timing model of [27] does not impose a maximal upper-bound on the clock drift, the protocol of [44] requires the clock-drift to be “small”; furthermore, it requires $\omega(n)$ rounds and an extensive use of delays (roughly $n\Delta$, where Δ is the latency of the network).

Our results establish that UC security is possible also in the “unrestricted” timing model (where the clock drift can be “large”); additionally, we reduce the use of delays to only $O(\Delta)$, and only require an $O(1)$ -round protocol; in fact, we also establish lower bounds showing that the *run time* (and thus the use of delays) of our protocol is optimal up to a constant factor. This model however requires a slight modification of the framework to incorporate delays and time-outs and is beyond the scope of the thesis.

UC with quasi-polynomial simulation. Pass [63] proposed a relaxation of the standard simulation-based definition of security, allowing for a super polynomial-time, or Quasi-polynomial simulation (QPS). Prabhakaran and Sahai [71] and Barak and Sahai [6] recently obtained general multi-party protocols that are concurrently QPS-secure without any trusted set-up, but rely on strong complexity assumptions.

³Note that the results of Trevisan and Vadhan [76] only show that extraction from sources with size bounded by some *fixed* polynomial is possible. In contrast, traditional techniques show that extraction from sources with *arbitrary* polynomial running-time is impossible.

Our results show how to construct using optimal complexity assumptions, while at the same time achieving a stronger (and more meaningful) notion of security, which (in analogy with [63]) requires that indistinguishability of simulated and real executions holds also for all of *quasi-polynomial time*; in contrast, [6] only achieves indistinguishability w.r.t distinguishers with running-time smaller than that of the simulator.⁴ We complement this result by a lower bound showing that our complexity assumptions, in essence, are necessary to achieve QPS security.

Stand-alone secure multi-party computation. The original construction of *stand-alone* secure m -party computation by Goldreich, Micali and Wigderson relies only on the existence of enhanced trapdoor permutations, but requires $O(m)$ -communication rounds. We obtain the first (asymptotic) improvement to the round complexity of this results without strengthening the underlying assumption. By relying on the original DDN construction of non-malleable commitments [25] (see also [51]) we already obtain a $\log \log m$ -round secure computation protocol. If instead relying on the recent construction from [50, 40], we obtain a $O(1)$ round protocol.⁵

Our results also establishes that, on top of stand-alone semi-honest oblivious-transfer protocol, no further assumptions are necessary to establish UC secure computation in e.g., the *uniform random string (URS) model* [19] or the “*multi-CRS*” model [41]; earlier results required additional assumptions (e.g., dense crypto systems).

1.2 Techniques

By relying on previous results [64, 66, 52, 19, 34] the construction of a UC protocol for realizing any multi-party functionality reduces to the task of construct-

⁴In essence, this means that anything an attacker can learn “on-line” (in poly-time) can be simulated “off-line” (in qpoly-time) in a way that is indistinguishable also “off-line”. In this language, [6] only achieves on-line indistinguishability.

⁵Earlier results improve the round-complexity by making stronger assumptions: (1) assuming *dense crypto systems*, Katz, Ostrovsky and Smith [46] achieved $O(\log m)$ rounds; (2) assuming collision-resistant hash-function, and additionally relying on *non-black box simulation* [1], Pass [64] achieved $O(1)$ rounds. Our results only use black-box simulation; as such they are a significant improvement also over any known protocol using black-box simulation (and in particular [46]).

ing a zero-knowledge protocol that is *concurrently simulatable*, and *concurrently simulation-sound* [75]—namely, even if an adversary receives multiple concurrent *simulated* proofs, it will not be able to prove any false statements. Concurrent simulation is easy to achieve in any model where we have a UC-puzzle. The tricky part is to obtain a zero-knowledge proof that is simultaneously straight-line extractable and simulation-sound.

To achieve this we introduce a new notion of non-malleability for interactive proofs, called *strong non-malleable witness indistinguishable* (\mathcal{SNMWI}). Informally, \mathcal{SNMWI} extends the notion of *strong witness indistinguishability* [31] to a man-in-the-middle setting: consider a man-in-the-middle attacker (MIM) that is participating in two interactions, a left interaction where it is acting as a verifier, and a right interaction where it is acting as a prover. \mathcal{SNMWI} requires that whenever the common inputs in the left interaction are indistinguishable, so are the views of *and witnesses* used by the MIM. \mathcal{SNMWI} is related to (and inspired by) the notion of *non-malleable witness indistinguishability*, recently introduced by Ostrovsky, Persiano and Visconti [61], but the actual security requirements are quite different.

As we show, \mathcal{SNMWI} is a relaxation of the notion of *simulation-extractability* [25, 67], and, as such, potentially easier to achieve. In particular, one of our main technical contributions is a construction of \mathcal{SNMWI} arguments of knowledge from any robust non-malleable commitment (with only constant overhead in round-complexity).

1.3 Overview

The main contribution of the thesis is a unified framework to securely realize any computational task in an arbitrary set-up infrastructure. Along the way, we introduce a new notion of strong non-malleable witness indistinguishability and construct protocols for the same. As an aside, we also show how to construct efficient protocols for the basic task of zero-knowledge in a slightly restricted concurrent setting.

Chapter 2 - Preliminaries We introduce basic notation and recall basic notions that will be used throughout the thesis.

Chapter 3 - General Framework for Secure Computation We recall the basic definition of concurrent or universal composable security (UC) and show how to extend it to any general set-up model or relaxed-security model. We introduce the notion of a UC-puzzle that characterizes when concurrent security is achievable in a model. Then we state our main feasibility theorem.

Chapter 4 - Strong Non-Malleable Witness Indistinguishable Proofs. We introduce a new notion of *strong non-malleable witness indistinguishability* and show how it can be constructed based on any non-malleable commitment. In this journey, we discuss sequential repetition and the notion of robustness, analogous to [49], for \mathcal{SNMWI} protocols.

Chapter 5 - Proof of the Main Theorem In this chapter, we show how to UC-realize any functionality in any generalized UC-model where there exists a UC-puzzle. Using previous works, this essentially boils down to realizing the Ideal Zero-Knowledge ($\mathsf{IdealZK}$) functionality. The core lemma that we show is that the $\mathsf{IdealZK}$ -functionality can be realized using a UC-puzzle, stand-alone secure SH-OT and a \mathcal{SNMWI} protocol.

Chapter 6 - Applications of General Framework Using our framework, we show essentially, how to obtain all previous known UC-feasibility results with improved round-complexity and computational assumptions. Additionally, we consider a new model of security and show how to achieve feasibility in the model.

Chapter 7 - Lower Bounds In this chapter we establish lower bounds for the Non-Uniform UC-model.

Chapter 8 - Efficient Concurrent Zero-Knowledge As an aside, we investigate how to construct efficient zero-knowledge protocols that are secure under a restricted concurrent setting. More precisely, we construct very efficient concurrent zero-knowledge proofs and arguments, by slightly modifying the definitions of security.

CHAPTER 2

PRELIMINARIES

2.1 Basic Notation

2.1.1 General Notation

We employ the following general notation.

INTEGER AND STRING REPRESENTATION. We denote by \mathbb{N} the set of natural numbers: $0, 1, 2, \dots$. Unless otherwise specified, a natural number is presented in its binary expansion (with no *leading* 0s) whenever given as an input to an algorithm. If $n \in \mathbb{N}$, we denote by 1^n the unary expansion of n (i.e., the concatenation of n 1's). We denote by $\{0, 1\}^n$ the set of n -bit long string, by $\{0, 1\}^*$ the set of binary strings, and by $[n]$ the set $\{1, \dots, n\}$.

We denote the concatenation of two strings x and y by $x|y$ (or more simply by xy). If α is a binary string, then $|\alpha|$ denotes α 's length and $\alpha_1 \dots \alpha_i$ denotes α 's i -bit prefix.

PROBABILISTIC NOTATION. We employ the following probabilistic notation from [39]. We focus on probability distributions $X : S \rightarrow R^+$ over finite sets S .

Probabilistic assignments. If D is a probability distribution and p a predicate, then " $x \xleftarrow{R} D$ " denotes the elementary procedure consisting of choosing an element x at random according to D and returning x , and " $x \xleftarrow{R} D \mid p(x)$ " denotes the operation of choosing x according to D until $p(x)$ is true and then returning x .

Probabilistic experiments. Let p be a predicate and D_1, D_2, \dots probability distributions, then the notation $\Pr[x_1 \xleftarrow{R} D_1; x_2 \xleftarrow{R} D_2; \dots : p(x_1, x_2, \dots)]$ denotes the probability that $p(x_1, x_2, \dots)$ will be true after the ordered execution of the probabilistic assignments $x_1 \xleftarrow{R} D_1; x_2 \xleftarrow{R} D_2; \dots$

New probability distributions. If D_1, D_2, \dots are probability distributions, the notation $\{x \xleftarrow{R} D_1; y \xleftarrow{R} D_2; \dots : (x, y, \dots)\}$ denotes the new probability

distribution over $\{(x, y, \dots)\}$ generated by the ordered execution of the probabilistic assignments $x \xleftarrow{R} D_1, y \xleftarrow{R} D_2, \dots$.

Probability ensembles. Let I be a countable index set. A *probability ensemble indexed by I* is a vector of random variables indexed by I : $X = \{X_i\}_{i \in I}$.

In order to simplify notation, we sometimes abuse of notation and employ the following “short-cut”: Given a probability distribution X , we let X denote the random variable obtained by selecting $x \leftarrow X$ and outputting x .

ALGORITHMS. We employ the following notation for algorithms.

Deterministic algorithms. By an algorithm we mean a Turing machine. We only consider *finite* algorithms, i.e., machines that have some fixed upper-bound on their running-time (and thus always halt). If M is a deterministic algorithm, we denote by $\text{STEPS}_{M(x)}$ the number of computational steps taken by M on input x . We say that an algorithm M has time-complexity $\text{TIME}_M(n) = t(n)$, if $\forall x \in \{0, 1\}^*$ $\text{STEPS}_{M(x)} \leq t(|x|)$. (Note that time complexity is defined as an upper-bound on the running time of M *independently* of its input.)

Probabilistic algorithms. By a probabilistic algorithms we mean a Turing machine that receives an auxiliary random tape as input. If M is a probabilistic algorithm, then for any input x , the notation “ $M_r(x)$ ” denotes the output of the M on input x when receiving r as random tape.

Oracle algorithms. Given two algorithms M, A , we let $M^A(x)$ denote the output of the algorithm M on input x , when given oracle access to A .

Emulation of algorithms. In counting computational steps, we assume that an algorithm M , given the code of a second algorithm A and an input x , can emulate the computation of A on input x with only linear overhead.

NEGLIGIBLE FUNCTIONS. The term “negligible” is used for denoting functions that are asymptotically smaller than the inverse of any fixed polynomial. More precisely, a function $\nu(\cdot)$ from non-negative integers to reals is called *negligible* if for every constant $c > 0$ and all sufficiently large n , it holds that $\nu(n) < n^{-c}$.

2.1.2 Protocol Notation

We assume familiarity with the basic notions of an *Interactive Turing Machine* [38] (ITM for brevity) and a *protocol*. Briefly, an ITM is a Turing Machine with a read-only *input* tape, a read-only *auxiliary input* tape, a read-only *random* tape, a read/write *work-tape*, a read-only communication tape (for receiving messages) a write-only communication tape (for sending messages) and finally an *output* tape. The content of the input (respectively auxiliary input) tape of an ITM A is called *the input* (respectively *auxiliary input*) of A and the content of the output tape of A , upon halting, is called *the output* of A .

A protocol (A, B) is a pair of ITMs that share communication tapes so that the (write-only) send-tape of the first ITM is the (read-only) receive-tape of the second, and vice versa. The computation of such a pair consists of a sequence of rounds 1, 2, In each round only one ITM is active, and the other is idle. A round ends with the active machine either halting—in which case the protocol ends—or by it entering a special *idle* state. The string m written on the communication tape in a round is called the *message sent* by the active machine to the idle machine.

In this thesis we consider protocols (A, B) where both ITMs A, B receive the *same* string as input (but not necessarily as auxiliary input); this input string will be denoted the *common input* of A and B .

We make use of the following notation for protocol executions.

Rounds. In a protocol (A, B) , a round $r \in N$ is denoted an A -round (respectively B -round) if A (respectively B) is active in round r in (A, B) . We say that a protocol has $r(n)$ rounds (or simply is an $r(n)$ -round protocol) if the protocol (A, B) consists of $r(n)$ -rounds of communication between A and B when executed on common input $x \in \{0, 1\}^n$.

Executions, transcripts and views. Let M_A, M_B be vectors of strings $M_A = \{m_A^1, m_A^2, \dots\}$, $M_B = \{m_B^1, m_B^2, \dots\}$ and let $x, r_1, r_2, z_1, r_2 \in \{0, 1\}^*$. We say that the pair $((x, z_1, r_1, M_A), (x, z_2, r_2, M_B))$ is an execution of the protocol (A, B) if, running ITM A on common input x , auxiliary input z_1 and random tape r_1 with ITM B on x , z_2 and r_2 , results in m_A^i being the i 'th message received by A and in m_B^i being the i 'th message received by B . We also

denote the above execution by $\langle A(z_1), B(z_2) \rangle(x)$ for uniformly chosen r_1 and r_2 .

If $((x, z_1, r_1, M_A), (x, z_2, r_2, M_B))$ is an execution, we refer to the tuple (M_A, M_B) as the *transcript* of the execution. We say that (M_A, M_B) is consistent w.r.t A on input x , if there exists auxiliary input z_1 and random tape r_1 such that when A is fed m_A^i as the i 'th message from B results in m_A^i being the i 'th message sent by A .

In an execution $((x, z_1, r_1, M_A), (x, z_2, r_2, M_B)) = (V_A, V_B)$ of the protocol (A, B) , we call V_A the *view of A* (in the execution), and V_B the *view of B* . We let $\text{view}_A[\langle A(z_1), B(z_2) \rangle(x)]$ denote A 's view in the execution $\langle A(z_1), B(z_2) \rangle(x)$ and $\text{view}_B[\langle A(z_1), B(z_2) \rangle(x)]$ B 's view in the same execution. (We occasionally find it convenient referring to an execution of a protocol (A, B) as a *joint view* of (A, B) .)

Outputs of executions and views. If e is an execution of a protocol $\langle A_1, A_2 \rangle$ we denote by $\text{OUTPUT}_i(e)$ the output of A_i , where $i \in \{1, 2\}$. Analogously, if v is the view of A , we denote by $\text{OUTPUT}(v)$ the output of A in v .

Counting ITM steps. Let A be an ITM and $v = (x, z, r, (m_1, m_2, \dots m_k))$. Then by $\text{STEPS}_A(v)$ we denote the number of computational steps taken by A running on common input x , auxiliary input z , random tape r , and letting the i th message received be m_i .

Time Complexity of ITMs. We say that an ITM A has time-complexity $\text{TIME}_A(n) = t(n)$, if for every ITM B , every common input x , every auxiliary inputs z_a, z_b , it holds that $A(x, z_a)$ *always* halts within $t(|x|)$ steps in an interaction with $B(x, z_b)$, regardless of the content of A and B 's random tapes). Note that time complexity is defined as an upper bound on the running time of A *independently* of the content of the messages it receives. In other words, the time complexity of A is the *worst-case* running time of A in *any* interaction.

2.2 Basic Notions

2.2.1 Basic Complexity Classes

We recall the definitions of the basic complexity classes **P**, **NP** and **BPP**.

THE COMPLEXITY CLASS **P**. We start by recalling the definition of the class **P**, i.e., the class of languages that can be decided in (deterministic) polynomial-time.

Definition 1 (Complexity Class **P**). *A language L is recognizable in (deterministic) polynomial-time if there exists a deterministic polynomial-time algorithm M such that $M(x) = 1$ if and only if $x \in L$. **P** is the class of languages recognizable in polynomial time.*

THE COMPLEXITY CLASS **NP**. We recall the class **NP**, i.e., the class of languages for which there exists a proof of membership that can be verified in polynomial-time.

Definition 2 (Complexity Class **NP**). *A language L is in **NP** if there exists a Boolean relation $R_L \subseteq \{0, 1\}^* \times \{0, 1\}^*$ and a polynomial $p(\cdot)$ such that R_L is recognizable in polynomial-time, and $x \in L$ if and only if there exists a string $y \in \{0, 1\}^*$ such that $|y| \leq p(|x|)$ and $(x, y) \in R_L$.*

The relation R_L is called a *witness relation* for L . We say that y is a witness for the membership $x \in L$ if $(x, y) \in R_L$. We will also let $R_L(x)$ denote the set of witnesses for the membership $x \in L$, i.e.,

$$R_L(x) = \{y : (x, y) \in R_L\}$$

We say a language has unique witness (called a unique-witness language), if for every statement in the language, there is only one witness associated with it, i.e. $|R_L(x)| = 1$ for all $x \in L$.

We let **co-NP** denote the complement of the class **NP**, i.e., a language L is in **co-NP** if the complement to L is in **NP**.

THE COMPLEXITY CLASS **BPP**. We recall the class **BPP**, i.e., the class of languages that can be decided in *probabilistic* polynomial-time (with two-sided error).

Definition 3 (Complexity Class **BPP**). *A language L is recognizable in probabilistic polynomial-time if there exists a probabilistic polynomial-time algorithm M such that*

- $\forall x \in L, \Pr [M(x) = 1] \geq 2/3$
- $\forall x \notin L, \Pr [M(x) = 0] \geq 2/3$

BPP is the class of languages recognizable in probabilistic polynomial time.

2.2.2 Indistinguishability

We rely on a generalization of the notion of indistinguishability [37], which considers $T(n)$ -bounded **distinguishers** and require the indistinguishability gap to be smaller than $\frac{1}{\text{poly}(T(n))}$.

Definition 4 (Strong $T(\cdot)$ -indistinguishability[63]). *Let X and Y be countable sets. Two ensembles $\{A_{x,y}\}_{x \in X, y \in Y}$ and $\{B_{x,y}\}_{x \in X, y \in Y}$ are said to be indistinguishable in time $T(\cdot)$ over $x \in X$, if for every probabilistic “distinguishing” algorithm D with running time $T(\cdot)$ in its first input, and every $x \in X, y \in Y$ it holds that:*

$$|\Pr [a \leftarrow A_{x,y} : D(x, y, a) = 1] - \Pr [b \leftarrow B_{x,y} : D(x, y, b) = 1]| < \frac{1}{\text{poly}(T(|x|))}$$

Definition 5 (Computational indistinguishability w.r.t \mathcal{C}). *Let X and Y be countable sets. Two ensembles $\{A_{x,y}\}_{x \in X, y \in Y}$ and $\{B_{x,y}\}_{x \in X, y \in Y}$ are said to be indistinguishable w.r.t \mathcal{C} over $x \in X$, if A, B are $q(\cdot)$ -indistinguishable for every function $q(\cdot) \in \mathcal{C}$.*

Standard indistinguishability can be obtained by setting \mathcal{C} to be the class of all polynomials.

2.2.3 Interactive Proofs and Arguments

Given a pair of interactive Turing machines, P and V , we denote by $\langle P, V \rangle(x)$ the random variable representing the (local) output of V when interacting with machine P on common input x , when the random input to each machine is uniformly and independently chosen.

Definition 6 ($T(\cdot)$ -sound Interactive Proof System). *A pair of interactive machines $\langle P, V \rangle$ is called $T(\cdot)$ -sound interactive proof system for a language L if machine V is polynomial-time and the following two conditions hold :*

- Completeness: *For every $x \in L$, $\Pr[\langle P, V \rangle(x) = 1] = 1$*
- Soundness: *For every $x \notin L$, and every interactive machine B ,*

$$\Pr[\langle B, V \rangle(x) = 1] \leq \frac{1}{T(|x|)}$$

In case that the soundness condition holds only with respect to a $T(n)$ -bounded prover, the pair $\langle P, V \rangle$ is called an $T(\cdot)$ -sound interactive argument.

For any class of functions \mathcal{C} , we say that $\langle P, V \rangle$ is an interactive proofs (interactive argument) w.r.t. \mathcal{C} if for all $T(\cdot) \in \mathcal{C}$ the protocol is a $T(\cdot)$ -sound interactive proof ($T(\cdot)$ -sound interactive argument).

2.2.4 Witness Indistinguishability

An interactive proof is said to be *witness indistinguishable* (\mathcal{WI}) if the verifier's view is "computationally independent" of the witness used by the prover for proving the statement—i.e. the view of the Verifier in the interaction with a prover using witness w_1 or w_2 for two different witnesses are indistinguishable [30].

Definition 7 (Witness-indistinguishability w.r.t \mathcal{C}). *Let $\langle P, V \rangle$ be an interactive proof system for a language $L \in \mathcal{NP}$. We say that $\langle P, V \rangle$ is \mathcal{C} -witness-indistinguishable for R_L , if for every probabilistic interactive machine V^* running in time $q(\cdot)$ for some $q(\cdot) \in \mathcal{C}$ and for every two sequences $\{w_x^1\}_{x \in L}$ and $\{w_x^2\}_{x \in L}$, such that $w_x^1, w_x^2 \in R_L(x)$ for every $x \in L$, the following ensembles are computationally indistinguishable w.r.t \mathcal{C} over $x \in L$:*

- $\{\text{VIEW}_{V^*}[\langle P(w_x^1), V^*(z) \rangle(x)]\}_{x \in L, z \in \{0,1\}^*}$
- $\{\text{VIEW}_{V^*}[\langle P(w_x^1), V^*(z) \rangle(x)]\}_{x \in L, z \in \{0,1\}^*}$

We say that the proof system is perfectly witness indistinguishable (Perfect- \mathcal{WI}) if the corresponding views are identically distributed. Often when using \mathcal{WI} -proofs in protocols, additional properties such as proof of knowledge and special-soundness are desirable. We formally define these below.

PROOFS (ARGUMENTS) OF KNOWLEDGE: Loosely speaking, an interactive proof is a proof of knowledge if the prover convinces the verifier that it *possesses*, or can *feasibly compute*, a witness for the statement proved. The notion of a proof of knowledge is essentially formalized as follows: an interactive proof of $x \in L$ is a proof of knowledge if there exists a probabilistic expected polynomial-time *extractor* machine E , such that for any prover P , E on input the description of P and any statement $x \in L$ readily outputs a valid witness for $x \in L$ if P succeeds in convincing the Verifier that $x \in L$. Formally,

Definition (Proof of knowledge [31]). *Let (P, V) be an interactive proof system for the language L . We say that (P, V) is a proof of knowledge for the witness relation R_L for the language L if there exists an probabilistic expected polynomial-time machine E , called the extractor, and a negligible function $\nu(n)$ such that for every machine P^* , every statement $x \in \{0, 1\}^n$, every random tape $r \in \{0, 1\}^*$ and every auxiliary input $z \in \{0, 1\}^*$,*

$$\Pr[\langle P'_r(z), V \rangle(x) = 1] \leq \Pr\left[E^{P'_r(x, z)}(x) \in R_L(x)\right] + \nu(n)$$

An interactive argument system $\langle P, V \rangle$ is an *argument of knowledge* if the above condition holds w.r.t. probabilistic polynomial-time provers.

SPECIAL-SOUND \mathcal{WI} PROOFS[21]: A 3-round public-coin interactive proof for the language $L \in \mathcal{NP}$ with witness relation R_L is special-sound with respect to R_L , if for any two transcripts (α, β, γ) and $(\alpha', \beta', \gamma')$ such that the initial messages α, α' are the same but the challenges β, β' are different, there is a deterministic procedure to extract the witness from the two transcripts that runs in polynomial time. A 4-round protocol is special sound if a witness can be extracted from any two transcripts $(\tau, \alpha, \beta, \gamma)$ and $(\tau', \alpha', \beta', \gamma')$ such that $\tau = \tau, \alpha = \alpha'$ and $\beta \neq \beta'$.

For simplicity, we use 3-round special-sound proofs in our protocols though our proof works also with 4-round proofs.

Special-sound \mathcal{WI} proofs for languages in \mathbf{NP} can be based on the existence of non-interactive commitment schemes, which in turn can be based on one-way permutations[11, 30]. Assuming only one-way functions, 4-round special-sound \mathcal{WI} proofs for all of \mathbf{NP} exists.

2.2.5 Zero-Knowledge

An interactive proof is said to be zero-knowledge (\mathcal{ZK}) if it yields nothing beyond the validity of the assertion being proved. This is formalized by requiring that the view of every probabilistic polynomial-time adversary V^* interacting with the honest prover P can be simulated by a probabilistic polynomial-time machine S (a.k.a. the simulator). The idea behind this definition is that whatever V^* might have learned from interacting with P , he could have actually learned by himself (by running the simulator S). The notion of \mathcal{ZK} was introduced by Goldwasser, Micali and Rackoff [38]. To make \mathcal{ZK} robust in the context of protocol composition, Goldreich and Oren [41] suggested to augment the definition so that the above requirement holds also with respect to all $z \in \{0,1\}^n$, where both V^* and S are allowed to obtain z as auxiliary input. The verifier's view of an interaction consists of the common input x , followed by its random tape and the sequence of prover messages the verifier receives during the interaction. The same definition generalizes when we consider efficient adversaries to be an arbitrary complexity class \mathcal{C} .

Definition 8 (Zero-Knowledge w.r.t. \mathcal{C}). *Let $\langle P, V \rangle$ be an interactive proof system. We say that $\langle P, V \rangle$ is zero-knowledge w.r.t. \mathcal{C} , if for every probabilistic interactive machine V^* running in time $q(\cdot)$ for some $q(\cdot) \in \mathcal{C}$, there exists a probabilistic algorithm S_q that runs in time polynomial in $(q(|x|), |x|)$ such that the following ensembles are indistinguishable over $x \in L$ w.r.t \mathcal{C} :*

- $\{\text{VIEW}_{V^*}[\langle P, V^*(z) \rangle(x)]\}_{x \in L, z \in \{0,1\}^*}$
- $\{S(x, z)\}_{x \in L, z \in \{0,1\}^*}$

A stronger variant of zero-knowledge is one in which the output of the simulator is statistically close to the verifier’s view of real interactions. We focus on argument systems, in which the soundness property is only guaranteed to hold with respect to polynomial time provers.

2.2.6 Commitment Schemes

Commitment schemes are the digital equivalent of physical envelopes. They enable a first party, referred to as the *sender*, to commit itself to a value while keeping it secret from a second party, the *receiver*; this property is called **hiding**. Furthermore, the commitment is **binding**, and thus in a later stage when the commitment is opened, it is guaranteed that the “opening” can yield only a single value determined in the committing phase. The opening phase traditionally consists of the sender simply sending the receiver the value v it committed to, as well as the random coins r it used. The receiver accepts the opening to v if the messages it received during the committing phase are produced by running the honest sender algorithm on input v and the random tape r .

Commitment schemes come in two different flavors, **perfectly-binding** and **perfectly-hiding**.

PERFECT-BINDING. In a perfectly-binding commitments, the binding property holds against unbounded adversaries, while the hiding property only holds against computationally bounded adversaries. Loosely speaking, the perfectly-binding property asserts that the transcript of the interaction fully determines the value committed to by the sender. The computational-hiding property guarantees that commitments to any two different values are computationally indistinguishable; actually, in most applications (and in particular for the construction of zero-knowledge proofs) we require that the indistinguishability of commitments holds even when the distinguisher receives an auxiliary “advice” string (this is sometimes called non-uniform computational hiding).

For simplicity, we present a definition of a commitment scheme for enabling a sender to commit to a *single* bit.

Definition 9 (Perfectly-binding commitment). *A perfectly-binding bit commitment scheme is a triple $(\mathsf{C}, \mathsf{O}, \mathsf{V})$ consisting of a commitment function C , an opening function O , and a verification function V such that the following properties hold:*

ment scheme is a pair of probabilistic polynomial-time interactive machines (S, R) satisfying the following properties:

- Perfect Binding: *For every malicious sender S' and auxiliary input z , it holds that $\text{view}_R[\langle S'(z), R \rangle(1^n)]$ is consistent w.r.t at most one input b for the honest Sender S .*
- Computational Hiding: *For every probabilistic polynomial-time ITM R' the following ensembles are computationally indistinguishable over $n \in \mathbb{N}$*
 - $\left\{ \text{view}_{R'}[\langle S(0), R'(z) \rangle(1^n)] \right\}_{n \in N, z \in \{0,1\}^*}$
 - $\left\{ \text{view}_{R'}[\langle S(1), R'(z) \rangle(1^n)] \right\}_{n \in N, z \in \{0,1\}^*}$

Above, the variable n is a parameter determining the security of the commitment scheme.

PERFECT-HIDING. In perfectly-hiding commitments, the hiding property holds against unbounded adversaries, while the binding property only holds against computationally bounded adversaries. Loosely speaking, the perfectly-hiding property asserts that commitments to any two different values are identically distributed. The computational-binding property guarantees that no polynomial time adversary algorithm is able to construct a commitment that can be opened in two different ways; again, for our applications, we actually require that the binding property holds also when providing the adversary with an “advice” string (this property is sometimes called non-uniform computational binding). We omit a formal definition of perfectly-hiding commitments and refer the reader to [31].

STATISTICAL BINDING/HIDING. We mention that it is often convenient to relax the perfectly-binding or the perfectly-hiding properties to only statistical binding or hiding. Loosely speaking, the **statistical-binding** property asserts that with overwhelming probability (instead of probability 1) over the coin-tosses of the receiver, the transcript of the interaction fully determines the committed value. The **statistical-hiding** property asserts that commitments to any two different values are statistically close (i.e., have negligible statistical difference, instead of being identically distributed).

EXISTENCE OF COMMITMENT SCHEMES. Non-interactive perfectly-binding commitment schemes can be constructed using any 1–1 one-way function (see Section 4.4.1 of [31]). Allowing some minimal interaction (in which the receiver first sends a single message), statistically-binding commitment schemes can be obtained from any one-way function [59, 43]. Perfectly-hiding commitment schemes can be constructed from any one-way permutation [60]. However, *constant-round* schemes are only known to exist under stronger assumptions; specifically, assuming the existence of a collection of certified clawfree permutations [32] (see also [31], Section 4.8.2.3). Constant-round statistically-hiding commitments can be constructed under the potentially weaker assumption of collision-resistant hash functions [23, 42].

CHAPTER 3

GENERAL FRAMEWORK FOR SECURE COMPUTATION

In this section, we present a generalization of the UC notion of security introduced by Canetti [12]. We first briefly recall the basic definition of secure computation in the UC model [36, 7, 58, 12], and then provide a brief description of the generalized model. We here focus only on *static* adversaries—i.e., players are corrupted upon invocation only.

3.1 Traditional UC

ENVIRONMENT. The model of execution includes a special entity called the UC-environment (or environment) Z . The environment “manages” the whole execution: it invokes all the parties at the beginning of the execution, generates all inputs and reads all outputs, and finally produces an output for the whole concurrent execution. Intuitively, the environment models the “larger world” in which the concurrent execution takes place (*e.g.*, for a distributed computing task over the Internet, the environment models all the other activities occurring on the Internet at the same time).

ADVERSARIAL BEHAVIOR. The model of execution also includes a special entity called the adversary, that represents adversarial activities that are directly aimed at the protocol execution under consideration. We consider a *static* adversary; that is, whenever a party is invoked, the adversary is notified by the environment whether the party is corrupted or not¹. When a party is corrupted, it shares all its tapes with the adversary and follows the instructions from the adversary for all its future actions.

While honest parties only communicate with the environment through the input/output of the functions they compute, the adversary is also able to exchange messages with the environment in an arbitrary way through out the computation².

¹In contrast, an adaptive adversary may corrupt a party during its computation, and as a function of what it sees.

²Through its interaction with the environment, the adversary is also able to influence the inputs to honest parties indirectly.

Furthermore, the adversary controls the scheduling of the delivery of all messages exchanged between parties (messages sent by the environment is delivered directly). Technically, this is modelled by letting the adversary read the outgoing message tapes of all parties and decide whether or not and when (if at all) to deliver the message to the recipient, therefore the communication is asynchronous and lossy. However, the adversary cannot insert messages and claim arbitrary sender identity. In other words, the communication is authenticated.

PROTOCOL EXECUTION. The *execution of a protocol π with the environment Z , adversary A and trusted party \mathcal{G}* proceeds as follows. The environment is the first entity activated in the execution, who then activates the adversary, and invokes other honest parties. At the time an honest party is invoked, the environment assigns it a unique identifier, and inquires the adversary whether it wants to corrupt the party or not. To start an execution of the protocol π , the environment initiates a *protocol execution session*, identified by a session identifier sid , and activates all the participants in that session. An honest party activated starts executing the protocol π thereafter and has access to the trusted party \mathcal{G} . We remark that in the UC model, the environment only initiates one protocol execution session.

Invoking parties. The environment invokes an honest party by passing input (*invoke*, P_i) to it. P_i is the globally unique identity for the party, and is picked dynamically by the environment at the time it is invoked. Immediately after that, the environment notifies the adversary of the invocation of P_i by sending the message (*invoke*, P_i) to it, who can then choose to corrupt the party by replying (*corrupt*, P_i). Note that here as the adversary is static, parties are corrupted only when they are “born” (invoked).

Session initiation. To start an execution of protocol π , the environment selects a subset U of parties that has been invoked so far. For each party $P_i \in U$, the environment activates P_i by sending a start-session message (*start-session*, P_i , sid , $c_{i,sid}$, $x_{i,sid}$) to it, where sid is a session id that identifies this execution. We remark that in the UC model, the environment starts only one session, and hence all the parties activated have the same session id.

Honest party execution. An honest party P_i , upon receiving (*start-session*, P_i ,

sid , $c_{i,sid}$, $x_{i,sid}$), starts executing its code $c_{i,sid}$ input $x_{i,sid}$. During the execution,

- the environment can read P_i 's output tape and at any time may pass additional inputs to P_i ;
- according to its code, P_i can send messages (delivered by the adversary) to other parties in the session, in the format $(P_i, P_j, s, \text{content})^3$, where P_j is the identity of the receiver;
- according to its code, P_i can send input to the trusted party in the format $(P_i, \mathcal{F}, s, \text{input})$.

Adversary execution. After activation, the adversary may perform one of the following activities at any time during the execution.

- The adversary can read the outgoing communication tapes of all honest parties and decides to deliver some of the messages.
- A can exchange arbitrary messages with the environment.
- The adversary can read the inputs, outputs, incoming messages of a corrupted party, and instruct the corrupted party for any action.

Output. The environment outputs a final result for the whole execution in the end.

In the execution of protocol π with security parameter $n \in N$, environment Z , adversary A and trusted party \mathcal{G} , we define $\text{EXEC}_{\pi,A,Z}^{\mathcal{G}}(n)$ to be the random variable describing the output of the environment Z , resulting from the execution of the above procedure.

Let \mathcal{F} be an ideal functionality; we denote by π_{ideal} the protocol accessing \mathcal{F} , called as the ideal protocol. In π_{ideal} parties simply interacts with \mathcal{F} with their private inputs, and receives their corresponding outputs from the functionality at the end of the computation. Then the ideal model execution of the functionality \mathcal{F} is just the execution of the ideal protocol π_{ideal} with environment Z , adversary A' and trusted party \mathcal{F} . The output of the execution is thus $\text{EXEC}_{\pi_{\text{ideal}},A',Z}^{\mathcal{F}}(n)$. On the

³The session id in the messages enables the receiver to correctly de-multiplexing a message to its corresponding session, even though the receiver may involve in multiple sessions simultaneously.

other hand, the real model execution does not require the aid of any trusted party. Let π be a multi-party protocol implementing \mathcal{F} . Then, the real model execution of π is the execution of π with security parameter n , environment Z and adversary A , whose output is the random variable $\text{EXEC}_{\pi,A,Z}(n)$. Additionally, the \mathcal{G} -Hybrid model execution of a protocol π is the execution of π with security parameter n , environment Z and adversary A and ideal functionality \mathcal{G} .

SECURITY AS EMULATION OF A REAL MODEL EXECUTION IN THE IDEAL MODEL. Loosely speaking, a protocol securely realizes an ideal functionality if it securely emulates the ideal protocol π_{ideal} . This is formulated by saying that for every adversary A in the real model, there exists an adversary A' (a.k.a. *simulator*) in the ideal model, such that no environment Z can tell apart if it is interacting with A and parties running the protocol, or A' and parties running the ideal protocol π_{ideal} .

Definition 10. (UC security) *Let \mathcal{F} and π_{ideal} be defined as above, π be a multi-party protocol in the \mathcal{G} -hybrid model. The protocol π is said to realize \mathcal{F} with UC security in \mathcal{G} -hybrid model, if for every uniform PPT adversary A , there exists a uniform PPT simulator A' , such that, for every non-uniform PPT environment Z , the following two ensembles are indistinguishable.*

$$\{\text{EXEC}_{\pi,A,Z}^{\mathcal{G}}(n)\}_{n \in N} \approx \{\text{EXEC}_{\pi_{\text{ideal}},A',Z}^{\mathcal{F}}(n)\}_{n \in N}$$

MULTI-SESSION EXTENSION OF IDEAL FUNCTIONALITIES Note that the UC model only considers a single session of the protocol execution. (The environment is only allowed to open one session). To consider multiple concurrent executions, we focus on the multi-session extension of ideal functionalities [12, 19]. More specifically, let $\hat{\mathcal{F}}$ be the multi-session extension of \mathcal{F} . $\hat{\mathcal{F}}$ runs multiple copies of \mathcal{F} , where each copy will be identified by a special “sub-session identifier”. Every k parties, trying access \mathcal{F} together, share a sub-session identifier, $ssid$. To compute, each party simply sends its private input together with $ssid$ to $\hat{\mathcal{F}}$. $\hat{\mathcal{F}}$ upon receiving all the inputs, activates the appropriate copy of \mathcal{F} identified by $ssid$ (running within $\hat{\mathcal{F}}$), and forwards the incoming messages to that copy. (If no such copy of \mathcal{F} exists then a new copy is invoked and is given that $ssid$.) Outputs generated by the copies of \mathcal{F} are returned to corresponding parties by $\hat{\mathcal{F}}$.

3.2 A Generalized Version of UC

In the UC model, the environment is modeled as a non-uniform \mathcal{PPT} machine and the ideal-model adversary (or simulator) as a (uniform) \mathcal{PPT} machines. We consider a generalized version (in analogy with [63, 71]) where we allow them to be in arbitrary complexity classes. Note, however, that the adversary is still \mathcal{PPT} . Additionally, we “strengthen” the definition by allowing the environment to output a bit string (instead of a single bit) at the end of an execution. In the traditional UC definition, it is w.l.o.g. enough for the environment to output a single bit [12]; in our generalized version this no longer holds and we are thus forced to directly consider the more stringent version.

We represent a generalized UC model by a 2-tuple $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$, where \mathcal{C}_{env} and \mathcal{C}_{sim} are respectively the classes of machines the environment and the simulator of the general model belong to. We consider only classes, \mathcal{C}_{env} and \mathcal{C}_{sim} , that are closed under probabilistic polynomial time computation.

Definition 11 $((\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})\text{-UC security})$. Let \mathcal{F} and π_{ideal} be, as defined above, and π be a multi-party protocol. The protocol π is said to realize \mathcal{F} with $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})\text{-UC security}$, if for every \mathcal{PPT} machine A , there exists a machine $A' \in \mathcal{C}_{\text{sim}}$, such that, for every $Z \in \mathcal{C}_{\text{env}}$, the following two ensembles are indistinguishable w.r.t \mathcal{C}_{sim} .

$$\{\text{EXEC}_{\pi, A, Z}(n)\}_{n \in N} \approx \{\text{EXEC}_{\pi_{\text{ideal}}, A', Z}^{\mathcal{F}}(n)\}_{n \in N}$$

Using the above notation, traditional UC is equivalent to $(\text{n.u.}\mathcal{PPT}, \mathcal{PPT})\text{-UC-security}$. We let QPS-UC denote $(\text{n.u.}\mathcal{PPT}, \mathcal{PQT})\text{-UC-security}$ ⁴ (where \mathcal{PQT} denotes probabilistic quasi-polynomial time algorithms), and Non-uniform UC denote $(\mathcal{PPT}, \text{n.u.}\mathcal{PPT})\text{-UC-security}$.

3.3 Main Result

By relying on previous results [64, 66, 52, 19, 34] the construction of a UC secure protocol for realizing any multi-party functionality reduces to the task of

⁴We mentioned that this is stronger than the notion of QPS security of [63, 71, 6] which only consider indistinguishability w.r.t \mathcal{PPT} ; we, in analogy with the notion of *strong QPS* of [63] require indistinguishability to hold also w.r.t \mathcal{PQT} .

constructing a zero-knowledge protocol that satisfies the following two properties:⁵

UC simulation: For every adversary A receiving honest proofs of statements x using witness w , where (x, w) are selected by an “environment” \mathcal{Z} , there exists a simulator S (which only get the statements x) such that no \mathcal{Z} can distinguish if it is talking to A or S .

Concurrent simulation-soundness: Even an adversary that receives an unbounded number of concurrently *simulated* proofs, of statements selected by the environment \mathcal{Z} , still is not able to prove any false statements.

We propose a framework for constructing protocols to securely realize any functionality in a general model $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$. First, we introduce the notion of a *UC-puzzle*. Then, we show how to use any UC-puzzle, stand-alone secure SH-OT and a \mathcal{SNMWI} protocol to construct a zero knowledge protocol $\langle P, V \rangle$ that is UC-simulatable and concurrently simulation-sound. More precisely, assuming the existence of stand-alone secure SH-OT protocol, a \mathcal{SNMWI} protocol and a UC-puzzle we provide a general procedure that compiles any functionality \mathcal{F} into a protocol π that securely realizes a multi-session extension of the ideal-functionality \mathcal{F} . Informally, a UC-*puzzle* is a protocol between two parties, a sender and a receiver that *cannot* be “solved” by an adversary acting as a receiver in the real model, but can be solved by a machine while simulating the interactions. For a model $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$, let $cl(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$ denote the complexity class that includes all computations by \mathcal{PPT} oracle Turing machines M with oracle access to \mathcal{C}_{env} and \mathcal{C}_{sim} . Before, providing a formal definition of a UC-puzzle, we state our main theorem.

Theorem 3 (Main Theorem). *Assume the existence of a t_P -round $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$ -secure UC-puzzle in a \mathcal{G} -hybrid model, t_{WI} -round \mathcal{SNMWI} protocol secure w.r.t $cl(\mathcal{C}_{\text{sim}}, \mathcal{C}_{\text{env}})$ and t_{OT} -round semi-honest oblivious transfer protocol secure w.r.t $cl(\mathcal{C}_{\text{sim}}, \mathcal{C}_{\text{env}})$. Then, for every “well-formed” functionality \mathcal{F} , there exists a $O(t_P + t_{WI} + t_{OT})$ -round protocol Π in the \mathcal{G} -hybrid model that realizes $\hat{\mathcal{F}}$ with $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$ -UC-security.*

⁵Formally, this can be modelled as implementing a particular “zero-knowledge” proof of membership functionality.

3.4 UC-puzzles

Roughly speaking, a UC puzzle is a protocol $\langle S, R \rangle$ between two players—a *sender* and a *receiver*—and a \mathcal{PPT} -computable relation \mathcal{R} , such that the following two properties hold:

Soundness: No efficient receiver R^* can successfully complete an interaction with S and also obtain a “trapdoor” y , such that $\mathcal{R}(\text{TRANS}, y) = 1$, where TRANS is the transcript of the interaction.

Statistical UC-simulation: For every efficient adversary A , participating in a polynomial number of concurrent executions with receivers R (i.e., A is acting as a puzzle sender in all these executions) and at the same time communicating with an environment Z , there exists a simulator S that is able to *statistically* simulate the view of A for Z , while at the same time outputting trapdoors to all successfully completed puzzles.

Formally, let $n \in N$ be a security parameter and $\langle S, R \rangle$ be a protocol between two parties, the sender S and the receiver R . We consider a **concurrent puzzle execution** for an adversary A . In a **concurrent puzzle execution**, A exchanges messages with a puzzle-environment $Z \in \mathcal{C}_{\text{env}}$ and participates as a sender concurrently in $m = \text{poly}(n)$ puzzles with honest receivers R_1, \dots, R_m . At the onset of a concurrent execution, Z outputs a session-identifier sid that all receivers in the concurrent puzzle execution receive as input. Thereafter, the puzzle-environment is allowed to exchange messages only with the adversary A . We compare a *real* and an *ideal* execution.

REAL EXECUTION. In the real execution, the adversary A on input 1^n , interacts with a puzzle-environment $Z \in \mathcal{C}_{\text{env}}$ and participates as a sender in m interactions using $\langle S, R \rangle$ with honest receivers that receive input sid (decided by Z). The adversary A is allowed to exchange arbitrary messages with environment Z when participating in puzzle interactions with the receivers as a sender. We assume without loss of generality that, after every puzzle-interaction, A honestly sends TRANS to Z , where TRANS is the puzzle-transcript. Finally, Z outputs a string in $\{0, 1\}^*$. We denote this by $\text{REAL}_{A,Z}(n)$.

IDEAL EXECUTION. Consider $A' \in \mathcal{C}_{\text{sim}}$ in the ideal-model that has a special output-tape (not accessible by Z). In the ideal execution, A' on input 1^n interacts with puzzle-environment Z . We denote the output of Z at the end of the execution by $\text{IDEAL}_{A',Z}(n)$.

Definition 12 (UC-Puzzle). *A pair $(\langle S, R \rangle, \mathcal{R})$ is a $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$ -secure UC-puzzle for a polynomial time computable relation \mathcal{R} and model $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$, if the following conditions hold.*

- **Soundness:** *For every malicious PPT receiver A , there exists a negligible function $\nu(\cdot)$ such that the probability that A , after an execution with R on common input 1^n , outputs y such that $y \in \mathcal{R}(\text{TRANS})$ where TRANS is the transcript of the messages exchanged in the interaction, is at most $\nu(n)$.*
- **Statistical Simulatability:** *For every adversary $A \in \mathcal{C}_{\text{adv}}$ participating in a concurrent puzzle execution, there is a simulator $A' \in \mathcal{C}_{\text{sim}}$ such that for all puzzle-environments $Z \in \mathcal{C}_{\text{env}}$, the ensembles $\{\text{REAL}_{A,Z}(n)\}_{n \in N}$ and $\{\text{IDEAL}_{A',Z}(n)\}_{n \in N}$ are statistically close over $n \in N$ and whenever A' sends a message of the form TRANS to Z , it outputs y in its special output tape such that $y \in \mathcal{R}(\text{TRANS})$.*

In other words, we require that no adversarial receiver can complete a puzzle with a trapdoor, but there exists a simulator (which does not rewind the environment it runs in) that can generate *statistically indistinguishable* puzzle transcripts joint with trapdoors. We highlight that the puzzle protocol $\langle S, R \rangle$ may make use of trusted set-up.

As we show, UC-puzzles in a trusted set-up model \mathcal{T} are *sufficient* for achieving UC secure computation with set-up \mathcal{T} . This result also holds in generalized versions of the UC framework (which allow us to consider QPS and non-uniform UC security).

CHAPTER 4
**STRONG NON-MALLEABLE WITNESS INDISTINGUISHABLE
 PROOFS**

4.1 Definition

We start by defining the notion of strong non-malleable witness-indistinguishability (\mathcal{SNMWI}) only for languages with *unique* witnesses; we next extend it to general NP-languages. Let R_L be the canonical witness relation for some language L with unique witnesses. Consider a, so-called, tag-based argument system for L —i.e., the prover and the verifier receive a “tag” as an additional common input, besides the statement x . \mathcal{SNMWI} considers a man-in-the-middle execution of the protocol $\langle P_s, V_s \rangle$, in which the adversary A simultaneously participates in two interactions of $\langle P_s, V_s \rangle$, one left and one right interaction. In the left interaction, the adversary A , on auxiliary input z , receives a proof of statement x from P_s on private input y such that $y \in R_L(x)$, using a fixed tag id . In the right interaction, A adaptively chooses a statement \tilde{x} and tag $\tilde{\mathsf{id}}$ and attempts to provide a proof to V_s . Let \tilde{y} denote the witness associated with \tilde{x} , unless either of the following happens (a) A fails in the right interaction or (b) $\mathsf{id} = \tilde{\mathsf{id}}$; in this case \tilde{y} is set to \perp . Let $\mathsf{mim}_{\langle P_s, V_s \rangle}^A(x, y, z, \mathsf{id})$ denote the random variable that describes the witness \tilde{y} combined with the view of A in the above man-in-the-middle experiment.

Definition 13 (Strongly Non-Malleable WI). *We say that $\langle P_s, V_s \rangle$ is strongly non-malleable witness-indistinguishable for R_L if for every non-uniform PPT man-in-the-middle adversary A , every $\mathsf{id} \in \{0,1\}^*$ and every two sequences of input distributions $\{D_n^1\}_{n \in N}$ and $\{D_n^2\}_{n \in N}$, the following holds: if $\{(x, y, z) \leftarrow D_n^1 : (x, z)\}_{n \in N}$ and $\{(x, y, z) \leftarrow D_n^2 : (x, z)\}_{n \in N}$ are computationally indistinguishable, so are the following ensembles:*

$$\begin{aligned} & \{(x, y, z) \leftarrow D_n^1 : \mathsf{mim}_{\langle P_s, V_s \rangle}^A(x, y, z, \mathsf{id})\}_{n \in N, \mathsf{id} \in \{0,1\}^*} \\ & \{(x, y, z) \leftarrow D_n^2 : \mathsf{mim}_{\langle P_s, V_s \rangle}^A(x, y, z, \mathsf{id})\}_{n \in N, \mathsf{id} \in \{0,1\}^*} \end{aligned}$$

When considering a general language, mim is not well defined, as \tilde{y} is not uniquely determined. In this case, whenever A chooses a statement \tilde{x} that does

not have a unique witness, simply let \tilde{y} output \perp . Furthermore, we only require that the above condition holds for well-behaved adversaries A , where A is said to be well-behaved, if, except with negligible probability A only chooses statements \tilde{x} with unique witnesses.

We remark that our notion of \mathcal{SNMWI} is similar in spirit to the notion of *non-malleable witness indistinguishability* (\mathcal{NMWI}) recently introduced by Ostrovsky, Persiano, and Visconti [61]. Both notions consider a flavor of non-malleability for \mathcal{WI} argument systems and (informally) require that the “witness in the right interaction” is “independent” of that of the left interaction. The main difference between the notions is that whereas the notion of \mathcal{NMWI} “only” requires this to hold when varying the witness used in the left interaction, but keeping the statement fixed, we also require indistinguishability whenever the statements in the left interactions are indistinguishable (just as the notion of *strong witness indistinguishability* [31]). As such, our notion is interesting—in fact, the most interesting—also when considering statements with unique witnesses, whereas \mathcal{NMWI} vacuously holds. In essence, the notion of \mathcal{NMWI} extends the notion of plain WI to the man-in-the-middle setting, whereas \mathcal{SNMWI} extends strong WI.

CONSTRUCTING \mathcal{SNMWI} ARGUMENT OF KNOWLEDGE. We provide two constructions of \mathcal{SNMWI} protocols.

- For the first construction, we show that \mathcal{SNMWI} is a relaxation of the notion of *simulation-extractability* [25, 67]. The notion of simulation-extractability, roughly speaking, requires the existence of an efficient machine $SIMEXT$, called a simulator extractor, that can *statistically* simulate the view of a man-in-the-middle adversary A (that participates in two interactions, a left one as a verifier and a right one as a prover) while at the same time extracting all the witnesses used by A .

It can be seen that this notion directly implies \mathcal{SNMWI} . In fact, in the case of languages with unique witnesses, a simulator-extractor can, given only the statement on the left, reconstruct both the view and the witness used by the adversary on the right (note that we here rely on the fact that simulation is statistically secure, or argue that the witness is correctly distributed). As such, we directly get that the $O(1)$ -round construction of [67] is a \mathcal{SNMWI} argument of knowledge; this construction relies on collision-resistant hash-

functions.

Then, relying on a constant-round construction of a simulation-extractable protocol based on collision-resistant hash function in [66] we obtain a $O(1)$ -round \mathcal{SNMWI} protocol. We provide this construction merely to show the relationship of \mathcal{SNMWI} to simulation-extractability. In our actual construction, we will rely on the \mathcal{SNMWI} argument constructed from a non-malleable commitment.

- To minimize assumptions, we provide a new construction of \mathcal{SNMWI} arguments of knowledge based on $O(1)$ -robust non-malleable commitment schemes. Then by relying on the construction of such non-malleable commitment scheme from [50, 40], we get the existence $O(1)$ -round \mathcal{SNMWI} argument of knowledge from any one-way function.

4.2 \mathcal{SNMWI} from Simulation-Extractability

The notion of *simulation extractability* was introduced in the work of Dolev, Dwork and Naor [25]; we here rely on a formalization due to Pass and Rosen [66]: Intuitively, a protocol is said to be simulation extractable if for any man-in-the-middle adversary A , there exists a simulator-extractor that can *statistically* simulate the views of both the left and the right interactions for A , while outputting a witness for the statement proved by A in the right interaction. We denote by $\text{view}_A(x, y, z, \text{id})$ the view of A in a real man-in-the-middle execution with inputs x, y, z and identity id in the left interaction. Let $\mathcal{S}(x, z, \text{id})$ be the output of the simulator, which consists of the simulated view of A and the witness it uses; we denote the former by $\mathcal{S}_1(x, z, \text{id})$ and the latter by $\mathcal{S}_2(x, z, \text{id})$.

As shown in [67], Simulation-extractability is sufficient for an interactive-proof to be non-malleable \mathcal{ZK} . Below, we show that any protocol that it is also sufficient for \mathcal{SNMWI} .

First, we recall the definition of simulation-extractability from [66]. Given a function $t = t(n)$ we use the notation $\{\cdot\}_{n,x,y,z,\text{id}}$ as shorthand for $\{\cdot\}_{n \in N, x \in L, y \in R_L \cap \{0,1\}^n, z \in \{0,1\}^*, \text{id} \in \{0,1\}^{t(n)}}.$

Definition 14 (Simulation-extractable protocol). *A family $\{\langle P_{\text{id}}, V_{\text{id}} \rangle\}_{n \in N, \text{id} \in \{0,1\}^*}$*

of interactive proofs is said to be simulation extractable with tags of length $t=t(n)$ if for any man-in-the-middle adversary A , there exists a probabilistic expected polytime machine \mathcal{S} such that:

1. The probability ensembles $\{\mathcal{S}_1(x, z, \text{id})\}_{n,x,y,z,\text{id}}$ and $\{\text{view}_A(x, y, z, \text{id})\}_{n,x,y,z,\text{id}}$ are statistically close.
2. Let $n \in N$, $x \in L \cap \{0, 1\}^n$, $z \in \{0, 1\}^*$, $\text{id} \in \{0, 1\}^{t(n)}$ and let (view, w) denote the output of $\mathcal{S}(x, z, \text{id})$ (on input some random tape). Let \tilde{x} be the right-execution statement appearing in view and let $\tilde{\text{id}}$ denote the right-execution identity. Then, if the right-execution in view is accepting AND $\text{id} \neq \tilde{\text{id}}$, then $R_L(\tilde{x}, w) = 1$.

We now proceed to show that all simulation-extractable protocols satisfy \mathcal{SNMWI} .

Lemma 1. Let $\{\langle P_{\text{id}}, V_{\text{id}} \rangle\}_{n \in N, \text{id} \in \{0, 1\}^{t(n)}}$ be a simulation-extractable protocol. Then, $\{\langle P_{\text{id}}, V_{\text{id}} \rangle\}_{n \in N, \text{id} \in \{0, 1\}^{t(n)}}$ is \mathcal{SNMWI} .

Proof. Assume for contradiction that $\{\langle P_{\text{id}}, V_{\text{id}} \rangle\}_{n \in N, \text{id} \in \{0, 1\}^{t(n)}}$ is not \mathcal{SNMWI} . Then, there exists a well-behaved adversary A , a distinguisher \mathcal{D} , two sequences of identifies $\{\text{id}_n\}_{n \in N}$ and $\{\text{id}'_n\}_{n \in N}$, two sequences of distributions D_n^1 and D_n^2 such that $\{(x, y, z) \leftarrow D_n^1 : (x, z)\}_{n \in N}$ and $\{(x, y, z) \leftarrow D_n^2 : (x, z)\}_{n \in N}$ are computationally-indistinguishable and a polynomial $p(\cdot)$ such that for infinitely many n ,

$$\left| \Pr \left[\{(x, y, z) \leftarrow D_n^1 : \mathcal{D}(\text{mim}_{\langle P_s, V_s \rangle}^A(x, y, z, \text{id}_n)) = 1\} \right] - \Pr \left[\{(x, y, z) \leftarrow D_n^2 : \mathcal{D}(\text{mim}_{\langle P_s, V_s \rangle}^A(x, y, z, \text{id}'_n)) = 1\} \right] \right| \geq \frac{1}{p(n)}$$

Using the adversary A and the distinguisher \mathcal{D} , we construct a distinguisher \mathcal{D}' that distinguishes $\{(x, y, z) \leftarrow D_n^1 : (x, z)\}$ from $\{(x, y, z) \leftarrow D_n^2 : (x, z)\}$ and thus arrive at a contradiction.

Since $\{\langle P_{\text{id}}, V_{\text{id}} \rangle\}_{n \in N, \text{id} \in \{0, 1\}^{t(n)}}$ is simulation-extractable, there exists a simulator S such that the view in $\text{mim}_{\langle P_s, V_s \rangle}^A(x, y, z, \text{id})$ and $S_1(x, z, \text{id})$ are statistically close. Further, the witness output by the simulator, i.e. $S_2(x, z, \text{id})$ is identical to the witness in $\text{mim}_{\langle P_s, V_s \rangle}^A(x, y, z, \text{id})$ since the adversary is well-behaved and hence, the

statement has an unique witness. Thus $\text{mim}_{\langle P_s, V_s \rangle}^A(x, y, z, \text{id})$ and $S(x, z, \text{id})$ are statistically-close. This means that there exists a negligible function $\nu(\cdot)$ such that

$$\left| \Pr [\{(\text{view}, w) \leftarrow \text{mim}_{\langle P_s, V_s \rangle}^A(x, y, z, \text{id}_n) : \mathcal{D}(\text{view}, w) = 1\}] - \Pr [\{(\text{view}, w) \leftarrow S(x, z, \text{id}'_n) : \mathcal{D}(\text{view}, w) = 1\}] \right| \leq \nu(n)$$

The distinguisher \mathcal{D}' proceeds as follows. On input (x, z) internally incorporates S with inputs (x, z, id_n) . Upon receiving the output (view, w) from S , \mathcal{D}' internally incorporates \mathcal{D} with input (view, w) and outputs what \mathcal{D} outputs. It follows from construction that the experiments $\{(\text{view}, w) \leftarrow S(x, z, \text{id}) : \mathcal{D}(\text{view}, w) = 1\}$ and $\{\mathcal{D}'(x, z)\}$ are identical. Therefore, for $i \in \{1, 2\}$ we have that

$$\left| \Pr [\{(x, y, z) \leftarrow D_n^i : (\text{view}, w) \leftarrow \text{mim}_{\langle P_s, V_s \rangle}^A(x, y, z, \text{id}_n) : \mathcal{D}(\text{view}, w) = 1\}] - \Pr [\{(x, y, z) \leftarrow D_n^i : \mathcal{D}'(x, z) = 1\}] \right| \leq \nu(n)$$

Thus,

$$\left| \Pr [\{(x, y, z) \leftarrow D_n^1 : \mathcal{D}'(x, z) = 1\}] - \Pr [\{(x, y, z) \leftarrow D_n^2 : \mathcal{D}'(x, z) = 1\}] \right| \geq \frac{1}{p(n)} - 2\nu(n)$$

and we arrive at a contradiction since by hypothesis $\{(x, y, z) \leftarrow D_n^1 : x, z\}_{n \in N}$ and $\{(x, y, z) \leftarrow D_n^2 : x, z\}_{n \in N}$ are computationally-indistinguishable.

□

As the protocol of [67] is both simulation-extractable and an argument of knowledge, we conclude that the existence of collision-resistant hash functions implies $O(1)$ -round \mathcal{SNMWI} argument of knowledge.

4.3 \mathcal{SNMWI} from any Non-Malleable Commitment

In this section we provide a construction of \mathcal{SNMWI} argument of knowledge systems based on any $O(1)$ -robust non-malleable commitment scheme (See the definition below). Then by relying on the construction of non-malleable commitments from [50, 40], we get the existence $O(1)$ -round \mathcal{SNMWI} from any one-way

function. (If we had relied on the original construction of [25] (see also [51]) we would instead have obtained a $\log n$ -round construction.) Below we first provide the formal definition of non-malleable commitments and then proceed to the construction of \mathcal{SNMWI} arguments of knowledge.

4.3.1 Non-malleable commitment schemes

Let $\langle C, R \rangle$ be a tag-based commitment scheme, and let $n \in N$ be a security parameter. Consider a man-in-the-middle adversary A that participates in one left and one right interaction simultaneously. In the left interaction the man-in-the-middle adversary A interacts with C receiving a commitment to values v , using identity id of its choice. In the right interaction A interacts with R attempting to commit to a related value \tilde{v} , again using identities of its choice $\tilde{\text{id}}$. If the right commitment is invalid, or undefined, its value \tilde{v} is set to \perp ; so is it, when the adversary picks the same identity left and right, that is $\tilde{\text{id}} = \text{id}$ — i.e., a commitment where the adversary uses the same identity as the honest committer is considered invalid. Let $\text{mim}_{\langle C, R \rangle}^A(v, z)$ denote a random variable that describes the values \tilde{v} and the view of A , in the above experiment.

Definition 15 ([25, 67, 51]). *A commitment scheme $\langle C, R \rangle$ is said to be non-malleable (with respect to itself) if for every probabilistic polynomial-time man-in-the-middle adversary A , the following ensembles are computationally indistinguishable:*

$$\begin{aligned} & \left\{ \text{mim}_{\langle C, R \rangle}^A(v_1, z) \right\}_{n \in N, v_1, v_2 \in \{0,1\}^n, z \in \{0,1\}^*} \\ & \left\{ \text{mim}_{\langle C, R \rangle}^A(v_2, z) \right\}_{n \in N, v_1, v_2 \in \{0,1\}^n, z \in \{0,1\}^*} \end{aligned}$$

NON-MALLEABILITY W.R.T. ARBITRARY k -ROUND PROTOCOLS. Consider a man-in-the-middle adversary A that participates in one left interaction, communicating with a machine B , and one right interaction, acting as a committer using the commitment scheme $\langle C, R \rangle$. As in the standard definition of non-malleability, A can adaptively choose the identity in the right interaction. We denote by $\text{mim}_{\langle C, R \rangle}^{B,A}(y, z)$ the random variable consisting of the view of $A(z)$ and the value it commits on the right in a man-in-the-middle execution when communicating

with $B(y)$ on the left and honest receivers on the right. Intuitively, we say that $\langle C, R \rangle$ is non-malleable w.r.t B if $\text{mim}_{\langle C, R \rangle}^{B,A}(y_1, z)$ and $\text{mim}_{\langle C, R \rangle}^{B,A}(y_2, z)$ are indistinguishable whenever interactions with $B(y_1)$ and $B(y_2)$ cannot be distinguished. More formally, let $\text{view}_A[\langle B(y), A(z) \rangle]$ denote the view of $A(z)$ in an interaction with $B(y)$.

Definition 16. Let $\langle C, R \rangle$ be a commitment scheme, and B an interactive Turing machine. We say the commitment scheme $\langle C, R \rangle$ is non-malleable w.r.t. B , if for every probabilistic polynomial-time man-in-the-middle adversary A , and every two sequences $\{y_n^1\}_{n \in N}$ and $\{y_n^2\}_{n \in N}$, such that

$$\{\text{view}_A[\langle B(y_n^1), A(z) \rangle]\}_{n \in N, z \in \{0,1\}^*} \approx \{\text{view}_A[\langle B(y_n^2), A(z) \rangle]\}_{n \in N, z \in \{0,1\}^*}$$

it holds that:

$$\left\{ \text{mim}_{\langle D, E \rangle, \langle C, R \rangle}^{B,A}(y_n^1, z) \right\}_{n \in N, z \in \{0,1\}^*} \approx \left\{ \text{mim}_{\langle D, E \rangle, \langle C, R \rangle}^{B,A}(y_n^2, z) \right\}_{n \in N, z \in \{0,1\}^*}$$

We say that $\langle C, R \rangle$ is non-malleable w.r.t arbitrary k -round protocols if $\langle C, R \rangle$ is non-malleable w.r.t any machine B that interacts with the man-in-the-middle adversary in k rounds, also called as a k -robust commitment scheme. Such commitment schemes are easy to construct: any commitment scheme that is “extractable” and has more than k “rewinding slots” is directly non-malleable w.r.t. arbitrary k -round protocols.

We focus on non-malleability w.r.t 5-round protocols. As all known non-malleable commitment scheme either directly satisfy this property (for instance, the non-malleable commitment schemes of [25, 51] have many rewinding slots) or can be easily modified to do so¹, we call such protocols *robust*.

4.3.2 \mathcal{SNMWI} Argument of Knowledge Protocol $\langle P_s, V_s \rangle$

Let $\langle C, R \rangle$ be a 4-robust non-malleable commitment, and $\langle \hat{P}, \hat{V} \rangle$ be a 4-round zero-knowledge argument of knowledge system [30, 9]. The \mathcal{SNMWI} argument of knowledge $\langle P_s, V_s \rangle$ for **NP** language L , proceeds in the following two phases, on

¹All non-malleable commitment scheme contain a proof of knowledge protocol as a sub protocol; non-malleability w.r.t k -round protocols is easily achieved by repeating this proof of knowledge protocol for k times. See [49] for more details.

common inputs the security parameter n , statement x , and identity id , and private input w , $(x, w) \in R_L$, to the prover.

1. In the Committing Phase, the prover provides (sequentially) two commitments to the witness w using $\langle C, R \rangle$.
2. In the Proving Phase, the prover proves to the verifier that it has committed to the valid witness w in the committing phase, using $\langle \hat{P}, \hat{V} \rangle$.

A formal description of the protocol appears in figure 4.1.

Protocol $\langle P_s, V_s \rangle$
Common Input: Statement $x \in L$, security parameter n and identity id .
Private Input for Prover: The witness w of statement x , $(w, x) \in R_L$.
Committing Phase:
P_s uniformly chooses σ_1 and σ_2 from $\{0, 1\}^{\text{poly}(n)}$.
$P_s \rightarrow V_s$: $\langle C, R \rangle$ commitment to w using randomness σ_1 . Let \mathcal{T}_1 be the transcript of messages generated.
$P_s \rightarrow V_s$: $\langle C, R \rangle$ commitment to w using randomness σ_2 . Let \mathcal{T}_2 be the transcript of messages generated.
Proving Phase:
$P_s \leftrightarrow V_s$: a $\langle \hat{P}, \hat{V} \rangle$ proof of the statement: there exist values w , σ_1 and σ_2 s.t $w \in R_L(x)$, and \mathcal{T}_1 and \mathcal{T}_2 are two valid commitments to w using randomness σ_1 and σ_2 respectively.

Figure 4.1: Strongly Non-Malleable \mathcal{WI} Argument of Knowledge for **NP**

It follows directly from the argument-of-knowledge property of $\langle \hat{P}, \hat{V} \rangle$ that $\langle P_s, V_s \rangle$ is an argument of knowledge. We therefore focus on the \mathcal{SNMWI} property of $\langle P_s, V_s \rangle$.

Lemma 2. *For every uniform \mathcal{PPT} man-in-the-middle adversary A , and every two sequences of input distributions $\{D_n^1\}_{n \in N}$ and $\{D_n^2\}_{n \in N}$, it holds that if $\{(x, y, z) \leftarrow D_n^1 : (x, z)\}_{n \in N}$ and $\{(x, y, z) \leftarrow D_n^2 : (x, z)\}_{n \in N}$ are computationally indistinguishable, so are the following ensembles:*

$$\begin{aligned} & \{(x, y, z) \leftarrow D_n^1 : \text{mim}_{\langle P_s, V_s \rangle}^A(x, y, z)\}_{n \in N} \\ & \{(x, y, z) \leftarrow D_n^2 : \text{mim}_{\langle P_s, V_s \rangle}^A(x, y, z)\}_{n \in N} \end{aligned}$$

Proof. Below fix an arbitrary adversary A and two sequences of input distributions $\{D_n^1\}_{n \in N}$ and $\{D_n^2\}_{n \in N}$, such that $\{(x, y, z) \leftarrow D_n^1 : (x, z)\}_{n \in N}$ and $\{(x, y, z) \leftarrow D_n^2 : (x, z)\}_{n \in N}$ are indistinguishable. The goal is to show that in a man-in-the-middle execution of $\langle P_s, V_s \rangle$, the view of A combined with the unique witness of the right interaction are indistinguishable, when the input distribution changes from D_n^1 to D_n^2 . Notice that when the right interaction succeeds, it follows from the soundness of the ZK proof that, except from negligible probability, A commits to the same value in the two commitments (of the Committing Phase) and the value is the valid witness of the right interaction. (In the negligible probability event that the right interaction succeeds but A commits to different values in the two commitments, the committed value is set to \perp . Moreover, if the right interaction fails or has the same identity as the left interaction, the committed value is set to \perp as well). Therefore, the combined view and value committed to by A in the two commitments are statistically close to the combined view and the valid witness on the right; we denote the former random variable by $\overline{\text{mim}}_{\langle P_s, V_s \rangle}^A(x, y, z)$. It thus suffices to show the indistinguishability of the following ensembles:

$$\begin{aligned} & \left\{ (x, y, z) \leftarrow D_n^1 : \overline{\text{mim}}_{\langle P_s, V_s \rangle}^A(x, y, z) \right\}_{n \in N} \\ & \left\{ (x, y, z) \leftarrow D_n^2 : \overline{\text{mim}}_{\langle P_s, V_s \rangle}^A(x, y, z) \right\}_{n \in N} \end{aligned}$$

Towards this, consider the following two scenarios:

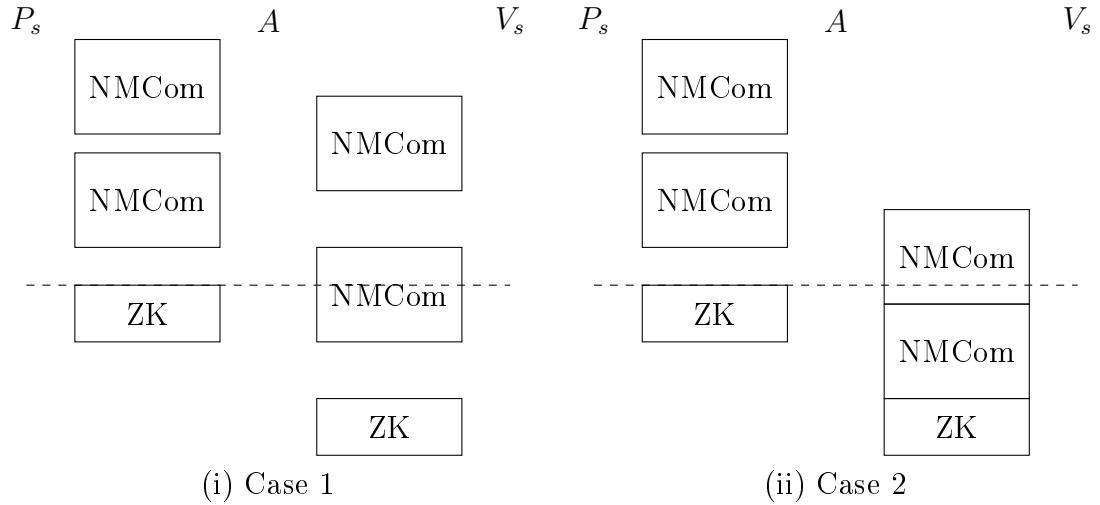


Figure 4.2: The two cases in a man-in-the-middle execution of $\langle P_s, V_s \rangle$ with adversary A .

Case 1: Consider the case that A completes the first commitment before the ZK

proof on the left starts, as shown in figure 4.2 (i). In this case, the value committed to by A in the first commitment is independent of the ZK proof in the left interaction. It essentially follows from the non-malleability of $\langle C, R \rangle$ and the (standard stand-alone) simulability of the ZK proof that, the combined view \mathcal{V} and value \tilde{v} A commits to in the *first* commitment would remain essentially the same, even if A had received two commitments to an arbitrary value, say 0^n , in the left interaction, instead of commitments to the true witness y . Suppose that this is the case (that A does receive commitments to 0^n in the left interaction). Then the only difference between the executions with input distributions D_n^1 and D_n^2 lies in the common input x and the auxiliary input z of the left interaction. It now follows from the robustness of $\langle C, R \rangle$ and the indistinguishability of $\{(x, y, z) \leftarrow D_n^1 : (x, z)\}_{n \in N}$ and $\{(x, y, z) \leftarrow D_n^2 : (x, z)\}_{n \in N}$ that, the view and the committed value, \mathcal{V} and \tilde{v} , are indistinguishable when the input distribution changes from D_n^1 to D_n^2 . Therefore, conditioned on Case 1 occurring in the execution, the view and value A commits to in the first commitment are indistinguishable, when changing from using distribution D_n^1 to D_n^2 . Formally, in a man-in-the-middle execution, we denote by $\overline{\text{mim}}1_{\langle P_s, V_s \rangle}^A(x, y, z)$, the view and the value $A(z)$ commits to in the first commitment if Case 1 occurs in the execution, and \perp , otherwise. Below, we show the following claim:

Claim 1. *The following ensembles are indistinguishable:*

$$\begin{aligned} & \left\{ (x, y, z) \leftarrow D_n^1 : \overline{\text{mim}}1_{\langle P_s, V_s \rangle}^A(x, y, z) \right\}_{n \in N} \\ & \left\{ (x, y, z) \leftarrow D_n^2 : \overline{\text{mim}}1_{\langle P_s, V_s \rangle}^A(x, y, z) \right\}_{n \in N} \end{aligned}$$

Case 2: Consider the case that the first commitment on the right coincides with the ZK proof on the left, as shown in figure 4.2 (ii). In this case, the *second* commitment on the right comes after both commitments on the left, which means that there are at most 4 rounds remaining in the left interaction (since by definition the ZK proof has 4 rounds). By relying on the non-malleability with respect to 5-round protocols of $\langle C, R \rangle$, it follows that the view and the value committed to by A in the second commitment, conditioned on Case 2 occurring, are indistinguishable, when the input distribution changes from D_n^1 or D_n^2 . Similar to the previous case, we define the random variable $\overline{\text{mim}}2_{\langle P_s, V_s \rangle}^A(x, y, z)$ to describe the view and the value committed to by A in

the second commitment, if Case 2 occurs, and \perp otherwise. Below, we show the following claim:

Claim 2. *The following ensembles are indistinguishable:*

$$\begin{aligned} & \left\{ (x, y, z) \leftarrow D_n^1 : \overline{\text{mim}}_{\langle P_s, V_s \rangle}^A(x, y, z) \right\}_{n \in N} \\ & \left\{ (x, y, z) \leftarrow D_n^2 : \overline{\text{mim}}_{\langle P_s, V_s \rangle}^A(x, y, z) \right\}_{n \in N} \end{aligned}$$

Before we proceed to provide the formal proofs of Claim 1 and 2. We first show that the lemma follows from the two claims. More precisely, for an arbitrary distinguisher D , there exists a negligible function ε , such that, for all n , the probability $\mathcal{P}(n)$ that D distinguishes the view and the value A commits to in an execution with distribution D_n^1 or D_n^2 is smaller than $\varepsilon(n)$, i.e.

$$\begin{aligned} \mathcal{P}(n) = & \left| \Pr \left[(x, y, z) \leftarrow D_n^1 : D(\overline{\text{mim}}_{\langle P_s, V_s \rangle}^A(x, y, z)) = 1 \right] \right. \\ & \left. - \Pr \left[(x, y, z) \leftarrow D_n^2 : D(\overline{\text{mim}}_{\langle P_s, V_s \rangle}^A(x, y, z)) = 1 \right] \right| < \varepsilon(n) \end{aligned}$$

Let $\alpha^b(n)$ (and $\beta^b(n)$ resp.) denote the probability that Case 1 (and Case 2 resp.) occurs in a man-in-the-middle execution with input distribution D_n^b . Since in any execution, either Case 1 or Case 2 occurs, it holds that

$$\begin{aligned} \Pr \left[(x, y, z) \leftarrow D_n^b : D(\overline{\text{mim}}_{\langle P_s, V_s \rangle}^A(x, y, z)) = 1 \right] = & \\ & \alpha^b(n) \Pr \left[(x, y, z) \leftarrow D_n^b : D(\overline{\text{mim}}_{\langle P_s, V_s \rangle}^A(x, y, z)) = 1 \mid \text{Case 1} \right] \\ & + \beta^b(n) \Pr \left[(x, y, z) \leftarrow D_n^b : D(\overline{\text{mim}}_{\langle P_s, V_s \rangle}^A(x, y, z)) = 1 \mid \text{Case 2} \right] \end{aligned}$$

For convenience, use A_n^b and B_n^b as the shorthands for the two terms on the right hand side above. Then,

$$\mathcal{P}(n) = |(A_n^1 + B_n^1) - (A_n^2 + B_n^2)| \leq |A_n^1 - A_n^2| + |B_n^1 - B_n^2|$$

We show below that Claim 1 implies that there exists a negligible function ε^1 , such that A_n^1 and A_n^2 differ by at most $\varepsilon^1(n)$. Similarly, by Claim 2 there is another negligible function ε^2 , such that, $|B_n^1 - B_n^2| < \varepsilon^2(n)$. Thus $\mathcal{P}(n)$ is bounded by $(\varepsilon^1(n) + \varepsilon^2(n))$, a negligible amount, as desired.

Assume for contradiction that there exists a polynomial p , such that $|A_n^1 - A_n^2|$ is at least $\frac{1}{p(n)}$. Then using D we can construct another distinguisher D' such that

$$\left| \Pr [(x, y, z) \leftarrow D_n^1 : D'(\overline{\text{mim}}1_{\langle P_s, V_s \rangle}^A(x, y, z)) = 1] - \Pr [(x, y, z) \leftarrow D_n^2 : D'(\overline{\text{mim}}1_{\langle P_s, V_s \rangle}^A(x, y, z)) = 1] \right| \geq \frac{1}{p(n)}$$

More precisely, D' , on receiving a view \mathcal{V} and the value v committed to by A in the first commitment, simply outputs 0, if both of them are \perp (which means Case 1 does not occur); otherwise, it feeds \mathcal{V} and v to D , and outputs whatever D outputs. In other words, D' outputs 1 if and only if Case 1 occurs in the execution and D outputs 1 on \mathcal{V} and v ; this happens with probability almost A_n^b (since when the right interaction succeeds in \mathcal{V} , except from negligible probability, the committed value v in the first commitment is just the value A commits to in both commitments). Then by our hypothesis, D' violates Claim 1. Using exactly the same argument, we can show that the difference between B_n^1 and B_n^2 is also negligible. This completes the proof.

Next we turn to prove Claim 1 and 2 formally.

Proof of Claim 1. Towards this, we consider a sequence of hybrid experiments H_0 , H_1 , H_2 and H_3 , each of which is a (simulated) man-in-the-middle execution of $\langle P_s, V_s \rangle$ with the adversary A . We denote by $\text{hyb}_n^i(A)$ the random variable describing the (simulated) view of A and the value that A commits in the first commitment in H_i . (The committed value is set to \perp if A fails to complete the right interaction in H_i or it uses the same identity in the left and right interactions.) Since we only focus on Case 1 in this proof, if in a hybrid experiment, Case 1 does not occur, the execution is cut off and the correspondingly $\text{hyb}_n^i(A)$ is set to \perp . Below, we describe how each hybrid experiment proceeds when Case 1 does occur, and we show that, in successive hybrids, $\{\text{hyb}_n^i(A)\}_{n \in N}$ and $\{\text{hyb}_n^{i+1}(A)\}_{n \in N}$ are indistinguishable.

Hybrid H_0 is simply an honest man-in-the-middle execution of $\langle P_s, V_s \rangle$ with A , using inputs (x, y, z) sampled randomly from D_n^1 . Since the execution is cut off and hyb_n^0 set to \perp , if Case 1 does not occur, it holds that:

$$\{\text{hyb}_n^0(A)\}_{n \in N} = \{(x, y, z) \leftarrow D_n^1 : \overline{\text{mim}}1_{\langle P_s, V_s \rangle}^A(x, y, z)\}_{n \in N}$$

Hybrid H_1 proceeds identically as H_0 until the ZK proof on the left is about to start; let \mathcal{T} be the transcript of the two commitments on the left, and ρ the partial joint view of A and the right receiver generated so far. If Case 1 occurs in ρ , instead of completing the rest of the execution honestly as in H_0 , H_1 simulates the execution by simulating the ZK proof on the left. More precisely, consider a verifier V^* for the ZK proof, which, on auxiliary input ρ , internally emulates the right interaction with A and V_s , by feeding them with their partial views from ρ , and forwarding messages in the external proof to A . By the ZK property, there exists a simulator S such that, S on inputs (x', z') outputs a view indistinguishable to the view of $V^*(z')$ participating in a ZK proof of the statement x' . By the construction of V^* , the view output by S contains the view \mathcal{V}_A of A and the view \mathcal{V}_{V_s} of V_s . Hence H_1 simply invokes S on inputs $((x, \mathcal{T}), \rho)$, and sets $\text{hyb}_n^1(A)$ to \mathcal{V}_A (outputted by S) and the value A commits to in the first commitment in ρ . We claim that the following holds for H_0 and H_1 .

$$\{\text{hyb}_n^0(A)\}_{n \in N} \approx \{\text{hyb}_n^1(A)\}_{n \in N}$$

Observe that H_0 and H_1 proceed identically before the ZK proof on the left starts; let ρ be any arbitrary partial joint view of A and the right verifier (in H_0 and H_1) before the left ZK proof starts. Conditioned on ρ occurring, if it is a Case 2 scenario, $\text{hyb}_n^0(A) = \text{hyb}_n^1(A) = \perp$; otherwise, it follows from the simulability of the ZK proof that the simulated view \mathcal{V}_A in $\text{hyb}_n^1(A)$ is indistinguishable from the perfect view in $\text{hyb}_n^0(A)$. Furthermore, in Case 1, the actual value v committed to in the first commitment is decided in ρ (since the first commitment completes in ρ); but it may be replaced with \perp if the right infection fails after ρ . However, by the indistinguishability between the views in $\text{hyb}_n^0(A)$ and $\text{hyb}_n^1(A)$, the probabilities that v is replaced with \perp in H_0 and H_1 differ by at most a negligible amount. Therefore we conclude the claim.

Hybrid H_2 proceeds identically to H_1 , with the exception that in the left interaction, the value committed to in the first commitment is 0^n instead of the valid witness y . We define hyb_n^2 similar to hyb_n^1 . When Case 1 occurs, hyb_n^2 is set to the simulated view of A and the value committed to in the first commitment on the right. It then follows from the non-malleability of $\langle C, R \rangle$

that

$$\{\mathsf{hyb}_n^1(A)\}_{n \in N} \approx \{\mathsf{hyb}_n^2(A)\}_{n \in N}$$

Hybrid H_3 This experiment proceeds identically to H_2 , except that in the left interaction, the second commitment is also set to 0^n . (hyb_n^3 is defined similar to hyb_n^2 .) Using exactly the same argument as for hybrids H_1 and H_2 , it follows that

$$\{\mathsf{hyb}_n^2(A)\}_{n \in N} \approx \{\mathsf{hyb}_n^3(A)\}_{n \in N}$$

The above hybrid experiments were considered for the case when the input distribution for the left interaction was D_n^1 . We can define similarly, hybrid experiments \overline{H}_0 to \overline{H}_3 for input distribution D_n^2 and denote $\overline{\mathsf{hyb}}_n^i(A)$ the corresponding random variables. Observe that, the only difference between hybrid \overline{H}_3 and H_3 is that the common input x and auxiliary input z are sampled from D_n^2 and D_n^1 respectively. (Note that we ignore the private input y to the left prover as in both the experiments H_3 and \overline{H}_3 , the left interaction is simulated without using y .) It then follows from the indistinguishability between $\{(x, y, z) \leftarrow D_n^1 : (x, z)\}_{n \in N}$ and $\{(x, y, z) \leftarrow D_n^2 : (x, z)\}_{n \in N}$, and the non-malleability w.r.t. arbitrary 5-round protocols of $\langle C, R \rangle$ (note that here we consider a non-interactive protocol that simply samples from the two distributions mentioned above), that

$$\{\mathsf{hyb}_n^3(A)\}_{n \in N} \approx \{\overline{\mathsf{hyb}}_n^3(A)\}_{n \in N}$$

This completes the proof of the claim.

□

Proof of Claim 2. Assume for contradiction that there exists a distinguisher D and a polynomial p , such that for infinitely many n , it holds that

$$\left| \Pr[(x, y, z) \leftarrow D_n^1 : D(\overline{\mathsf{mim}}2_{\langle P_s, V_s \rangle}^A(x, y, z)) = 1] - \Pr[(x, y, z) \leftarrow D_n^2 : D(\overline{\mathsf{mim}}2_{\langle P_s, V_s \rangle}^A(x, y, z)) = 1] \right| \geq \frac{1}{p(n)}$$

Fix one such n . Given A , we construct an adversary \tilde{A} , a machine B , and a distinguisher \tilde{D} , such that, B , on input 1^n or 2^n , interacts with \tilde{A} in 5 rounds, and \tilde{D} distinguishes the view and committed value (using $\langle C, R \rangle$) by \tilde{A} in interaction with B on different inputs, with probability $\frac{1}{p(n)}$.

The machine $B(b^n)$, in essence, simply interacts as the prover of the ZK proof of the protocol $\langle P_s, V_s \rangle$. More precisely, $B(b^n)$, first prepares a random statement s for the ZK proof, i.e. it honestly emulates a man-in-the-middle execution of $\langle P_s, V_s \rangle$ with A using inputs $(x, y, z) \leftarrow D_n^b$, until the ZK proof on the left starts. Let \mathcal{T} be the transcript of the two commitments in the left interaction, and ρ the partial joint view of $A(z)$ and the right verifier generated so far. $B(b^n)$, then sets the statement s to (x, \mathcal{T}) and sends the statement s together with z and ρ to \tilde{A} , followed by a ZK proof of the statement that \mathcal{T} is a transcript of two commitments to the valid witness of x using $\langle C, R \rangle$. The adversary \tilde{A} , on the other hand, externally interacts with $B(b^n)$, and, on receiving s , z and ρ from $B(b^n)$ in the first round, starts emulating honestly a man-in-the-middle execution of $\langle P_s, V_s \rangle$ from ρ . More specifically, it first feeds $A(z)$ and the right verifier their partial views from ρ . To emulate the left interaction, it externally forwards messages in the ZK proof to $B(b^n)$. For the right interaction, it emulates the right receiver honestly for A , except that, if the second commitment starts outside ρ (i.e. Case 2 occurs), it forwards the commitment externally.

The distinguisher \tilde{D} , on input the view \mathcal{V} and the values \tilde{v} committed to by \tilde{A} , first reconstructs the view \mathcal{V}_A and the value \tilde{v}' committed to in the second commitment by $A(z)$ in emulation by \tilde{A} , i.e. \tilde{D} extracts \mathcal{V}_A from \mathcal{V} ; if Case 2 occurs in \mathcal{V}_A , it sets \tilde{v}' to \tilde{v} if the right interaction succeeds in \mathcal{V}_A and \perp otherwise; if Case 2 does not occur, it sets both \mathcal{V}_A and \tilde{v}' to \perp . \tilde{D} then executes the distinguisher D on the reconstructed view and committed value, \mathcal{V}_A and \tilde{v}' , and outputs what D outputs. From the construction of \tilde{A} it follows that the emulation of the man-in-the-middle execution with A is identical to an actual execution with A . Therefore, \mathcal{V}_A and \tilde{v}' are identically distributed to $\{(x, y, z) \leftarrow D_n^b : \overline{\text{mim}}2_{\langle P_s, V_s \rangle}^A(x, y, z)\}$ when \tilde{A} interacts externally with $B(b^n)$. Hence, \tilde{D} distinguishes the view and the values committed to by \tilde{A} with probability $\frac{1}{p(n)}$, when interacting with $B(1^n)$ or $B(2^n)$.

Given that the view and the value committed to by \tilde{A} after the interactions with B on input 1^n or 2^n are distinguishable (by D), by the non-malleability w.r.t. arbitrary 5-round protocols of $\langle C, R \rangle$, it follows that *even only the views* of \tilde{A} in interactions with B on input 1^n or 2^n are distinguishable. (Here we rely on the fact that the ZK proof in the protocol $\langle P_s, V_s \rangle$ has only 4 rounds, and thus B interacts with \tilde{A} in 5 rounds. Note that the choice of statement by B only adds one round.) However, we show in the subclaim below that the view of \tilde{A} in interaction with

$B(1^n)$ or $B(2^n)$ are indistinguishable, which gives a contradiction, and concludes the claim.

Subclaim 1.

$$\left\{ \text{view}_{\tilde{A}}[\langle B(1^n), \tilde{A} \rangle] \right\}_{n \in N} \approx \left\{ \text{view}_{\tilde{A}}[\langle B(2^n), \tilde{A} \rangle] \right\}_{n \in N}$$

This subclaim follows from the fact that the whole protocol $\langle P_s, V_s \rangle$ is zero-knowledge (this follows using standard techniques), and hence strongly \mathcal{WI} (see [31]).

□

□

4.4 Robust \mathcal{SNMWI} Arguments

In our actual constructions, we require a slightly stronger non-malleability requirement from the \mathcal{SNMWI} proofs. We here consider the notion of robust non-malleability analogous to [49] with respect to arbitrary k -round protocols. \mathcal{SNMWI} considers a man-in-the-middle execution of the protocol $\langle P_s, V_s \rangle$, in which the adversary A simultaneously participates in two interactions of $\langle P_s, V_s \rangle$, one left and one right interaction. A k -robust \mathcal{SNMWI} proof additionally considers man-in-the-middle adversaries that interacts with a machine B on the left in k -rounds and one interaction of $\langle P_s, V_s \rangle$ on the right. More precisely, in the left interaction, the adversary A , on auxiliary input z , interacts with machine B on input v in at most k -rounds. In the right interaction, A adaptively chooses a statement \tilde{x} and tag \tilde{id} and attempts to provide a proof to V_s . Let \tilde{y} denote the witness associated with \tilde{x} , unless A fails in the right interaction ; in this case \tilde{y} is set to \perp . Let $\text{mim}_{\langle P_s, V_s \rangle}^{A, B}(v)$ denote the random variable that describes the witness \tilde{y} combined with the view of A in the above man-in-the-middle experiment.

Definition 17 (Strongly Non-Malleable \mathcal{WI} w.r.t. B). *Let $\langle P_s, V_s \rangle$ be a tag-based argument system for R_L and B a \mathcal{PPT} ITM. We say that $\langle P_s, V_s \rangle$ is strongly non-malleable witness-indistinguishable w.r.t. B , If for every two sequences of inputs*

$\{v_n^1\}_{n \in \mathbb{N}}$ and $\{v_n^2\}_{n \in \mathbb{N}}$, such that, for all \mathcal{PPT} machines \tilde{A} , it holds that

$$\left\{ \text{view}_{\tilde{A}}[\langle B(v_n^1), \tilde{A}(z) \rangle] \right\}_{n \in \mathbb{N}, z \in \{0,1\}^*} \approx \left\{ \text{view}_{\tilde{A}}[\langle B(v_n^2), \tilde{A}(z) \rangle] \right\}_{n \in \mathbb{N}, z \in \{0,1\}^*}$$

then it also holds that, for every non-uniform \mathcal{PPT} man-in-the-middle adversary A ,

$$\left\{ \text{mim}_{\langle P_s, V_s \rangle}^{A,B}(v_n^1, z) \right\}_{n \in \mathbb{N}, z \in \{0,1\}^*} \approx \left\{ \text{mim}_{\langle P_s, V_s \rangle}^{A,B}(v_n^2, z) \right\}_{n \in \mathbb{N}, z \in \{0,1\}^*}$$

We say that $\langle P_s, V_s \rangle$ is k -robust strongly non-malleable witness-indistinguishable if $\langle P_s, V_s \rangle$ is strongly non-malleable witness indistinguishable w.r.t. every \mathcal{PPT} machine B that sends at most k messages in its interaction with the adversary.

Now, we proceed to show how to construct a k -robust \mathcal{SNMWI} argument. In fact, we will show that the protocol $\langle P_s, V_s \rangle$ constructed in the previous section from non-malleable commitments is k -robust if the non-malleable commitment scheme used in the protocol is k -robust.

Lemma 3. *Let $k > 4$ and $\langle C, R \rangle$ be a k -robust non-malleable commitment scheme. Then the protocol $\langle P_s, V_s \rangle$ described in Figure 4.1 is a k -robust \mathcal{SNMWI} argument of knowledge for R_L .*

Proof: Assume for contradiction, there exists an adversary \tilde{A} , machine B , input sequences $\{v_n^1\}_{n \in \mathbb{N}}$ and $\{v_n^2\}_{n \in \mathbb{N}}$, auxiliary input sequence $\{z_n\}_{n \in \mathbb{N}}$, distinguisher D , polynomial $p(\cdot)$ such that

$$\left\{ \text{view}_{\tilde{A}}[\langle B(v_n^1), \tilde{A}(z_n) \rangle] \right\}_{n \in \mathbb{N}} \approx \left\{ \text{view}_{\tilde{A}}[\langle B(v_n^2), \tilde{A}(z_n) \rangle] \right\}_{n \in \mathbb{N}} \quad (4.1)$$

and for infinitely many n , it holds that

$$\Pr \left[D(\text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}, B}(v_n^1, z_n)) = 1 \right] - \Pr \left[D(\text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}, B}(v_n^2, z_n)) = 1 \right] \geq \frac{1}{p(n)} \quad (4.2)$$

Fix an n for which Equation 4.2 holds. Then consider the following man-in-the-middle adversary \tilde{A}^* for $\langle C, R \rangle$. Adversary \tilde{A}^* interacts with machine B on the left and with an honest receiver on the right using $\langle C, R \rangle$. On input z_n , \tilde{A}^* incorporates \tilde{A} and proceeds as follows:

- For the messages exchanged by \tilde{A} on the left with B , \tilde{A}^* simply forwards the message with the external machine B on the left.

- For the right interaction, \tilde{A}^* chooses the first commitment proved using $\langle C, R \rangle$ and forward the messages to the external receiver on the right.
- All other interactions are emulated internally. On completion, \tilde{A}^* outputs what \tilde{A} outputs.

First, we note that the value committed to in the first (or second) commitment using $\langle P_s, V_s \rangle$ is equal to the witness used by the adversary in the right interaction, except with negligible probability. This follows from the statistical-binding property of the commitment and the soundness of the \mathcal{ZK} protocol. Therefore, we have that the following pairs of ensembles are statistically-close.

- $\left\{ \text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}^*, B}(v_n^1, z_n) \right\}_{n \in \mathbb{N}}$ and $\left\{ \text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}, B}(v_n^1, z_n) \right\}$
- $\left\{ \text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}^*, B}(v_n^2, z_n) \right\}_{n \in \mathbb{N}}$ and $\left\{ \text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}, B}(v_n^2, z_n) \right\}$

Now, using Equation 4.2, it follows that

$$\Pr \left[D(\text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}^*, B}(v_n^1, z_n)) = 1 \right] - \Pr \left[D(\text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}^*, B}(v_n^2, z_n)) = 1 \right] \geq \frac{1}{p(n)} - \nu(n)$$

for some negligible function $\nu(\cdot)$. Therefore, \tilde{A}^* violates k -robustness of $\langle C, R \rangle$ and we arrive at a contradiction. \square

4.5 Sequential composition of \mathcal{SNMWI} Arguments

In this section, we prove that \mathcal{SNMWI} arguments are closed under 2-sequential repetitions. The proof extends for constant repetitions, however, for the purposes of this thesis, we only require \mathcal{SNMWI} arguments secure under two sequential repetitions. Given a protocol $\langle P, V \rangle$, let $\langle P^2, V^2 \rangle$ denote the protocol obtained by repeating the interaction $\langle P, V \rangle$ sequentially twice.

Proposition 1. *Let $\langle P_s, V_s \rangle$ be a \mathcal{SNMWI} argument for language for R_L that is zero-knowledge. Then $\langle P_s^2, V_s^2 \rangle$ is a \mathcal{SNMWI} argument for R_L .*

Proof: Assume for contradiction, there exists a man-in-the-middle adversary A , ensembles $\{D_n^1\}_n$ and $\{D_n^2\}_n$, sequences $\{\text{id}_n\}_n$ polynomial $p(n)$, such that the following ensembles are indistinguishable for $n \in \mathbb{N}$

- $\{(x, y, z) \leftarrow D_n^1 : (x, z)\}_{n \in \mathbb{N}}$
- $\{(x, y, z) \leftarrow D_n^2 : (x, z)\}_{n \in \mathbb{N}}$

and for infinitely many n , the following holds:

$$\begin{aligned} & \Pr \left[(x, y, z) \leftarrow D_n^1 : D(\mathsf{mim}_{\langle P_s^2, V_s^2 \rangle}^A(x, y, z, \mathsf{id}_n)) = 1 \right] \\ & - \Pr \left[(x, y, z) \leftarrow D_n^1 : D(\mathsf{mim}_{\langle P_s^2, V_s^2 \rangle}^A(x, y, z, \mathsf{id}_n)) = 1 \right] \geq \frac{1}{p(n)} \end{aligned} \quad (4.3)$$

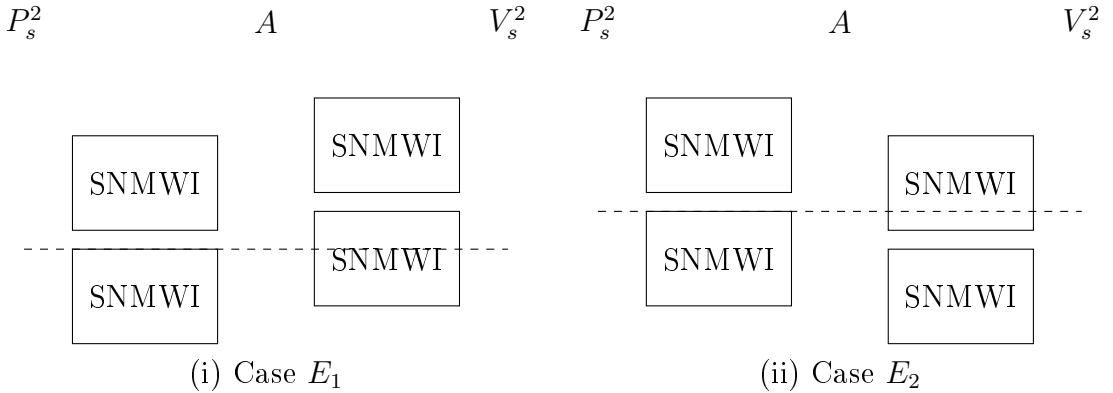


Figure 4.3: The two cases E_1 and E_2 in a man-in-the-middle execution of $\langle P_s^2, V_s^2 \rangle$ with adversary A

In a man-in-the middle execution with A , two scenarios arise depending on how the adversary schedules the messages. The first scenario, that we represent by the event E_1 occurs when A completes the first $\langle P_s, V_s \rangle$ proof as the prover on the right in $\langle P_s^2, V_s^2 \rangle$ before the first $\langle P_s, V_s \rangle$ proof is received by A on the left. The second scenario or event E_2 occurs when A completes the first $\langle P_s, V_s \rangle$ proof on the right after the first $\langle P_s, V_s \rangle$ proof on the left is received. See Figure 4.3 for a pictorial representation of the two scenarios. It follows that, if A successfully completes without aborting, exactly one of E_1 or E_2 definitely occurs.

Therefore it holds that one of the two equations below holds for infinitely many n .

$$\begin{aligned} & \Pr \left[(x, y, z) \leftarrow D_n^1 : D(\mathsf{mim}_{\langle P_s^2, V_s^2 \rangle}^A(x, y, z, \mathsf{id}_n)) = 1 \wedge E_1 \right] \\ & - \Pr \left[(x, y, z) \leftarrow D_n^1 : D(\mathsf{mim}_{\langle P_s^2, V_s^2 \rangle}^A(x, y, z, \mathsf{id}_n)) = 1 \wedge E_1 \right] \geq \frac{1}{2p(n)} \end{aligned} \quad (4.4)$$

$$\begin{aligned} & \Pr \left[(x, y, z) \leftarrow D_n^1 : D(\mathsf{mim}_{\langle P_s^2, V_s^2 \rangle}^A(x, y, z, \mathsf{id}_n)) = 1 \wedge E_2 \right] \\ & - \Pr \left[(x, y, z) \leftarrow D_n^1 : D(\mathsf{mim}_{\langle P_s^2, V_s^2 \rangle}^A(x, y, z, \mathsf{id}_n)) = 1 \wedge E_2 \right] \geq \frac{1}{2p(n)} \end{aligned} \quad (4.5)$$

In either case, we show how to construct an adversary that violates the \mathcal{SNMWI} property of $\langle P_s, V_s \rangle$ and arrive at a contradiction.

RESOLVING CASE 1: Assume for contradiction that Equation 4.4 holds for infinitely many n .

Consider an adversary A_1 that on input (x, z) proceeds as follows. It incorporates A in the following manner:

- For the $\langle P_s, V_s \rangle$ proof on the left, A_1 forwards messages in the first $\langle P_s, V_s \rangle$ proof received by A to the external prover (on the left).
- For the $\langle P_s, V_s \rangle$ proof on the right, A_1 forwards messages in the first $\langle P_s, V_s \rangle$ proof given by A to the external verifier (on the right).
- All other messages are simulated internally. On completion of the left proof, A_1 cuts off A and outputs the partial view of A at that instant. If E_1 does not occur, A_1 simply outputs \perp .

Using the \mathcal{SNMWI} property of $\langle P_s, V_s \rangle$ we obtain the following claim.

Claim 3. *The following distributions are indistinguishable over $n \in \mathbb{N}$:*

- $\left\{ (x, y, z) \leftarrow D_n^1; (\text{VIEW}, w) \leftarrow \text{mim}_{\langle P_s, V_s \rangle}^{A_1}(x, y, z, \text{id}_n) : ((x, \text{VIEW}), w) \right\}_{n \in \mathbb{N}}$
- $\left\{ (x, y, z) \leftarrow D_n^2; (\text{VIEW}, w) \leftarrow \text{mim}_{\langle P_s, V_s \rangle}^{A_1}(x, y, z, \text{id}_n) : ((x, \text{VIEW}), w) \right\}_{n \in \mathbb{N}}$

Next, consider a second adversary A_2 that on input $(x, (\text{VIEW}, w))$ proceeds as follows. It incorporates A in the following manner:

- If $\text{VIEW} = \perp$, A_2 halts outputting \perp . Otherwise, it feeds A with the messages in VIEW .
- For the $\langle P_s, V_s \rangle$ proof on the left, A_2 forwards messages in the second $\langle P_s, V_s \rangle$ proof received by A to the external prover (on the left).
- All other messages are simulated internally. On completion, A_2 outputs what A outputs concatenated with w .

Define distributions \tilde{D}_n^1 and \tilde{D}_n^2 as follows:

$$\begin{aligned}\tilde{D}_n^1 &= \left\{ (x, y, z) \leftarrow D_n^1; (\text{VIEW}, w) \leftarrow \mathsf{mim}_{\langle P_s, V_s \rangle}^{A_1}(x, y, z, \mathsf{id}_n) : (x, y, (\text{VIEW}, w)) \right\} \\ \tilde{D}_n^2 &= \left\{ (x, y, z) \leftarrow D_n^2; (\text{VIEW}, w) \leftarrow \mathsf{mim}_{\langle P_s, V_s \rangle}^{A_1}(x, y, z, \mathsf{id}_n) : (x, y, (\text{VIEW}, w)) \right\}\end{aligned}$$

It follows from construction that view emulated by A_2 in the man-in-the-middle experiment when the inputs are chosen by first sampling $(x, y, z) \leftarrow \tilde{D}_n^1$ (similarly \tilde{D}_n^2) and using (x, z) as input is identical to the view of A in the man-in-the-middle experiment using $\langle P_s^2, V_s^2 \rangle$ when inputs come from D_n^1 (resp., D_n^2). Furthermore, as we consider only well-behaved adversaries (i.e. adversaries that prove statements with unique witnesses), it follows that the witness w output by A_2 satisfies the property that is the witness of the statement A is proving on the right using $\langle P_s^2, V_s^2 \rangle$ in the internal emulation. Therefore, it holds that.

$$\begin{aligned}&\left\{ (x, y, z) \leftarrow \tilde{D}_n^1 : \mathsf{view}_{A_2}[\langle P(y), A_2(z) \rangle(x)] \right\} \\ &= \left\{ (x, y, z) \leftarrow D_n^1 : \mathsf{mim}_{\langle P_s^2, V_s^2 \rangle}^A(x, y, z, \mathsf{id}_n) \right\} \text{, and} \\ &\left\{ (x, y, z) \leftarrow \tilde{D}_n^2 : \mathsf{view}_{A_2}[\langle P(y), A_2(z) \rangle(x)] \right\} \\ &= \left\{ (x, y, z) \leftarrow D_n^2 : \mathsf{mim}_{\langle P_s^2, V_s^2 \rangle}^A(x, y, z, \mathsf{id}_n) \right\}\end{aligned}$$

Therefore, using Equation 4.4, we have that

$$\begin{aligned}\Pr \left[(x, y, z) \leftarrow \tilde{D}_n^1 : D(\mathsf{view}_{A_2}[\langle P(y), A_2(z) \rangle(x)]) = 1 \right] \\ - \Pr \left[(x, y, z) \leftarrow \tilde{D}_n^1 : D(\mathsf{view}_{A_2}[\langle P(y), A_2(z) \rangle(x)]) = 1 \right] \geq \frac{1}{2p(n)}\end{aligned}$$

From Claim 3, it follows that the following ensembles are indistinguishable over $n \in \mathbb{N}$

- $\left\{ (x, y, z) \leftarrow \tilde{D}_n^1 : (x, z) \right\}_{n \in \mathbb{N}}$
- $\left\{ (x, y, z) \leftarrow \tilde{D}_n^2 : (x, z) \right\}_{n \in \mathbb{N}}$

Since, D distinguishes the output of the man-in-the middle experiment with A_2 for infinitely many n , A_2 violates the Strong \mathcal{WI} property of $\langle P_s, V_s \rangle$. Thus, we arrive at a contradiction.

RESOLVING CASE 2: Assume for contradiction that Equation 4.5 holds for infinitely many n .

Along the lines of Case 1, we again consider an adversary A_1 that on input (x, z) proceeds as follows. It incorporates A in the following manner:

- For the $\langle P_s, V_s \rangle$ proof on the left, A_1 forwards messages in the first $\langle P_s, V_s \rangle$ proof received by A to the external prover (on the left).
- All other messages are simulated internally. On completion of the left proof, A_1 cuts off A and outputs the partial view of A at that instant. If E_2 does not occur, A_1 simply outputs \perp .

Using the Strong \mathcal{WI} property of $\langle P_s, V_s \rangle$ we obtain the following claim.

Claim 4. *The following distributions are indistinguishable over $n \in \mathbb{N}$:*

- $\{(x, y, z) \leftarrow D_n^1 : \text{VIEW}_{A_1}[\langle P_s(y), A_1(z) \rangle(x)]\}_{n \in \mathbb{N}}$
- $\{(x, y, z) \leftarrow D_n^2 : \text{VIEW}_{A_1}[\langle P_s(y), A_1(z) \rangle(x)]\}_{n \in \mathbb{N}}$

Next, consider a second adversary A_2 that on input (x, VIEW) proceeds as follows. It incorporates A in the following manner:

- If $\text{VIEW} = \perp$, A_2 halts outputting \perp . Otherwise, it feeds A with the messages in VIEW .
- For the $\langle P_s, V_s \rangle$ proof on the left, A_2 forwards messages in the second $\langle P_s, V_s \rangle$ proof received by A to the external prover (on the left).
- For the $\langle P_s, V_s \rangle$ proof on the right, A_2 forwards messages in the second $\langle P_s, V_s \rangle$ proof given by A to the external verifier (on the right).
- All other messages are simulated internally. On completion, A_2 outputs what A outputs.

Define distributions \tilde{D}_n^1 and \tilde{D}_n^2 as follows:

$$\begin{aligned}\tilde{D}_n^1 &= \{(x, y, z) \leftarrow D_n^1; \text{VIEW} \leftarrow \text{VIEW}_{A_1}[\langle P_s(y), A_1(z) \rangle(x)] : (x, y, \text{VIEW})\} \\ \tilde{D}_n^2 &= \{(x, y, z) \leftarrow D_n^2; \text{VIEW} \leftarrow \text{VIEW}_{A_1}[\langle P_s(y), A_1(z) \rangle(x)] : (x, y, \text{VIEW})\}\end{aligned}$$

As in Case 1, it follows from construction that view emulated by A_2 in the man-in-the-middle experiment when the inputs are according to \tilde{D}_n^1 (similarly \tilde{D}_n^2) is identical to the view of A in the man-in-the-middle experiment using $\langle P_s^2, V_s^2 \rangle$ when inputs come from D_n^1 (resp., D_n^2). Furthermore, as we consider only well-behaved adversaries (i.e. adversaries that prove statements with unique witnesses), it follows that the witness w used by the adversary in the second $\langle P_s, V_s \rangle$ proof is identical to the witness of the statement A is proving on the right using $\langle P_s^2, V_s^2 \rangle$ in the internal emulation. Therefore, it holds that.

$$\begin{aligned} & \left\{ (x, y, z) \leftarrow \tilde{D}_n^1 : \mathsf{mim}_{\langle P_s, V_s \rangle}^{A_2}(x, y, z, \mathsf{id}_n) \right\} \\ &= \left\{ (x, y, z) \leftarrow D_n^1 : \mathsf{mim}_{\langle P_s^2, V_s^2 \rangle}^A(x, y, z, \mathsf{id}_n) \right\}, \text{ and} \\ & \left\{ (x, y, z) \leftarrow \tilde{D}_n^2 : \mathsf{mim}_{\langle P_s, V_s \rangle}^{A_2}(x, y, z, \mathsf{id}_n) \right\} \\ &= \left\{ (x, y, z) \leftarrow D_n^2 : \mathsf{mim}_{\langle P_s^2, V_s^2 \rangle}^A(x, y, z, \mathsf{id}_n) \right\} \end{aligned}$$

Therefore, using Equation 4.5, we have that

$$\begin{aligned} & \Pr \left[(x, y, z) \leftarrow \tilde{D}_n^1 : D(\mathsf{mim}_{\langle P_s, V_s \rangle}^{A_2}(x, y, z, \mathsf{id}_n)) = 1 \right] \\ & - \Pr \left[(x, y, z) \leftarrow \tilde{D}_n^2 : D(\mathsf{mim}_{\langle P_s, V_s \rangle}^{A_2}(x, y, z, \mathsf{id}_n)) = 1 \right] \geq \frac{1}{2p(n)} \end{aligned}$$

From Claim 4, it follows that the following ensembles are indistinguishable for $n \in \mathbb{N}$

- $\left\{ (x, y, z) \leftarrow \tilde{D}_n^1 : (x, z) \right\}_{n \in \mathbb{N}}$
- $\left\{ (x, y, z) \leftarrow \tilde{D}_n^2 : (x, z) \right\}_{n \in \mathbb{N}}$

Since, D distinguishes the output of the man-in-the middle experiment with A_2 for infinitely many n , A_2 violates the \mathcal{SNMWI} property of $\langle P_s, V_s \rangle$. Thus, we arrive at a contradiction. This completes the proof of Proposition 1

□

Remark 1. *Following a similar proof strategy, it is possible to show that Strong \mathcal{WI} is also closed under constant sequential repetition.*

Remark 2. *The above proof can be extended to any constant number of sequential repetitions by establishing that the view of the adversary after receiving each proof on the left remains indistinguishable. However, it remains an open question as to whether \mathcal{SNMWI} or even the weaker property of Strong \mathcal{WI} is closed under non-constant sequential repetitions.*

CHAPTER 5

PROOF OF THE MAIN THEOREM

In this chapter, we state and prove the main theorem.

Theorem 4 (Main Theorem) (restatement)). *Assume the existence of a t_P -round $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$ -secure UC-puzzle in a \mathcal{G} -hybrid model, t_{WI} -round \mathcal{SNMWI} protocol secure w.r.t $\text{cl}(\mathcal{C}_{\text{sim}}, \mathcal{C}_{\text{env}})$ and t_{OT} -round semi-honest oblivious transfer protocol secure w.r.t $\text{cl}(\mathcal{C}_{\text{sim}}, \mathcal{C}_{\text{env}})$. Then, for every “well-formed” functionality \mathcal{F} , there exists a $O(t_P + t_{WI} + t_{OT})$ -round protocol Π in the \mathcal{G} -hybrid model that realizes $\hat{\mathcal{F}}$ with $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$ -UC-security.*

On a high-level, the compilation proceeds in two steps:

- First, every functionality is compiled into a protocol in the ZK-Hybrid model. In the ZK-Hybrid, all parties have access to the ideal zero-knowledge functionality called `IdealZK` functionality. This step is formalized in the `IdealZK`-lemma (Lemma 4).
- In the second step, assuming the existence of a UC-puzzle, a semi-honest oblivious transfer protocol and a \mathcal{SNMWI} protocol, we show that the `IdealZK` functionality can be securely realized in the real-model. This step is formalized in the Puzzle-lemma (Lemma 5).

Lemma 4 (`IdealZK`-Lemma). *Assume the existence of t -round stand-alone secure semi-honest oblivious transfer secure w.r.t $\text{cl}(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$. For every well-formed functionality \mathcal{F} , there exists a $O(t)$ -round protocol Π in the ZK-Hybrid model, such that, for every adversary $A \in \mathcal{C}_{\text{sim}}$ in the $\hat{\mathcal{F}}_{\text{IdealZK}}\text{-Hybrid}$ model, there exists an adversary simulator $A' \in \mathcal{C}_{\text{sim}}$, such that for every environment $Z \in \mathcal{C}_{\text{env}}$, the following two ensembles are indistinguishable w.r.t $\text{cl}(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$.*

- $\left\{ \text{EXEC}_{\Pi, A, Z}^{\hat{\mathcal{F}}_{\text{IdealZK}}}(n) \right\}_{n \in N}$
- $\left\{ \text{EXEC}_{\pi_{\text{ideal}}, A', Z}^{\hat{\mathcal{F}}}(n) \right\}_{n \in N}$

where $\hat{\mathcal{F}}$ is the ideal-functionality implementing the multi-session extension of f .

This lemma is implicit for the original UC model in the previous works [64, 8, 34, 19, 66]. The series of works shows that in the standard model (i.e. $(n.u.\mathcal{PPT}, \mathcal{PPT})$ -UC), assuming the existence of stand-alone secure semi-honest oblivious transfer, any well-formed functionality can be securely realized in the $\hat{\mathcal{F}}_{IdealZK}$ -Hybrid model. On a high-level, given a well-formed functionality \mathcal{F} , the compilation proceeds in three steps:

1. Construct a protocol $\Pi_{\mathcal{F}}^1$ that UC realizes \mathcal{F} in the \mathcal{F}_{OT} -hybrid model in the presence of semi-honest, static adversaries.
2. Assuming the existence of stand-alone secure semi-honest oblivious-transfer, construct a protocol Π_{OT} that UC-realizes \mathcal{F}_{OT} in the presence of semi-honest, static adversaries. Then using the protocol obtained from Step 1, obtain a protocol $\Pi_{\mathcal{F}}^2$ that UC-realizes \mathcal{F} in the presence of semi-honest, static adversaries.
3. Compile any protocol Π to Π' in the ZK-Hybrid, so that, for any malicious adversary A interacting with Π in ZK-Hybrid, there exists a semi-honest static adversary S that interacts with Π' , such that no environment can distinguish if it we interacting with A in ZK-Hybrid running Π' or S in the plain-life model running Π . Then using the protocol obtained from Step 2, obtain a protocol $\Pi_{\mathcal{F}}^3$ that is UC-secure in ZK-Hybrid.

We remark that we require the protocol obtained from Step 3 to be secure for \mathcal{C}_{env} -adversaries. This can be obtained by following the same steps as above, with the exception that the semi-honest oblivious transfer protocol must be secure w.r.t $cl(\mathcal{C}_{sim}, \mathcal{C}_{env})$.

The main technical contribution of our work is the following lemma:

Lemma 5 (Puzzle-Lemma). *Let Π' be a protocol in the $\hat{\mathcal{F}}_{IdealZK}$ -Hybrid model. Assume the existence of a $(\mathcal{C}_{env}, \mathcal{C}_{sim})$ -secure t_P -round puzzle $\langle S, R \rangle$ in a \mathcal{G} -hybrid model and a t_{OT} -round stand-alone malicious oblivious-transfer protocol $\langle S_{OT}, R_{OT} \rangle$ secure w.r.t $cl(\mathcal{C}_{sim}, \mathcal{C}_{env})$ and a t_{WI} -round t_{OT} -robust \mathcal{SNMWI} protocol $\langle P_s, V_s \rangle$ secure w.r.t $cl(\mathcal{C}_{sim}, \mathcal{C}_{env})$. Then, there exists a $O(t_P + t_{WI} + t_{OT})$ -round protocol Π in the \mathcal{G} -hybrid such that, for every uniform \mathcal{PPT} adversary A , there exists a simulator $A' \in \mathcal{C}_{sim}$, such that, for every environment $Z \in \mathcal{C}_{env}$, the following two ensembles are indistinguishable over N w.r.t \mathcal{C}_{sim} .*

- $\left\{ \text{EXEC}_{\Pi, A, Z}^{\mathcal{G}}(n) \right\}_{n \in N}$
- $\left\{ \text{EXEC}_{\Pi', A', Z}^{\hat{\mathcal{F}}_{\text{IdealZK}}}(n) \right\}_{n \in N}$

Before we proceed to proving the Puzzle Lemma we formally define the **IdealZK**-functionality and oblivious-transfer.

5.1 IdealZK Functionality

The notion of ideal functionalities is a central tool in the framework of universal composability [12], and can be thought of as the introduction of a trusted third party that is designed to perform a specific task. The communication with the trusted third party is specified as follows. All parties have a secret, authenticated and unblockable channel to the trusted third party through which they send their input. The trusted party computes the output, hands the output of the corrupted parties to the adversary, and asks the adversary to deliver the output to the honest parties. The adversary does not see the output destined to the honest party but may deliver it (or not), at will. One of the most basic and useful ideal functionalities in the design of cryptographic protocols is the ideal zero-knowledge proof of knowledge functionality.

THE IDEALZK FUNCTIONALITY: The ideal zero-knowledge functionality is parameterized by an **NP**-relation R_L and defined as follows:

- Upon receiving a message $(\text{ZK-Prover}, j, x, w)$ from a party P_i (called the prover), the functionality sends $(\text{ZK-Proof}, i, x, R_L(x, w))$ to party P_j (called the verifier). Unless the party P_j is corrupted, the functionality also sends the message $(\text{ZK-Proof}, i, x, R_L(x, w))$ to the adversary (controlling the network).

Remark 3. *Since we consider an asynchronous execution of protocols, formally, both the input and the output of the functionality should also contain a session identifier (as was done in [12]) not to mix up messages from different sessions. In order to simplify the notation (for this and other ideal functionalities that we use), we leave out the session identifier and instead assume that this information*

is added to all messages (sent to and from ideal functionalities) in some canonical way.

Thus, slightly over-simplified, the prover sends an instance-witness pair (x, w) and an index j to the ideal functionality, which in turn, sends $(x, 1)$ to the verifier (party P_j) if w is a valid witness for $x \in L$ and $(x, 0)$ otherwise. In order to model provers that fail in proving true instance (clearly, one can not force a prover to prove something that it does not want to), we define a special symbol \perp such that for every R_L and every x , it holds that $(x, \perp) \notin R_L$. The setting in which the parties have access to (multiple) ideal zero-knowledge functionalities is called the ZK-Hybrid model. The power of this model has been demonstrated in for example [19] where a universally composable two-party computation protocol was constructed.

5.2 Oblivious Transfer

Oblivious Transfer is generally modeled as a secure two-party computation task following the real-world/ideal-world paradigm. However, we do not require a protocol implementing oblivious transfer in the strongest setting and therefore present a definition that is sufficient for the protocols presented in this work. First, we define the oblivious transfer functionality.

Let $s_1, s_2 \in \{0, 1\}^n$ and $b \in \{1, 2\}$. Then the 1-out-of-2 string OT functionality $\text{OT}^n(\cdot)$ is defined as

$$\text{OT}^n((s_1, s_2), b) = (\perp, s_b)$$

REAL WORLD. In the real world, the Sender S and Receiver R execute the given protocol Π on common security parameter n and respective private inputs (s_1, s_2) and b , where $s_1, s_2 \in \{0, 1\}^n$ and $b \in \{1, 2\}$. A probabilistic polynomial-time adversary, who may corrupt one of the parties and observe all of its internal data. In the malicious case, the adversary has full control over the messages sent by the corrupted party while in the semi-honest or passive case, the adversary may only try to deduce information by performing computations on observed data, but otherwise follows the protocol's instructions. At the end of the interaction, the

adversary may output an arbitrary function of its view. The output of the real world (on the given pair of initial inputs) is defined as the random variable $\text{REAL}_{S,R}(n, s_1, s_2, b)$ the adversary's output.

IDEAL WORLD. In the ideal world, an incorruptible trusted party implementing the OT functionality is employed. That is, the “protocol” in the ideal world instructs each party to send its input to the trusted party, who computes the functionality and sends to each party its output. The interaction of the adversary with the ideal process and the output of the ideal process represented by the random variable $\text{IDEAL}_{S,R}^{\text{OT}}(n, s_1, s_2, b)$ and defined analogously to the above definitions for the real process. The adversary attacking the ideal process is referred to as a simulator.

Definition 18. *We say that a protocol $\langle S, R \rangle$ is a stand-alone secure semi-honest oblivious-transfer (SH-OT) protocol if the following conditions hold:*

1. *For every semi-honest PPT receiver R^* , there exists a PPT simulator R' such that the following distributions are indistinguishable over $n \in \mathbb{N}$:*
 - $\{\text{REAL}_{S,R^*}(n, s_1, s_2, b)\}_{n \in \mathbb{N}, s_1, s_2 \in \{0,1\}^n, b \in \{1,2\}}$
 - $\{\text{IDEAL}_{S,R'}^{\text{OT}}(n, s_1, s_2, b)\}_{n \in \mathbb{N}, s_1, s_2 \in \{0,1\}^n, b \in \{1,2\}}$
2. *For every semi-honest PPT sender S^* , there exists a PPT simulator S' such that the following distributions are indistinguishable over $n \in \mathbb{N}$:*
 - $\{\text{REAL}_{S^*,R}(n, s_1, s_2, b)\}_{n \in \mathbb{N}, s_1, s_2 \in \{0,1\}^n, b \in \{1,2\}}$
 - $\{\text{IDEAL}_{S',R}^{\text{OT}}(n, s_1, s_2, b)\}_{n \in \mathbb{N}, s_1, s_2 \in \{0,1\}^n, b \in \{1,2\}}$

We further say that $\langle S, R \rangle$ stand-alone secure malicious oblivious-transfer (m-OT) if both the conditions hold for all malicious adversaries.

Definition 19. *A stand-alone secure m-OT protocol $\langle S, R \rangle$ is said to be input-binding w.r.t the Sender if for every malicious Sender S^* and auxiliary input z and every view of the honest Receiver on any input b , i.e. $\text{view}_R[\langle S^*(z), R(b) \rangle]$ is consistent with at most one input (s_1, s_2) for an honest Sender. Similarly, $\langle S, R \rangle$ is said to be input-binding w.r.t the Receiver if for every malicious Receiver R^* and auxiliary input z and every view of the honest Sender on any input (s_1, s_2) , i.e. $\text{view}_S[\langle S(s_1, s_2), R^*(z) \rangle]$ is consistent with at most one input b for an honest*

Receiver. Finally, we say that $\langle S, R \rangle$ is **input-binding** if it is input-binding w.r.t the Sender and Receiver.

In our construction, we will require an input-binding stand-alone secure m-OT. However, due to the following result it suffices to assume the existence of stand-alone secure SH-OT.

Proposition 2 ([34, 31]). *Assume the existence of a t -round stand-alone secure SH-OT. Then, there exists an $O(t)$ -round input-binding stand-alone secure m-OT.*

The proof of the proposition is implicit in [31, 34], where they show how to transform any semi-honest secure computation protocol to one that is secure w.r.t malicious adversaries. The high-level idea is that in the transformed protocol, first, every party P_i commit to their inputs and randomness at the beginning of the protocol execution. Then using a fair coin-tossing protocol the parties obtain uniformly generated random tapes. Thereafter, all parties execute the semi-honest protocol with the exception that, after each step, every party P_i proves to every other party in zero-knowledge that the message generated is consistent with the value committed to at the beginning. Since the parties commit to their inputs and randomness, it will follow that the protocol is also input-binding.

EXISTENCE OF OBLIVIOUS TRANSFER PROTOCOLS. We know how to construct $O(1)$ -round stand-alone secure semi-honest oblivious transfer protocols based on enhanced trapdoor permutations, homomorphic encryption.

5.3 The Puzzle Lemma

In this section, we prove the puzzle lemma. More precisely, we provide a general transformation that transforms any protocol Π in the ZK-Hybrid model into a protocol Π' in the real model. First, we describe a special-purpose zero-knowledge protocol that enables this transformation.

SPECIAL-PURPOSE ZK PROTOCOL $\langle P, V \rangle$. Let $(\langle S, R \rangle, \mathcal{R})$ be a $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$ -secure puzzle in the \mathcal{G} -hybrid, $\langle P_s, V_s \rangle$ be a \mathcal{SNMWZ} protocol secure w.r.t $cl(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$ and $\langle S_{OT}, R_{OT} \rangle$ be 1-out-of-2 malicious string oblivious-transfer protocol secure w.r.t $cl(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$. Let L be a language in **NP** with witness relation R_L .

On common input instance x , witness relation R_L , prover and verifier identities id_P and id_V , and additional auxiliary input $w = R_L(x)$ for the prover, the protocol $\langle P, V \rangle$ proceeds in 5-stages as follows:

Stage 1: The Prover and Verifier participate in a puzzle-interaction where the Verifier assumes the role of the sender and the Prover as the receiver. Let $\text{TRANS}_{V \rightarrow P}$ be the transcript of the messages exchanged in this Stage.

Stage 2: The Prover and Verifier participate in a second puzzle-interaction with the roles reversed, i.e. the Prover is the sender and the Verifier is the receiver. Let $\text{TRANS}_{P \rightarrow V}$ be the transcript of the messages exchanged in this Stage.

Stage 3: The Prover first selects a random string $r \in \{0, 1\}^n$. Then the Prover and Verifier interact using $\langle S_{OT}, R_{OT} \rangle$, where the Prover is the sender with inputs $(r, r \oplus w)$ and the Verifier is the receiver with input 1. Let TRANS_{OT} be the transcript of the messages exchanged in this stage.

Stage 4: The Verifier commits to s using a perfectly binding commitment scheme com^1 . Then it proves using *two* $\langle P_s, V_s \rangle$ proofs in succession with identity id_V , the statement that it either committed to a string s that contains a valid witness establishing the verifiers input as index 1 in TRANS_{OT} and the string output by the receiver at the end of the Stage 3 protocol or a string s such that $(s, \text{TRANS}_{V \rightarrow P}) \in \mathcal{R}$.

Stage 5: The Prover sends the string r in the clear and commits to s using com . Then the prover proves using *two* $\langle P_s, V_s \rangle$ proofs in succession with id_P , the statement that it either committed to a string s that establishes that the inputs used by the prover in TRANS_{OT} is (r, r') such that $r \oplus r' \in R_L(x)$ or a string s such that $(s, \text{TRANS}_{P \rightarrow V}) \in \mathcal{R}$.

REALIZING THE IDEALZK-FUNCTIONALITY: Given any protocol Π' in $\hat{\mathcal{F}}_{\text{idealZK}}\text{-Hybrid}$ model and the special-purpose zero-knowledge protocol $\langle P, V \rangle$, the protocol Π in the real model is constructed from Π' by instantiating the $\hat{\mathcal{F}}_{\text{idealZK}}$ functionality using $\langle P, V \rangle$. All invocations of the $\hat{\mathcal{F}}_{\text{idealZK}}$ functionality with input $(\text{ZK-prover}, sid, ssid, P_j, x, w)$ from an honest party P_i is replaced with an instance of $\langle P, V \rangle$ between P_i and P_j on common inputs x, w , identities $\text{id}_P = (P_i, sid, ssid)$

¹For simplicity of exposition, we construct the protocol $\langle P, V \rangle$ using a perfectly binding commitment scheme, it is actually sufficient to use a statistically binding commitments.

and $\text{id}_V = (P_j, sid, ssid)$. To prove correctness, we need to show that for every adversary $A \in \mathcal{PPT}$ in the real-model, there exists a simulator $S \in \mathcal{C}_{\text{sim}}$ such that no environment $Z \in \mathcal{C}_{\text{env}}$ can distinguish if it is interacting with A in the real-model or S in the $\hat{\mathcal{F}}_{\text{IdealZK-Hybrid}}$.

Recall that in any $\langle P_s, V_s \rangle$ interaction (Stage 4 or Stage 5), a prover can alternatively use a “fake” witness which is a witness to the puzzle to convince the verifier. However, to prove correctness of our simulator, we require that no adversary controlling the prover be able to commit to the fake witness in any instance of $\langle P_s, V_s \rangle$ -subprotocol. For the remainder of the proof, still informally, we say that an adversary is *non-abusing* in an execution, if except with negligible probability, it never commits to a fake witness.

We construct the simulator in two steps.

Step 1: First, we consider a hybrid experiment H_n^0 where all the puzzle-interactions part of $\langle P, V \rangle$ are simulated. For this hybrid, we show that for every adversary $A \in \mathcal{C}_{\text{adv}}$ in the real-model, there exists a machine $A^* \in \mathcal{C}_{\text{sim}}$ such that no environment $Z \in \mathcal{C}_{\text{env}}$ can distinguish A in the real-model and A^* in H_n^0 . This machine A^* additionally outputs a valid witness for every puzzle interaction where the adversary controls the sender on a special-output tape. This step essentially follows from the definition of the puzzle by constructing an adversary using A that participates in a concurrent puzzle execution. Let $\text{HYBRID}_{A^*, Z}^0(n)$ denote the output of Z in experiment H_n^0 . More precisely, we establish the following lemma.

Lemma 6. *For every adversary $A \in \mathcal{PPT}$, there exists an adversary $A^* \in \mathcal{C}_{\text{sim}}$ such that for every environment $Z \in \mathcal{C}_{\text{env}}$, it holds that*

$$\left\{ \text{EXEC}_{\Pi, A, Z}^G(n) \right\}_{n \in N} \approx \left\{ \text{HYB}_{A^*, Z}^0(n)(Z) \right\}_{n \in N}$$

Furthermore, A^* is non-abusing in H_n^0 .

Step 2: Using A^* we construct S and prove that no environment Z can distinguish between A^* in the H_n^0 and S in the $\hat{\mathcal{F}}_{\text{IdealZK-Hybrid}}$. This step relies on the straight-line extractability of the special-purpose \mathcal{ZK} protocol $\langle P, V \rangle$ and the robustness \mathcal{SNMWI} property of the $\langle P_s, V_s \rangle$.

On a high-level, S internally incorporates A^* and emulates an execution with A^* . All messages from A^* are forwarded externally (to the respective entities) except messages that are part of any execution using $\langle P, V \rangle$, which are instead dealt with internally.

- In $\langle P, V \rangle$ -interactions where A^* controls the verifier, S simulates the prover messages internally for A^* . Recall that the honest prover uses the witness of the statement to generate inputs to the Stage 3 OT protocol and the Stage 5 \mathcal{SNMWI} -proof. As S does not possess a witness, it instead uses two random strings as input for the Stage 3 OT protocol and uses the fake witness in the Stage 5 \mathcal{SNMWI} -proof (the witness to the puzzle). We remark that the simulation is correct only if the verifier does not cheat in choosing the inputs for the Stage 3 OT protocol. This can be ensured if the string s committed to by A^* in the Stage 4 is not the fake witness.
- In executions where A^* controls the prover, S emulates the code of the honest verifier with the exception that A' runs the Stage 3 OT protocol with input index 2 instead of 1 and again uses a fake witness in the Stage 4 \mathcal{SNMWI} -proof. Finally, using the string s received from the prover in the Stage 3 OT protocol and string s' that it received in Stage 5 from the prover, A' computes $w = s \oplus s'$ as the witness for the statement and forwards it externally to the $\hat{\mathcal{F}}_{\text{idealZK}}$ -functionality. Note that the witness extracted by S is valid witness to the statement only if A^* does not cheat in choosing the inputs for the OT protocol. This can be ensured if the string s committed to by A^* in the Stage 5 is not the fake witness.

More precisely, in this step, we prove the following lemma.

Lemma 7. *For every adversary $A^* \in \mathcal{C}_{\text{sim}}$ that is non-abusing in H_n^0 , there exists a simulator $S \in \mathcal{C}_{\text{sim}}$ such that for every environment $Z \in \mathcal{C}_{\text{env}}$, it holds that*

$$\left\{ \text{HYBRID}_{A^*, Z}^0(n)(Z) \right\}_{n \in N} \approx \left\{ \text{EXEC}_{\Pi', S, Z}^{\hat{\mathcal{F}}_{\text{idealZK}}}(n) \right\}_{n \in N}$$

Furthermore, S is non-abusing in $\hat{\mathcal{F}}_{\text{idealZK}}\text{-Hybrid}$.

The puzzle lemma follows from Lemma 6 and Lemma 7 using a standard hybrid argument. Before we proceed to the actual proofs, we formalize the non-abusing property.

In an execution of $\langle P, V \rangle$, where an adversary controls either the prover or the verifier, we say that the event CHEAT occurs if:

- The adversary commits to w such that $w \in \mathcal{R}\text{TRANS}_{V \rightarrow P}$ in Stage 5 when controlling the prover, or,
- The adversary commits to w such that $w \in \mathcal{R}\text{TRANS}_{P \rightarrow V}$ in Stage 4 when controlling the verifier.

We say that an adversary is NON-ABUSING if the probability that CHEAT occurs for any $\langle P, V \rangle$ interaction where the adversary controls one of the parties is negligible.

5.3.1 Step 1: Simulating puzzle interactions

In the first hybrid experiment H_n^0 the execution proceeds identically to the real-execution, with the exception that, in Stage 1 and Stage 2 of all $\langle P, V \rangle$ -interactions, the parties instead of participating in the protocol to generate a puzzle, receive a complete (simulated) puzzle-transcript for each Stage from the adversary A^* . We show that for every adversary A in the real-model there exists an adversary A^* such that no environment can distinguish if it is interacting with A in the real-model or A^* in H_n^0 . Furthermore, for every puzzle interaction where the sender is controlled by the adversary,² A^* outputs a valid witness w corresponding to the puzzle-transcript in a *special-output tape*. The existence of A^* and the correctness of the simulation essentially follows from the concurrent simulatability of the puzzle $\langle S, R \rangle$.

Lemma 8 (Lemma 6 (restated)). *For every adversary $A \in \mathcal{PPT}$, there exists an adversary $A^* \in \mathcal{C}_{\text{sim}}$ such that for every environment $Z \in \mathcal{C}_{\text{env}}$, it holds that*

$$\left\{ \text{EXEC}_{\Pi, A, Z}^G(n) \right\}_{n \in N} \approx \left\{ \text{HYBRID}_{A^*, Z}^0(n)(Z) \right\}_{n \in N}$$

²These are Stage 1 puzzles for $\langle P, V \rangle$ -interactions where the adversary controls the verifier and Stage 2 puzzles for interactions where the adversary controls the prover

Furthermore, A^* is NON-ABUSING in H_n^0 .

Proof: We begin by describing an adversary $A_{\text{puz}} \in \mathcal{PPT}$ constructed from A that participates in a concurrent puzzle execution. Then using the simulator A'_{puz} corresponding to A_{puz} that exists from the definition of the puzzle, we construct A^* .

The adversary A_{puz} internally incorporates A and begins an emulation of an execution with A . All messages that are not part of puzzle interactions from A are forwarded to the external puzzle environment Z_{puz} . Every puzzle interaction where A assumes the role of the sender in the internal emulation, A_{puz} forwards the messages to and from an external receiver. For puzzle interactions with A as the receiver, A_{puz} emulates the interaction with A internally by running the code of an honest sender with A . For all puzzle interactions between honest parties, A_{puz} internally emulates a puzzle interaction by running the code of the receiver and sender. At the end of every puzzle interaction, A_{puz} forwards the puzzle-transcript to Z_{puz} . As this defines an adversary participating in a concurrent puzzle execution, there exists a simulator A'_{puz} that simulates all puzzle interactions and outputs witnesses for all puzzles where the adversary controls the sender. More precisely, A'_{puz} is such that, for every environment $Z_{\text{puz}} \in \mathcal{C}_{\text{env}}$, it holds that

$$\{\text{REAL}_{A_{\text{puz}}, Z_{\text{puz}}}(n)\}_{n \in N} \approx \{\text{IDEAL}_{A'_{\text{puz}}, Z_{\text{puz}}}(n)\}_{n \in N}$$

Given A'_{puz} , we describe the construction of A^* . A^* internally incorporates A'_{puz} and emulates the ideal experiment of a concurrent puzzle execution. Externally, A^* interacts with the honest parties and Z in the hybrid experiment H_n^0 . All messages received from external parties are fed internally to A'_{puz} and interpreted as coming from an internal emulation of Z_{puz} . Messages received from A'_{puz} in the internal emulation, on the other hand, are forwarded externally to the party *as intended* by A in the original execution with Z (we assume without loss of generality that the identity of the recipient is encoded in every message). All messages from A'_{puz} that contain puzzle-transcripts are forwarded to the corresponding honest parties. Finally, A^* outputs on its special output tape whatever A'_{puz} outputs on its special output tape.

PROVING INDISTINGUISHABILITY: Assume for contradiction, there exists an environment Z that violates the indistinguishability, i.e. there exists a distinguisher

D and polynomial $p(\cdot)$ such that for infinitely many n ,

$$\Pr[D(\text{EXEC}_{\Pi,A,Z}^G(n)) = 1] - \Pr[D(\text{HYBRID}_{A^*,Z}^0(n)) = 1] \geq \frac{1}{p(n)}$$

Using Z we construct an environment Z_{puz} that violates the security of the puzzle in a concurrent puzzle execution with A_{puz} and thus arrive at a contradiction. More precisely, consider an environment Z_{puz} that internally incorporates all the honest parties and Z in a real-world execution with adversary A . Z_{puz} internally begins an emulation with the parties and proceeds as follows: It forwards the messages it receives from A_{puz} internally to the corresponding honest party or Z (as encoded in the message). Finally, Z_{puz} outputs what Z outputs. Recall that, the puzzle-environment is allowed to interact only with the adversary and in particular cannot access \mathcal{G} . This is ensured in our construction of Z_{puz} , since only the puzzle interactions access \mathcal{G} and all those are emulated internally by A_{puz} .

Claim 5.

$$\Pr[D(\text{REAL}_{A_{\text{puz}}, Z_{\text{puz}}}(n)) = 1] - \Pr[D(\text{IDEAL}_{A'_{\text{puz}}, Z_{\text{puz}}}(n)) = 1] \geq \frac{1}{p(n)}$$

Proof: By construction of Z_{puz} and A_{puz} , it directly follows that:

1. The output of Z in the internal emulation by Z_{puz} when interacting with A_{puz} is identically distributed to $\text{EXEC}_{\Pi,A,Z}^G(n)$.
2. The output of Z in the internal emulation by Z_{puz} when interacting with A'_{puz} is identically distributed to $\text{HYBRID}_{A^*,Z}^0(n)$.

The claim now follows from the fact that D distinguishes $\text{EXEC}_{\Pi,A,Z}^G(n)$ and $\text{HYB}_{A^*,Z}^0(n)$ with probability at least $\frac{1}{p(n)}$. \square

We conclude the proof of indistinguishability by observing that Claim 5 violates the (statistical-)simulatability of the puzzle protocol $\langle S, R \rangle$ and thus arrive at a contradiction.

PROVING A^* IS NON-ABUSING IN H_n^0 : First, we establish that A is NON-ABUSING in the real-world experiment. We then reduce the NON-ABUSING property of A^* in H_n^0 to the statistical-simulatability of the $\langle S, R \rangle$ protocol.

Claim 6. *A* is NON-ABUSING in the real-world experiment.

Proof: Assume for contradiction there exists an adversary $A \in \mathcal{PPT}$ and polynomial $p(\cdot)$ such that, for infinitely many n , in the $\text{EXEC}_{\Pi, A, Z}^G(n)$, A commits to the fake witness corresponding to some instance of the $\langle P_s, V_s \rangle$ -protocol (i.e. com message in Stage 4 or 5 of a $\langle P, V \rangle$ -interaction) with probability at least $\frac{1}{p(n)}$. We construct a cheating receiver $\tilde{R} \in \mathcal{C}_{\text{adv}}$ using A that violates the soundness condition of the puzzle and arrive at a contradiction.

More formally, \tilde{R} on input n proceeds as follows: It incorporates A , Z and all the honest parties and emulates the experiment $\text{EXEC}_{\Pi, A, Z}^G(n)$ in the following manner:

- For a randomly chosen puzzle interaction where the adversary A controlling the receiver interacts with a sender controlled by an honest party, \tilde{R} forwards messages in the $\langle S, R \rangle$ -interaction to the external sender S .
- Once the execution has concluded, \tilde{R} applies the (stand-alone) extractor guaranteed by the proof-of-knowledge property of $\langle P_s, V_s \rangle$ on the Stage 4 or 5 proofs (as the case may be) to extract a witness w . If $w \in \mathcal{R}(\text{TRANS})$, where TRANS is the transcript of the $\langle S, R \rangle$ -interaction with the external sender, then \tilde{R} outputs w and halts. Otherwise, \tilde{R} outputs \perp .

Since there are only polynomially many puzzles and $\langle P_s, V_s \rangle$ interactions, \tilde{R} chooses the $\langle P_s, V_s \rangle$ interaction corresponding to which A commits to the fake witness with non-negligible probability. By the proof of knowledge property of the $\langle P_s, V_s \rangle$ proof and the statistical-binding property of the com scheme, it follows that, if the string committed to by A in the proof is a witness to the puzzle, then except with negligible probability, \tilde{R} extracts w such that $w \in \mathcal{R}(\text{TRANS})$. Therefore \tilde{R} outputs a valid witness to the puzzle with non-negligible probability and this violates the soundness condition of the $\langle S, R \rangle$ protocol against \mathcal{PPT} adversaries and we arrive at a contradiction. This concludes the proof of the claim. \square

As in the proof of indistinguishability, we reduce the proof of the claim to the statistical-simulatability of the $\langle S, R \rangle$ protocol. Assume for contradiction, for some A and A^* described above, there exists a polynomial $p(\cdot)$ such that, for infinitely many n , A^* commits to a fake witness with probability $\frac{1}{p(n)}$ in H_n^0 .

Fix an n for which this happens. Consider $\overline{Z_{\text{puz}}}$ that proceeds identically to Z_{puz} (described above), with the exception that it outputs a randomly chosen puzzle-transcript and the corresponding commitment following the puzzle interaction. Let $q(n)$ be a bound on the total number of $\langle P_s, V_s \rangle$ -interactions. Consider a distinguisher D that on input $(\text{TRANS}, \text{com}(w))$, computes w using exhaustive search and outputs 1 if $w \in \mathcal{R}(\text{TRANS})$ and 0 otherwise. Note that we allow D to run unbounded time, since we are interested only in violating statistical simulability.

Claim 7.

$$\Pr[D(\text{REAL}_{A_{\text{puz}}, Z_{\text{puz}}}(n)) = 1] - \Pr[D(\text{IDEAL}_{A'_{\text{puz}}, Z_{\text{puz}}}(n)) = 1] \geq \frac{1}{p(n)q(n)} - \nu(n)$$

for some negligible function $\nu(\cdot)$.

Proof: By construction of $\overline{Z_{\text{puz}}}$ and A_{puz} , it directly follows that: the internal emulation by $\overline{Z_{\text{puz}}}$ when interacting with A_{puz} proceeds identically to the real-world experiment. Similarly, the internal emulation by $\overline{Z_{\text{puz}}}$ when interacting with A'_{puz} proceeds identically to the experiment H_n^0 . It now follows that D on input $\text{IDEAL}_{A'_{\text{puz}}, Z_{\text{puz}}}(n)$ outputs 1 if

1. A^* commits to the fake witness corresponding to some $\langle P_s, V_s \rangle$ instance. This happens with probability $\frac{1}{p(n)}$.
2. $\overline{Z_{\text{puz}}}$ picks the instance on which A^* commits to fake witness. This occurs with probability at least $\frac{1}{q(n)}$.
3. Given the output $(\text{TRANS}, \text{com}(\sigma))$ of $\overline{Z_{\text{puz}}}$, D computes σ correctly. Since com is a statistically-binding commitment, it holds that, except with probability $\nu_1(n)$ for some negligible function $\nu_1(\cdot)$, D computes an unique w corresponding to $\text{com}(w)$.

Therefore,

$$\Pr[D(\text{IDEAL}_{A'_{\text{puz}}, Z_{\text{puz}}}(n)) = 1] \geq \frac{1}{p(n)q(n)} - \nu_1(n)$$

It follows from Claim 6 that

$$\Pr[D(\text{REAL}_{A_{\text{puz}}, Z_{\text{puz}}}(n)) = 1] \leq \nu_2(n)$$

for some negligible function $\nu_2(\cdot)$.

The proof of the claim now follows from the preceding two equations. \square

Claim 7 now implies that D and $\overline{Z_{\text{puz}}}$ violate the statistical-simulatability of the puzzle protocol $\langle S, R \rangle$ and thus we arrive at a contradiction. This concludes the proof that A^* is NON-ABUSING in H_n^0 . We remark that this is the only place where we require that the puzzle is *statistically simulatable*.

This concludes the proof of Lemma 6 and Step 1 of the proof. \square

5.3.2 Step 2: Simulating Zero-Knowledge

DESCRIPTION OF THE SIMULATOR S : S incorporates A^* , and internally emulates an execution with A^* in the following manner: All messages exchanged with A^* that are *not* part of the protocol $\langle P, V \rangle$ are forwarded outside to respective parties. Only messages exchanged as part of interactions using $\langle P, V \rangle$ are dealt with internally. More precisely, when party P_i wishes to prove a statement x to P_j , S proceeds as follows:

Case: Prover (P_i) is honest and Verifier (P_j) is controlled by S . S (controlling party P_j) receives a message of the type $(\text{ZK-Proof}, i, x, 1)$ from $\hat{\mathcal{F}}_{\text{idealZK}}$ and starts simulating messages for P_i internally. Initially, A^* sends a puzzle-transcript $\text{TRANS}_{V \rightarrow P}$ for Stage 1 and $\text{TRANS}_{P \rightarrow V}$ for Stage 2 to P_i and outputs a witness w' (corresponding to the Stage 1 transcript) in the special output tape, which is stored by S internally. In Stage 3, S runs the code of an honest sender for the OT protocol with two randomly chosen strings s_0, s_1 as input. In Stage 4, S runs the code of the honest verifier for the $\langle S, R \rangle$ protocol. In Stage 5, S sends s_0 and then convinces A^* (acting as P_j) using the fake witness w' that it received from A^* . More precisely, it commits to w' and then proves using $\langle S, R \rangle$ that $w' \in \mathcal{R}(\text{TRANS}_{V \rightarrow P})$.

Case: Prover (P_i) is controlled by A and Verifier (P_j) is honest. S (controlling party P_i) starts simulating the verifier messages for P_j internally. As before, A^* sends a puzzle-transcript $\text{TRANS}_{V \rightarrow P}$ for Stage 1 and $\text{TRANS}_{P \rightarrow V}$ for Stage 2 to P_i and outputs a witness w' (corresponding to the Stage 2

transcript) in the special output tape. In Stage 3, S (acting as P_j) runs the code of an honest receiver for the OT protocol with input 2 (instead of 1) and receives a string r' . In Stage 4, S convinces A^* (acting as P_i) using the fake witness w' that it received from A^* . In Stage 5, after receiving the string r , S runs to code of the honest verifier of the $\langle S, R \rangle$ protocol. On successful completion, S computes $w = r \oplus r'$ and checks if w is a valid witness for x . If it is valid S sends $(\text{ZK-Prover}, j, x, w)$ to the $\hat{\mathcal{F}}_{\text{IdealZK}}$ functionality. Otherwise S outputs \perp and halts.

Case: Prover (P_i) is honest and Verifier (P_j) is honest. S (controlling the network) receives the message $(\text{ZK-Prover}, P_i, P_j, sid, ssid, x, 1)$ from the IdealZK functionality. Internally S , again receives a puzzle-transcript TRANS from A^* . Since, we are in the secure channels model, to simulate the interaction between honest parties to A^* , S merely sends null messages of appropriate length for each message passed in $\langle P, V \rangle$.

We now proceed to prove correctness of simulation. This is formalized in the following lemma.

Lemma 9 (Lemma 7 (restated)). *For every adversary $A^* \in \mathcal{C}_{\text{sim}}$ that is NON-ABUSING in H_n^0 , there exists a simulator $S \in \mathcal{C}_{\text{sim}}$ such that for every environment $Z \in \mathcal{C}_{\text{env}}$, it holds that*

$$\left\{ \text{HYBRID}_{A^*, Z}^0(n) \right\}_{n \in N} \approx \left\{ \text{EXEC}_{\Pi^*, S, Z}^{\hat{\mathcal{F}}_{\text{IdealZK}}}(n) \right\}_{n \in N}$$

Towards proving the lemma we consider the following intermediate hybrid experiments:

Experiment H_n^1 : This experiment proceeds identically to H_n^0 with the exception that in all $\langle P_s, V_s \rangle$ interactions in which the prover is honest, the fake witness (i.e. witness to the puzzle) is used to convince the verifier instead of the real witness. For this experiment, we prove the following claim.

Claim 8. *For every adversary $A^* \in \mathcal{C}_{\text{sim}}$ that is NON-ABUSING in H_n^0 , and environment $Z \in \mathcal{C}_{\text{env}}$, it holds that*

$$\left\{ \text{HYBRID}_{A^*, Z}^0(n) \right\}_{n \in N} \approx \left\{ \text{HYBRID}_{A^*, Z}^1(n) \right\}_{n \in N}$$

Furthermore, A^ is NON-ABUSING in H_n^1 .*

Experiment H_n^2 : H_n^2 proceeds identically to H_n^1 with the exception that in all $\langle S_{OT}, R_{OT} \rangle$ interactions where the verifier is honest, the receivers inputs to the Stage 3 OT protocol is replaced from index 1 to index 2. For this experiment, we prove the following claim.

Claim 9. *For every adversary $A^* \in \mathcal{C}_{\text{sim}}$ that is NON-ABUSING in H_n^1 and environment $Z \in \mathcal{C}_{\text{env}}$, it holds that*

$$\left\{ \text{HYBRID}_{A^*, Z}^1(n) \right\}_{n \in N} \approx \left\{ \text{HYBRID}_{A^*, Z}^2(n) \right\}_{n \in N}$$

Furthermore, A^ is NON-ABUSING in H_n^2 .*

Experiment H_n^3 : H_n^3 proceeds identically to H_n^2 with the exception that, in all $\langle S_{OT}, R_{OT} \rangle$ interactions where the prover is honest, the senders inputs in the Stage 3 OT protocol is replaced by randomly strings r_1, r_2 . For this experiment, we prove the following claim.

Claim 10. *For every adversary $A^* \in \mathcal{C}_{\text{sim}}$ that is NON-ABUSING in H_n^2 and environment $Z \in \mathcal{C}_{\text{env}}$, it holds that*

$$\left\{ \text{HYBRID}_{A^*, Z}^2(n) \right\}_{n \in N} \approx \left\{ \text{HYBRID}_{A^*, Z}^3(n) \right\}_{n \in N}$$

Furthermore, A^ is NON-ABUSING in H_n^3 .*

Recall that the simulator S in the $\hat{\mathcal{F}}_{\text{idealZK}}\text{-Hybrid}$ incorporates A^* and all the honest parties and emulates the experiment H_n^3 . It follows from the description that conditioned on A^* not proving a false statement, the view of A^* internally emulated is identical to the view of A^* in H_n^3 . Hence, if A^* is NON-ABUSING in H_n^3 , we have that

$$\left\{ \text{HYBRID}_{A^*, Z}^3(n) \right\}_{n \in N} \approx \left\{ \text{EXEC}_{\Pi', S, Z}^{\hat{\mathcal{F}}_{\text{idealZK}}}(n) \right\}_{n \in N}$$

The proof of Lemma 7 follows combining the above equation with Claims 8-10 using a standard hybrid argument.

COMPARING H_n^0 AND H_n^1 . Consider a sequence of intermediate experiments $E_n^1, \dots, E_n^{q(n)}$ where $q(\cdot)$ is a polynomial bounding the maximum number of proofs by the adversary A^* . E_n^i is the experiment with A^* that proceeds identically to H_n^0 , with the exception that in the first i proofs using $\langle P, V \rangle$, where an honest party

interacts with the adversary A^* , the honest party uses the fake witness of the real witness in the $\langle P_s, V_s \rangle$ sub-protocols. Recall that the fake witness is a witness to the puzzle transcript and is output by A^* in a special output tape. More precisely, in experiment E_n^i , the first i proofs using $\langle P, V \rangle$, the honest party is simulated identical to the experiment H_n^0 with the following exceptions:

- If the prover in the $\langle P, V \rangle$ -interaction is honest, the prover commits to fake witness (i.e. the witness of the puzzle-transcript $\text{TRANS}_{V \rightarrow P}$) in Stage 5 and convinces A^* in the $\langle P_s, V_s \rangle$ using the fake witness.
- If the verifier is honest, then the verifier commits to fake witness (i.e. the witness of the puzzle-transcript $\text{TRANS}_{P \rightarrow V}$) in Stage 4 and convinces A^* in the $\langle P_s, V_s \rangle$ using the fake witness.

We restate Claim 8 followed by the proof.

Claim 11 (Claim 8 restated). *For every adversary $A^* \in \mathcal{C}_{\text{sim}}$ that is NON-ABUSING in H_n^0 , and environment $Z \in \mathcal{C}_{\text{env}}$, it holds that*

$$\left\{ \text{HYBRID}_{A^*, Z}^0(n) \right\}_{n \in N} \approx \left\{ \text{HYBRID}_{A^*, Z}^1(n) \right\}_{n \in N}$$

Furthermore, A^* is NON-ABUSING in H_n^1 .

Proof: From the description it follows that the experiment $E_n^{q(n)}$ is identical to H_n^1 . Let E_n^0 denote the experiment H_n^0 . For hybrid experiment E_n^i , define random variables $\text{HYB}_{A^*, Z}^i(n)$ and $\text{WIT}_{A^*, Z}^i(n)$ to be respectively the output of Z and the value committed to by A^* corresponding to a randomly chosen $\langle P_s, V_s \rangle$ interaction where A^* controls the prover (the value is well-defined since the non-interactive commitment scheme com is perfectly-binding).

Assume for contradiction that there exists an adversary A^* that is NON-ABUSING in F_n^0 , distinguisher D and polynomial $p(\cdot)$ such that for infinitely many n , either of the following holds:

$$\begin{aligned} \Pr [D(\text{HYB}_{A^*, Z}^0(n)) = 1] - \Pr [D(\text{HYB}_{A^*, Z}^{q(n)}(n)) = 1] &\geq \frac{1}{p(n)}, \text{ or} \\ \Pr [\text{WIT}_{A^*, Z}^{q(n)}(n) \text{ is a fake witness}] &\geq \frac{1}{p(n)} \end{aligned}$$

Then there exists a function $i(\cdot)$ such that $i(n) \in [q(n)]$ for all n , such that for infinitely many n , either of the following holds:

$$\Pr \left[D(\text{HYB}_{A^*,Z}^{i(n)}(n)) = 1 \right] - \Pr \left[D(\text{HYB}_{A^*,Z}^{i(n)-1}(n)) = 1 \right] \geq \frac{1}{p(n)q(n)} \quad (5.1)$$

$$\begin{aligned} & \Pr \left[\text{WIT}_{A^*,Z}^{i(n)}(n) \text{ is a fake witness} \right] \\ & \quad - \Pr \left[\text{WIT}_{A^*,Z}^{i(n)-1}(n) \text{ is a fake witness} \right] \geq \frac{1}{p(n)q(n)} \end{aligned} \quad (5.2)$$

Fix a particular n for which that happens and let $i = i(n)$. Using A^* , we construct a man-in-the-middle adversary \tilde{A}^* and distributions D_n^1 and D_n^2 that violate the \mathcal{SNMWI} property of the $\langle P_s^2, V_s^2 \rangle$ protocol. Recall that in Stage 4 and 5, $\langle P_s, V_s \rangle$ protocol is repeated sequentially twice. For the proof of this claim, we consider the execution in Stage 4 and 5 after the commitment message as an interaction using $\langle P_s^2, V_s^2 \rangle$. Furthermore, since $\langle P_s, V_s \rangle$ is \mathcal{SNMWI} , using Proposition 1 it follows that $\langle P_s^2, V_s^2 \rangle$ is also \mathcal{SNMWI} .

First, note that experiments E_n^{i-1} and E_n^i proceed identically up until the end of Stage 4 of the i^{th} proof if the prover is honest in the i^{th} proof and until the end of Stage 3 if the verifier is honest in the i^{th} proof. Let $\Gamma(A^*, Z, n)$ denote the set of all possible joint views τ of A^* and all the parties such that the next message is the beginning of either a Stage 4 or Stage 5 message depending on which party is honest in the i^{th} proof using $\langle P, V \rangle$. Let $\overline{\text{WIT}}_{A^*,Z}^1(n)$ and $\overline{\text{WIT}}_{A^*,Z}^2(n)$ denote random variables that represent the value committed to by value committed to by A^* in a randomly chosen $\langle P_s^2, V_s^2 \rangle$ interaction that begins after the messages in τ are exchanged in experiments E_n^{i-1} and E_n^i respectively. It now follows that Equation 5.2 continues to hold, even if we replace $\text{WIT}_{A^*,Z}^{i-1}(n)$ and $\text{WIT}_{A^*,Z}^i(n)$ to $\overline{\text{WIT}}_{A^*,Z}^1(n)$ and $\overline{\text{WIT}}_{A^*,Z}^2(n)$ respectively. This is because, if the com message corresponding to a $\langle P_s^2, V_s^2 \rangle$ proof occurs in the messages in τ , then the probability that the value in the commitment is fake is identical in experiments E_n^{i-1} and E_n^i and thus can be safely ignored. Furthermore, we combine the case when Equation 5.1 or Equation 5.2 holds by constructing a distinguisher \tilde{D} that achieves the following in either of the two cases.

$$\begin{aligned} & \Pr \left[\tilde{D}(\text{HYB}_{A^*,Z}^i(n), \overline{\text{WIT}}_{A^*,Z}^1(n)) = 1 \right] \\ & \quad - \Pr \left[\tilde{D}(\text{HYB}_{A^*,Z}^{i-1}(n), \overline{\text{WIT}}_{A^*,Z}^2(n)) = 1 \right] \geq \frac{1}{p(n)q(n)} \end{aligned} \quad (5.3)$$

Such a distinguisher exists when Equation 5.1 holds; consider \tilde{D} that ignores the second input and runs D on the first input. When Equation 5.2 holds, consider a distinguisher \tilde{D} that merely checks and outputs 1 if and only if the second input is a witness to a puzzle interaction (i.e. fake witness).

Define distributions D_n^1 and D_n^2 as follows: Sample τ according to $\Gamma(A^*, Z, n)$ and,

- If in τ , A^* is controlling the prover in the i^{th} proof, then D_n^1 outputs

$$((x :: \text{TRANS}_{V \rightarrow P} :: \text{TRANS}_{OT} :: r, \text{com}(w, s)), (w, s), \tau,)$$

and D_n^2 outputs

$$((x :: \text{TRANS}_{V \rightarrow P} :: \text{TRANS}_{OT} :: r, \text{com}(w', s)), (w', s), \tau)$$

where corresponding to the i^{th} proof, x is the statement that the honest prover is proving in the $\langle P, V \rangle$ interaction, $w \in R_L(x)$, $\text{TRANS}_{V \rightarrow P}$ and TRANS_{OT} are transcripts of Stages 1 and 3, r is the input revealed by the honest prover in Stage 5 and $w' \in \mathcal{R}(\text{TRANS}_{V \rightarrow P})$.

- If in τ , A^* is controlling the verifier in the i^{th} proof, then D_n^1 outputs

$$((\text{TRANS}_{P \rightarrow V} :: \text{TRANS}_{OT}, \text{com}(w, r)), (w, r), \tau)$$

and $D_n^2(\tau)$ outputs

$$((\text{TRANS}_{P \rightarrow V} :: \text{TRANS}_{OT}, \text{com}(w', r)), (w', r), \tau)$$

where corresponding to the i^{th} proof, TRANS_{OT} and $\text{TRANS}_{P \rightarrow V}$ are the transcripts of Stage 2 and 3 and w is the witness used by the honest verifier in Stage 4, $w' \in \mathcal{R}(\text{TRANS}_{P \rightarrow V})$.

For a partial transcript τ sampled from $\Gamma(A^*, Z, n)$, define $\text{id}(\tau)$ to be the identifier of the honest party participating in the i^{th} proof.

The adversary \tilde{A}^* proceeds as follows on input $((\text{TRANS}, \text{com}(\sigma)), \tau)$. It incorporates A^* , Z and all the parties and internally emulates the experiment E_n^{i-1} in the following manner.

- It starts by feeding the parties all messages in τ .
- For the i^{th} proof, if the prover is honest, \tilde{A}^* feeds $\text{com}(\sigma)$ as the commitment in Stage 5 to A^* and forwards messages in the $\langle P_s^2, V_s^2 \rangle$ -interaction to the external prover (on the left).
- For a randomly chosen $\langle P_s^2, V_s^2 \rangle$ -interaction that begins after the messages in τ and A^* controls the prover, \tilde{A}^* forwards the messages to the external verifier (on the right).
- On completion of the internal emulation , \tilde{A}^* outputs what Z outputs in the internal emulation.

It follows from the description that the interaction emulated internally by \tilde{A}^* is identical to E_n^{i-1} when the inputs are chosen from D_n^1 and identical to E_n^i when the inputs are chosen from D_n^2 . Since \tilde{A}^* outputs what Z outputs, it follows that

$$\begin{aligned} (\text{HYB}_{A^*,Z}^{i-1}(n), \overline{\text{WIT}}_{A^*,Z}^1(n)) &= \left\{ (x, y, z) \leftarrow D_n^1 : \text{mim}_{\langle P_s^2, V_s^2 \rangle}^{\tilde{A}^*}(x, z, \text{id}(z)) \right\} \\ (\text{HYB}_{A^*,Z}^i(n), \overline{\text{WIT}}_{A^*,Z}^2(n)) &= \left\{ (x, y, z) \leftarrow D_n^2 : \text{mim}_{\langle P_s^2, V_s^2 \rangle}^{\tilde{A}^*}(x, z, \text{id}(z)) \right\} \end{aligned}$$

Using Equation 5.3, it follows that:

$$\begin{aligned} \Pr \left[(x, y, z) \leftarrow D_n^1 : \tilde{D}(\text{mim}_{\langle P_s^2, V_s^2 \rangle}^{\tilde{A}^*}(x, z, \text{id}(z))) = 1 \right] \\ - \Pr \left[(x, y, z) \leftarrow D_n^1 : \tilde{D}(\text{mim}_{\langle P_s^2, V_s^2 \rangle}^{\tilde{A}^*}(x, z, \text{id}(z))) = 1 \right] \geq \frac{1}{p(n)q(n)} \end{aligned}$$

Therefore, \tilde{D} distinguishes the man-in-the-middle experiments with \tilde{A}^* when the inputs are chosen according to D_n^1 and D_n^2 . Furthermore, since the com scheme is computationally-hiding , we have that the following distributions are indistinguishable over $n \in \mathbb{N}$:

- $\{((\text{TRANS}, \text{com}(\sigma)), y, z) \leftarrow D_n^1 : ((\text{TRANS}, \text{com}(\sigma)), z)\}_{n \in \mathbb{N}}$
- $\{((\text{TRANS}, \text{com}(\sigma')), y, z) \leftarrow D_n^2 : ((\text{TRANS}, \text{com}(\sigma')), z)\}_{n \in \mathbb{N}}$

Therefore, \tilde{A}^* with D_n^1 and D_n^2 violates the \mathcal{SNMWI} property of $\langle P_s^2, V_s^2 \rangle$ and we arrive at a contradiction. This concludes the proof of the claim. \square

COMPARING H_n^1 AND H_n^2 . Again, we consider a sequence of intermediate experiments $F_n^0 = H_n^1, F_n^1, \dots, F_n^{q(n)}$ where F_n^i is the experiment that proceeds identical

to H_n^1 , with the exception that in the first i proofs using $\langle P, V \rangle$ where honest parties receive a proof from A^* , the honest verifier is simulated so as to use index 2 instead of index 1 as input to the Stage 3 OT protocol. We restate Claim 9 and then provide the proof.

Claim 12 (Claim 9 restated). *For every adversary $A^* \in \mathcal{C}_{\text{sim}}$ that is NON-ABUSING in H_n^1 and environment $Z \in \mathcal{C}_{\text{env}}$, it holds that*

$$\left\{ \text{HYBRID}_{A^*, Z}^1(n) \right\}_{n \in N} \approx \left\{ \text{HYBRID}_{A^*, Z}^2(n) \right\}_{n \in N}$$

Furthermore, A^* is NON-ABUSING in H_n^2 .

Proof: By construction, the experiment $F_n^{q(n)}$ is identical to H_n^2 . We also have that the experiments F_n^i and F_n^{i-1} proceed identically up until the end of Stage 2 in the i^{th} proof using $\langle P, V \rangle$. Let $\Gamma(A^*, z, n)$ denote the set of all possible joint views τ of A^* and all the parties such that the next message is the beginning of Stage 3 in the i^{th} proof using $\langle P, V \rangle$ where the prover is honest. Define random variables for F_n^i , $\text{HYB}_{A^*, Z}^i(n)$ and $\text{WIT}_{A^*, Z}^i(n)$ to be respectively the output of Z and the value committed to by A^* corresponding to a randomly chosen $\langle P_s, V_s \rangle$ interaction where A^* controls the prover. Assume for contradiction that there exists an adversary A^* that is NON-ABUSING in experiment F_n^0 , distinguisher D and polynomial $p(\cdot)$ such that for infinitely many n , either of the following holds:

$$\begin{aligned} \Pr[D(\text{HYB}_{A^*, Z}^0(n)) = 1] - \Pr[D(\text{HYB}_{A^*, Z}^{q(n)}(n)) = 1] &\geq \frac{1}{p(n)} \\ \Pr[\text{WIT}_{A^*, Z}^{q(n)}(n) \text{ is a fake witness}] &\geq \frac{1}{p(n)} \end{aligned}$$

Then, there exists a function $i(\cdot)$ such that $i(n) \in [q(n)]$ for all n .

$$\begin{aligned} \Pr[D(\text{HYB}_{A^*, Z}^{i(n)}(n)) = 1] - \Pr[D(\text{HYB}_{A^*, Z}^{i(n)-1}(n)) = 1] &\geq \frac{1}{p(n)q(n)} \\ \Pr[\text{WIT}_{A^*, Z}^{i(n)}(n) \text{ is a fake witness}] \\ - \Pr[\text{WIT}_{A^*, Z}^{i(n)-1}(n) \text{ is a fake witness}] &\geq \frac{1}{p(n)q(n)} \end{aligned}$$

Just as in hybrids H_n^0 and H_n^1 , we can construct distinguisher \tilde{D} such that for infinitely many n , it holds that:

$$\begin{aligned} \Pr[\tilde{D}(\text{HYB}_{A^*, Z}^{i(n)}(n), \overline{\text{WIT}}_{A^*, Z}^1(n)) = 1] \\ - \Pr[\tilde{D}(\text{HYB}_{A^*, Z}^{i(n)-1}(n), \overline{\text{WIT}}_{A^*, Z}^2(n)) = 1] &\geq \frac{1}{p(n)q(n)} \end{aligned} \quad (5.4)$$

where $\overline{\text{WIT}}_{A^*,Z}^1(n)$ and $\overline{\text{WIT}}_{A^*,Z}^2(n)$ denote random variables that represent the value committed to by value committed to by A^* in a randomly chosen $\langle P_s, V_s \rangle$ interaction that begins after the messages in τ are exchanged in experiments F_n^{i-1} and F_n^i respectively.

Fix a particular n for which that happens and let $i = i(n)$. Using A^* , we construct a man-in-the-middle adversary \tilde{A}^* and machine B that violate the robust- \mathcal{SNMWI} property of the $\langle P_s, V_s \rangle$ protocol.

The machine B on input $v_n^b = (b, n)$, proceeds as follows: it runs the code of an honest receiver for the $\langle S_{OT}, R_{OT} \rangle$ protocol and interacts with external adversary \tilde{A}^* with input index b .

The adversary \tilde{A}^* on auxiliary input τ , proceeds as follows: It incorporates A^* , Z and all the parties and internally emulates the experiment F_n^{i-1} in the following manner.

- It starts by feeding the parties all messages in τ .
- For the i^{th} proof, \tilde{A}^* forwards messages in the Stage 3 OT protocol to the external machine B (on the left).
- For a randomly chosen $\langle P_s, V_s \rangle$ -interaction that begins after the messages in τ and A^* controls the prover, \tilde{A}^* forwards the messages to the external verifier (on the right).
- On completion of the internal emulation, \tilde{A}^* outputs what Z outputs in the internal emulation.

It follows from the description that the interaction emulated internally by \tilde{A}^* when the auxiliary input z is chosen according to the distribution $\Gamma(A^*, Z, n)$ is identical to F_n^{i-1} when B 's input is v_n^1 and F_n^i when B 's input is v_n^2 . Therefore, it follows that

$$\begin{aligned} (\text{HYB}_{A^*,Z}^{i-1}(n), \overline{\text{WIT}}_{A^*,Z}^1(n)) &= \text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}^*, B}(v_n^1) \\ (\text{HYB}_{A^*,Z}^i(n), \overline{\text{WIT}}_{A^*,Z}^2(n)) &= \text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}^*, B}(v_n^2) \end{aligned}$$

Now it follows from Equation 5.4 that:

$$\Pr \left[\tilde{D}(\text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}^*, B}(v_n^1)) = 1 \right] - \Pr \left[\tilde{D}(\text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}^*, B}(v_n^2)) = 1 \right] \geq \frac{1}{p(n)q(n)}$$

Since the OT protocol is receiver private, we additionally have that:

$$\left\{ \text{view}_{\tilde{A}}[\langle B(v_n^1), \tilde{A}(z) \rangle] \right\}_{n \in N} \approx \left\{ \text{view}_{\tilde{A}}[\langle B(v_n^2), \tilde{A}(z) \rangle] \right\}_{n \in N}$$

Since $\langle P_s, V_s \rangle$ is t_{OT} -robust and B interacts in at most t_{OT} rounds, we have that \tilde{A}^* , B and \tilde{D} violates the t_{OT} -robustness of the $\langle P_s, V_s \rangle$ and arrive at a contradiction. This concludes the proof of the claim. \square

COMPARING H_n^2 AND H_n^3 . Again, we consider a sequence of hybrid experiments $H_n^2 = G_n^0, G_n^1, \dots, G_n^{q(n)}$ where G_n^i is a hybrid experiment with A^* that is identical to H_n^2 , with the exception that *last* i proofs given by honest provers using $\langle P, V \rangle$ ordered by completion of Stage 4, the honest prover is simulated so as to use two random strings r_1, r_2 as input to the OT protocol instead of inputs that add up to the witness w .

As in the previous hybrids, we construct an adversary that violates the robustness of the \mathcal{SNMWI} protocol. Towards achieving this, we prove in a preliminary step that in experiments $G_n^{i(n)}$ for every function i , the adversary does not commit to the fake witness (i.e. is NON-ABUSING) in any $\langle P_s, V_s \rangle$ proofs that complete before the end of Stage 4 of the i^{th} interaction. More precisely, consider the truncated experiments \tilde{G}_n^i which proceeds identically to G_n^i , with the exception that the adversary A^* halts after completing the Stage 4 proof in the i^{th} interaction and outputs its view (this can be achieved by constructing a wrapper that internally emulates the code of A^* and cuts it off when required). Define $\widetilde{\text{HYB}}_{A^*, Z}^i(n)$ to be the random variable that represents the partial view of A^* at the end of the experiment \tilde{G}_n^i . Also, $\widetilde{\text{WIT}}_{A^*, Z}^{i,j}(n)$ represent the value committed to by A^* in the j^{th} $\langle P_s, V_s \rangle$ interaction where A^* controls the prover in \tilde{G}_n^i . If the j^{th} proof starts before the beginning of Stage 4 or after the end of Stage 4 in the i^{th} proof then let $\widetilde{\text{WIT}}_{A^*, Z}^{i,j}(n) = \perp$. We now have the following claim.

Subclaim 2. *For any function $i : \mathbb{N} \rightarrow \mathbb{N}$ and $j : \mathbb{N} \rightarrow \mathbb{N}$, the following distributions are indistinguishable over $n \in \mathbb{N}$:*

- $\left\{ (\widetilde{\text{HYB}}_{A^*, Z}^i(n), \widetilde{\text{WIT}}_{A^*, Z}^{i,j}(n)) \right\}_{n \in \mathbb{N}}$
- $\left\{ (\widetilde{\text{HYB}}_{A^*, Z}^{i-1}(n), \widetilde{\text{WIT}}_{A^*, Z}^{i-1,j}(n)) \right\}_{n \in \mathbb{N}}$

Consequently, for every adversary A^* that is NON-ABUSING in \tilde{G}_n^0 , there exists a

negligible function $\nu(\cdot)$ such that

$$Pr\left\{ \text{WIT}_{A^*,Z}^{i,j}(n) \text{ is a fake witness} \wedge j^{th} \langle P_s, V_s \rangle\text{-proof completes before Stage 4 of } i^{th} \langle P, V \rangle \text{ interaction} \right\} \leq \nu(n)$$

We turn towards comparing hybrids H_n^2 and H_n^3 and defer the proof of the claim to the end. As before, we have that the experiment $G_n^{q(n)}$ is identical to H_n^3 . We also have that the experiments G_n^i and G_n^{i-1} proceed identically up until the end of Stage 2 in the i^{th} proof using $\langle P, V \rangle$ (reverse ordered from the end). Let $\Gamma(A^*, z, n)$ denote the set of all possible joint views τ of A^* and all the parties such that the next message is the beginning of Stage 3 in the i^{th} proof using $\langle P, V \rangle$ where the prover is honest. Define random variables for G_n^i , $\text{HYB}_{A^*,Z}^i(n)$ and $\text{WIT}_{A^*,Z}^i(n)$ to be respectively the output of Z and the value committed to by A^* corresponding to a randomly chosen $\langle P_s, V_s \rangle$ interaction where A^* controls the prover.

Claim 13 (Claim 10 restated). *The following ensembles are indistinguishable over $n \in \mathbb{N}$:*

- $\left\{ \text{HYBRID}_{A^*,Z}^2(n) \right\}_{n \in \mathbb{N}} \left(= \left\{ \text{HYB}_{A^*,Z}^0(n) \right\}_{n \in \mathbb{N}} \right)$
- $\left\{ \text{HYBRID}_{A^*,Z}^3(n) \right\}_{n \in \mathbb{N}} \left(= \left\{ \text{HYB}_{A^*,Z}^{q(n)}(n) \right\}_{n \in \mathbb{N}} \right)$

Furthermore, if A^* is NON-ABUSING in G_n^0 , then it is NON-ABUSING in $G_n^{q(n)}$.

Proof: Assume for contradiction, either one of the following occurs,

Case 1: There exists a function $j : \mathbb{N} \rightarrow \mathbb{N}$ and polynomial $p(\cdot)$ such that for infinitely many n ,

$$\Pr \left[\text{WIT}_{A^*,Z}^{q(n),j(n)}(n) \text{ is a fake witness} \right] \geq \frac{1}{p(n)}$$

Case 2: There exists a distinguisher D and polynomial $p(\cdot)$ such that for infinitely many n ,

$$\Pr \left[D(\text{HYB}_{A^*,Z}^0(n)) = 1 \right] - \Pr \left[D(\text{HYB}_{A^*,Z}^{q(n)}(n)) = 1 \right] \geq \frac{1}{p(n)}$$

RESOLVING CASE 1: If Case 1 occurs for infinitely many n , then there exists a function $i : \mathbb{N} \rightarrow \mathbb{N}$ such that

$$\Pr \left[D(\text{HYB}_{A^*,Z}^{i(n)}(n)) = 1 \right] - \Pr \left[D(\text{HYB}_{A^*,Z}^{i(n)-1}(n)) = 1 \right] \geq \frac{1}{p(n)q(n)} \quad (5.5)$$

Let the second $\langle P_s, V_s \rangle$ proof in Stage 4 of the i^{th} interaction be the j^{th} $\langle P_s, V_s \rangle$ proof in which the adversary controls the prover. We assume further that

$$\Pr \left[\text{WIT}_{A^*,Z}^{i(n),j}(n) \text{ is a fake witness} \right] \leq \frac{1}{4p(n)q(n)} \quad (5.6)$$

$$\Pr \left[\widetilde{\text{WIT}}_{A^*,Z}^{i(n)-1,j}(n) \text{ is a fake witness} \right] \leq \frac{1}{4p(n)q(n)} \quad (5.7)$$

This follows without loss of generality for sufficiently large n from Sub-Claim 2.

Fix an n for which Equations 5.5-5.7 hold and let $i = i(n), j = j(n)$. Using A^* , we construct a distinguisher that distinguishes $(\widetilde{\text{HYB}}_{A^*,Z}^i(n), \widetilde{\text{WIT}}_{A^*,Z}^{i,j}(n))$ and $(\widetilde{\text{HYB}}_{A^*,Z}^{i-1}(n), \widetilde{\text{WIT}}_{A^*,Z}^{i-1,j}(n))$ with non-negligible probability and arrive at contradiction to Sub-Claim 2.

The distinguisher D^* on input (τ, w) proceeds as follows:

- If w is a witness to a puzzle interaction, then D halts outputting 0.
- Otherwise, D^* interprets w as a witness for the Stage 4 proof in i^{th} $\langle P, V \rangle$ -interaction. If w is not a fake witness, then let r be the value part of the witness w , that represents the receiver's output in the Stage 3 OT protocol. For the i^{th} proof, in Stage 5, D^* feeds r and continues emulation.
- On completion, D^* feeds the output of Z to D and outputs what D outputs.

It follows from the description that the internal emulation by D^* when the input is sampled from $(\widetilde{\text{HYB}}_{A^*,Z}^i(n), \widetilde{\text{WIT}}_{A^*,Z}^{i,j}(n))$ proceeds identically to G_n^i if $\widetilde{\text{WIT}}_{A^*,Z}^{i,j}(n)$ is not the fake witness. Similarly, the internal emulation proceeds identically to G_n^{i-1} when the input is sampled from $(\widetilde{\text{HYB}}_{A^*,Z}^{i-1}(n), \widetilde{\text{WIT}}_{A^*,Z}^{i-1,j}(n))$ if $\widetilde{\text{WIT}}_{A^*,Z}^{i-1,j}(n)$ is not the fake witness. Since $\widetilde{\text{WIT}}_{A^*,Z}^{i,j}(n)$ and $\widetilde{\text{WIT}}_{A^*,Z}^{i-1,j}(n)$ are each not fake except with

probability $\frac{1}{4p(n)q(n)}$, it follows that

$$\begin{aligned}
& \Pr \left[D^*(\widetilde{\text{HYB}}_{A^*,Z}^i(n), \widetilde{\text{WIT}}_{A^*,Z}^{i,j}(n)) = 1 \right] \\
& \quad - \Pr \left[D^*(\widetilde{\text{HYB}}_{A^*,Z}^{i-1}(n), \widetilde{\text{WIT}}_{A^*,Z}^{i-1,j}(n)) = 1 \right] \\
\geq & \Pr \left[D(\text{HYB}_{A^*,Z}^i(n)) = 1 \right] \\
& \quad - \Pr \left[D(\text{HYB}_{A^*,Z}^{i-1}(n)) = 1 \right] - \frac{2}{4p(n)q(n)} \\
\geq & \frac{1}{p(n)q(n)} - \frac{2}{4p(n)q(n)} \\
= & \frac{1}{2p(n)q(n)}
\end{aligned}$$

Therefore, D^* with the functions i and j as defined above, contradicts Sub-Claim 2 and this concludes the proof of Case 1.

RESOLVING CASE 2: If Case 2 occurs for infinitely many n , then there exists a function $i : \mathbb{N} \rightarrow \mathbb{N}$ such that either of the following case occurs:

Case 2a: There exists a function $i : \mathbb{N} \rightarrow \mathbb{N}$ such that for infinitely many n ,

$$\begin{aligned}
& \Pr \left[\text{WIT}_{A^*,Z}^{i(n),j(n)}(n) \text{ is a fake witness} \right] \\
& \quad - \Pr \left[\text{WIT}_{A^*,Z}^{i(n)-1,j(n)}(n) \text{ is a fake witness} \right] \geq \frac{1}{p(n)q(n)}
\end{aligned}$$

and a $\langle P_s, V_s \rangle$ proof corresponding to the $j(n)^{th}$ commitment completes before Stage 4 of the i^{th} $\langle P, V \rangle$ interaction completes. This case directly contradicts Sub-Claim 2 and hence can be ruled out.

Case 2b: There exists a function $i : \mathbb{N} \rightarrow \mathbb{N}$ such that for infinitely many n ,

$$\begin{aligned}
& \Pr \left[\text{WIT}_{A^*,Z}^{i(n),j(n)}(n) \text{ is a fake witness} \right] \\
& \quad - \Pr \left[\text{WIT}_{A^*,Z}^{i(n)-1,j(n)}(n) \text{ is a fake witness} \right] \geq \frac{1}{p(n)q(n)}
\end{aligned}$$

and a $\langle P_s, V_s \rangle$ proof corresponding to the $j(n)^{th}$ commitment begins after Stage 4 of the i^{th} $\langle P, V \rangle$ interaction completes.

In this case, we follow the same approach as Case 1. Assume further Equations 5.6 and 5.7 hold. Then consider a distinguisher D^* that proceeds identically as in Case 1, with the exception that on completion, for a randomly

chosen $\langle P_s, V_s \rangle$ -interaction that begins after Stage 4 of the i^{th} interaction where A^* controls the prover, D^* applies the (stand-alone) extractor guaranteed by the proof-of-knowledge property of $\langle P_s, V_s \rangle$ and extracts a witness w . D^* then outputs 1 if witness w is fake and 0 otherwise. As before, we have that the internal emulation carried out by D^* when inputs are sampled from $(\widetilde{\text{HYB}}_{A^*,Z}^i(n), \widetilde{\text{WIT}}_{A^*,Z}^{i,j}(n))$ and $(\widetilde{\text{HYB}}_{A^*,Z}^{i-1}(n), \widetilde{\text{WIT}}_{A^*,Z}^{i-1,j}(n))$ proceed identically to G_n^i and G_n^{i-1} respectively if $\widetilde{\text{WIT}}_{A^*,Z}^{i,j}(n)$ and $\widetilde{\text{WIT}}_{A^*,Z}^{i-1,j}(n)$ are not fake. Furthermore, whenever the execution completes, except with probability $\nu(n)$ for some negligible function $\nu(\cdot)$, the extraction succeeds. Therefore,

$$\begin{aligned} & \Pr \left[D^*(\widetilde{\text{HYB}}_{A^*,Z}^i(n), \widetilde{\text{WIT}}_{A^*,Z}^{i,j}(n)) = 1 \right] \\ & \quad - \Pr \left[D^*(\widetilde{\text{HYB}}_{A^*,Z}^{i-1}(n), \widetilde{\text{WIT}}_{A^*,Z}^{i-1,j}(n)) = 1 \right] \\ & \geq \Pr \left[\overline{\text{WIT}}_{A^*,Z}^{i,j}(n) \text{ is a fake witness} \right] \\ & \quad - \Pr \left[\overline{\text{WIT}}_{A^*,Z}^{i-1,j}(n) \text{ is a fake witness} \right] - \frac{2}{4p(n)q(n)} - 2\nu(n) \\ & \geq \frac{1}{2p(n)q(n)} - 2\nu(n) \end{aligned}$$

and again we obtain a contradiction to Sub-Claim 2.

Remark 4. Note that Cases 2a and 2b do not necessarily include all the $\langle P_s, V_s \rangle$ proofs given by the adversary. Consider a $\langle P_s, V_s \rangle$ -proof that begins before Stage 4 of the i^{th} begins and completes after Stage 4 finishes. This proof is not considered in either of the cases. However, it suffices to ensure that some $\langle P_s, V_s \rangle$ proof is considered for the commitment message in Stage 4 or Stage 5 (as the case may be) in every $\langle P, V \rangle$ interaction. This holds because in each of Stage 4 and Stage 5, two $\langle P_s, V_s \rangle$ proofs are provided and at least one of the two proofs fall in either Case 2a or Case 2b.

This concludes the proof of Claim 10. It only remains to prove Sub-Claim 2.

Proof of Sub-Claim 2: Assume for contradiction, there exists distinguisher D , functions $i, j : \mathbb{N} \rightarrow \mathbb{N}$ and polynomial $p(\cdot)$ such that for infinitely many n ,

$$\begin{aligned} & \Pr \left[D \left(\widetilde{\text{HYB}}_{A^*,Z}^{i(n)}(n), \widetilde{\text{WIT}}_{A^*,Z}^{i(n),j(n)}(n) \right) = 1 \right] \\ & \quad - \Pr \left[D \left(\widetilde{\text{HYB}}_{A^*,Z}^{i(n)-1}(n), \widetilde{\text{WIT}}_{A^*,Z}^{i(n)-1,j(n)}(n) \right) = 1 \right] \geq \frac{1}{p(n)} \end{aligned} \tag{5.8}$$

Fix a particular n for which this happens and let $i = i(n), j = j(n)$. Using A^* , we construct an adversary \tilde{A} and machine B such that \tilde{A} violates the k -robustness of the $\langle P_s, V_s \rangle$ proof and arrive at a contradiction. Observe that the only difference in experiments \tilde{G}_n^{i-1} and \tilde{G}_n^i is in the inputs used by the honest prover in the Stage 3 OT protocol of i^{th} interaction. Now, using the idea from the previous step, we consider a machine B plays that participates as the sender in the $\langle S_{OT}, R_{OT} \rangle$ protocol and on inputs 1 and 2 it changes from real to fake inputs. The adversary \tilde{A}^* internally incorporates A^* and forwards the Stage 3 of i^{th} instance to B and the j^{th} $\langle P_s, V_s \rangle$ proof on the right. We arrive at a contradiction to the robustness of the $\langle P_s, V_s \rangle$ proof by showing that interaction with B on real and fake inputs are indistinguishable.

More formally, let $\Gamma(A^*, z, n)$ denote the set of all possible joint views τ of A^* and all the parties in \tilde{G}_n^i such that the next message is the beginning of Stage 3 in the i^{th} proof. Given τ , define $v_b(\tau) = (b, w)$ where w is the real witness used by the honest prover in the i^{th} proof.

The machine B on input $v_n^b = (b, w)$, proceeds as follows:

- If $b = 1$, it chooses a random string r and runs the code of an honest sender for the $\langle S_{OT}, R_{OT} \rangle$ protocol with inputs $r, r \oplus w$.
- If $b = 2$, it choose two random strings r_1, r_2 and runs the code of an honest sender for the $\langle S_{OT}, R_{OT} \rangle$ protocol with inputs r_1, r_2 .

The adversary \tilde{A}^* on auxiliary input τ , proceeds as follows: It incorporates A^* , Z and all the parties and internally emulates the experiment G_n^0 in the following manner.

- It starts by feeding the parties all messages in τ .
- For the i^{th} proof, \tilde{A}^* forwards messages in the Stage 3 OT protocol to the external machine B (on the left).
- For the j^{th} $\langle P_s, V_s \rangle$ -interaction where A^* controls the prover, \tilde{A}^* forwards the messages to the external verifier (on the right). We remark that by definition, the j^{th} proof begins after the Stage 3 of the i^{th} proof begins.

- On completion of Stage 4 in the i^{th} proof, \tilde{A}^* cuts off the experiment and outputs the joint partial view of Z , A^* and all honest parties in the internal emulation.

Note that \tilde{A} will not be able to carry out the internal emulation beyond Stage 4 in the i^{th} proof without knowing the first input used by B in the OT protocol. However, this will not be a problem since we consider only the truncated experiment \tilde{G}_n^i . It therefore follows from the description that the interaction emulated internally by \tilde{A}^* when the auxiliary input z is chosen according to the distribution $\Gamma(A^*, Z, n)$ is identical to \tilde{G}_n^{i-1} when B 's input is v_n^1 and \tilde{G}_n^i when B 's input is v_n^2 . Therefore, it follows that

$$\begin{aligned}\left(\widetilde{\text{HYB}}_{A^*,Z}^{i-1}(n), \widetilde{\text{WIT}}_{A^*,Z}^{i-1,j}(n)\right) &= \left\{\tau \leftarrow \Gamma(A^*, Z, n) : \text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}^*, B}(v_n^1(\tau))\right\} \\ \left(\widetilde{\text{HYB}}_{A^*,Z}^i(n), \widetilde{\text{WIT}}_{A^*,Z}^{i,j}(n)\right) &= \left\{\tau \leftarrow \Gamma(A^*, Z, n) : \text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}^*, B}(v_n^2(\tau))\right\}\end{aligned}$$

Using Equation 5.8, it follows that

$$\begin{aligned}\Pr\left[\tau \leftarrow \Gamma(A^*, Z, n) : D\left(\text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}^*, B}(v_n^1(\tau))\right)\right] \\ - \Pr\left[\tau \leftarrow \Gamma(A^*, Z, n) : D\left(\text{mim}_{\langle P_s, V_s \rangle}^{\tilde{A}^*, B}(v_n^1(\tau))\right)\right] \geq \frac{1}{p(n)}\end{aligned}$$

Using sender privacy of the OT protocol in Stage 3, we have the following sub-claim:

Subclaim 3.

$$\left\{\text{view}_{\tilde{A}}[\langle B(v_1(z)), \tilde{A}(z) \rangle]\right\}_{n \in N, z \in \{0,1\}^*} \approx \left\{\text{view}_{\tilde{A}}[\langle B(v_2(z)), \tilde{A}(z) \rangle]\right\}_{n \in N, z \in \{0,1\}^*}$$

Since $\langle P_s, V_s \rangle$ is t_{OT} -robust and B interacts in at most t_{OT} rounds, it follows from the above claim that \tilde{A}^* , B and D violates the t_{OT} -robustness of the $\langle P_s, V_s \rangle$ and thus we arrive at a contradiction. This concludes the proof of the Sub-Claim 2. It only remains to prove the sub-claim.

Proof of Sub-Claim 3: Since the OT protocol is sender-private, it follows that there exists a cheating receiver that can distinguish the inputs in the ideal world where the parties have access to a ideal OT-functionality. However, the inputs used in the OT protocol when τ is sampled through Γ satisfy the property that the string obtained by the receiver from the ideal OT-functionality in the ideal-world is uniformly random. Thus, the view of the receiver in the ideal-world when

the senders inputs are chosen according to $v_1(\tau)$ and $v_2(\tau)$ are identical when τ is sampled from Γ . Therefore, the views in the real-world must at least be computationally indistinguishable. \square \square

This concludes the proof of the Lemma 7. \square

CHAPTER 6

APPLICATIONS OF THE GENERAL FRAMEWORK

In this section, we provide protocols to realize any functionality securely in different UC models. On a high-level, we construct a UC-puzzle in the different UC-models and obtain feasibility as a corollary to our main theorem.

More precisely, for secure computation to be feasible in a general model $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$, we require:

- A $(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$ -secure UC-puzzle.
- Existence of stand-alone SH-OT protocol that is secure w.r.t $cl(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$. These in turn can be constructed by relying on the existence of enhanced trapdoor permutation or homomorphic encryption secure w.r.t $cl(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$.
- A \mathcal{SNMWI} argument of knowledge protocol that is secure w.r.t $cl(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$; by this we mean an interactive argument system where both the argument of knowledge and \mathcal{SNMWI} property holds w.r.t adversaries in $cl(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$. These in turn can be constructed in $O(1)$ -rounds by relying on one-way functions secure w.r.t $cl(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$ using the construction provided in Chapter 4.

Since stand-alone SH-OT protocol secure w.r.t $cl(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$ is necessary for UC-security and its existence implies the existence of one-way functions secure w.r.t $cl(\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{sim}})$, it suffices to construct a UC-puzzle to demonstrate feasibility in a general model. Below, we provide UC-puzzles for various UC models.

6.1 Non-Uniform UC

In this model, we consider a uniform \mathcal{PPT} adversary and non-uniform \mathcal{PPT} simulator. We prove the following feasibility result for this model.

Theorem 5. *Assume the existence of t -round stand-alone secure SH-OT protocol, and an evasive promise problem in \mathbf{BPP} . Then, for every well-formed ideal functionality \mathcal{F} , there exists a $O(t)$ -round protocol π that realizes $\hat{\mathcal{F}}$ with Non-Uniform UC-security.*

As mentioned before, we construct a UC-puzzle in this model and obtain the theorem as corollary of Theorem 3. First, we introduce some new complexity theoretic assumptions that we use in the construction. Recall the notion of an *evasive sets* [33] and promise problems.

Definition 20. A set S is said to be *evasive*, if for all n , $S \cap \{0, 1\}^n \neq \emptyset$ and for any \mathcal{PPT} machine M , there is a negligible function $\nu(\cdot)$, such that, $\Pr[M(1^n) \in S \cap \{0, 1\}^n] \leq \nu(n)$

Definition 21. A promise-problem Δ is a pair of disjoint sets $\Delta_Y, \Delta_N \subseteq \{0, 1\}^*$ and $\Delta_Y \cup \Delta_N$ is called the promise.

Definition 22. A promise problem $\Delta = (\Delta_Y, \Delta_N)$ is *evasive*, if for all n , $\Delta_Y \cap \{0, 1\}^n \neq \emptyset$ and for \mathcal{PPT} machine M , there is a negligible function $\nu(\cdot)$, such that,

$$\Pr[M(1^n) \in \{0, 1\}^n \setminus \Delta_N] \leq \nu(n)$$

We now turn towards constructing a UC-puzzle in this model. For simplicity, we begin by constructing a puzzle relying on a slightly stronger assumption, namely, the existence of an *evasive* set Δ in \mathcal{P} . Then, we give the actual construction assuming the existence of an *evasive* promise-problem in **BPP**.

Lemma 10. Assume the existence of an *evasive* set L in \mathcal{P} . Then, there exists a puzzle in $(\mathcal{PPT}, n.u.\mathcal{PPT})$ with an empty protocol.

Proof: Let λ denote the empty string. Define the puzzle $\mathcal{P}_{nu} = (\langle S, R \rangle, \mathcal{R})$ as follows:

Protocol $\langle S, R \rangle$: S and R on input 1^n run the empty protocol.

Relation: $\mathcal{R} = \{(x, \lambda) | x \in L\}$

We prove soundness and statistical simulatability of the puzzle.

SOUNDNESS: Since, L is evasive, no cheating \mathcal{PPT} receiver can output x such that $(x, \lambda) \in \mathcal{R}$, i.e. $x \in L$ with more than negligible probability.

STATISTICAL-SIMULABILITY: Consider an adversary A that participates in a concurrent puzzle execution with environment Z . We construct a nuPPT adversary A' that receives $\delta \in L$ as non-uniform advice and proceeds as follows. It incorporates A internally and emulates an execution with A . It forwards all messages from A to Z , except the messages involved in the puzzle interactions with A . However, since the protocol is empty, there are no messages exchanged in the puzzle interaction. To output a witness, A' simply outputs δ on its special output tape whenever A sends $(\text{TRANS} = \lambda, C)$ to Z for a puzzle interaction. Finally, since the interaction between A' with Z is identical to the interaction between A with Z , the real and ideal executions are perfectly indistinguishable to Z . \square

We show at the end of this section that the stronger assumption is implied by a number of more standard assumptions, such as,

- the existence of a uniform collision-resistant hash-function.
- the existence of a language $L \in \mathbf{NE}$ (where $\mathbf{NE} = \mathbf{Ntime}(2^{O(n)})$) that is hard-on-the-average for probabilistic exponential time machines, i.e. $\mathbf{BPTIME}(2^{O(n)})$; in other words, a scaled-up version of the assumption that there exists a language in \mathbf{NP} that is hard on the average for \mathcal{PPT} machines.
- the existence of a language $L \in \mathbf{NE} \cap \text{coNE}$ that is *worst-case* hard for probabilistic exponential time machines; in other words, a scaled up version of the assumption that there exists a language in $\mathbf{NP} \cap \text{coNP}$ that is worst-case hard for \mathcal{PPT} machines.

We turn towards constructing a puzzle based on the weaker assumption.

Lemma 11. *Assume the existence of an evasive promise-problem Δ in \mathbf{BPP} . Then, there exists a puzzle in $(\mathcal{PPT}, n.u.\mathcal{PPT})$.*

Proof: Let $\Delta = (\Delta_Y, \Delta_N)$ and M be the \mathcal{PPT} machine that decides Δ . To construct a puzzle, we modify the protocol from the previous lemma as follows: The sender samples r_1, \dots, r_n uniformly at random from $\{0, 1\}^{r(n)}$, where $r(\cdot)$ is the polynomial that bounds the number of random bits used by M for inputs of length n . The relation \mathcal{R} is defined as $\{(w, (r_1, \dots, r_n)) | \#\{i | M_{r_i}(w) = 1\} > \frac{n}{3}\}$, where M_r stands for the machine M with the random tape fixed to r . The soundness follows from the evasiveness of Δ . For simulability, consider A' that receives an

element δ from Δ_Y as non-uniform advice and outputs δ as witness for a puzzle-transcripts $\text{TRANS} = (r_1, \dots, r_n)$, if $\#\{i | M_{r_i}(w) = 1\} > \frac{n}{3}$. If for some puzzle-transcript TRANS , the condition is not true, A' aborts outputting \perp . Conditioned on A' not aborting, it follows that the simulation is perfectly indistinguishable to Z . For every $w \in \Delta_Y$, using Chernoff-bound we have that the probability A' aborts is $\text{poly}(n)2^{-O(n^2)}$ which is exponentially small. Thus, $\{\text{REAL}_{A,Z}(n)\}_{n \in \mathbb{N}}$ and $\{\text{IDEAL}_{A',Z}(n)\}_{n \in \mathbb{N}}$ are statistically close. \square

Quite surprisingly, we show in Section 7, that the weaker assumption is in fact necessary to achieve secure computation with Non-Uniform UC-security. Since stand-alone secure SH-OT is also necessary, we obtain necessary and sufficient conditions for Non-Uniform UC-security to be feasible.

6.2 Quasi-Polynomial UC

In this model, we consider a \mathcal{PPT} adversary and a \mathcal{PQT} simulator. Assuming the existence of one-way functions that are “invertable” by \mathcal{PQT} machines, we show in Lemma 12 and obtain the following feasibility result.

Theorem 6. *Assume the existence of stand-alone secure SH-OT secure w.r.t \mathcal{PQT} and one-way functions that can inverted w.p. 1 in \mathcal{PQT} . Then, for every well-formed ideal functionality \mathcal{F} , there exists a $O(t)$ -round protocol π that realizes $\hat{\mathcal{F}}$ with QPS UC-security.*

We remark that one-way functions that are invertable by \mathcal{PQT} machines can be constructed based on one-way functions with sub-exponential hardness. Therefore, assuming sub-exponential hardness, we obtain as corollary an $O(t)$ -round protocol that securely realizes any functionality with QPS-UC-security. In comparison with previous works, we obtain the following improvements.

Complexity Assumptions: We require stand-alone secure SH-OT w.r.t \mathcal{PQT} and one-way functions with sub-exponential hardness (or one-way functions invertable by \mathcal{PQT}). The protocol in [6] apart from relying on the particular assumption of the existence of enhanced-trapdoor permutation secure w.r.t \mathcal{PQT} , also requires collision-resistant hash functions that are secure w.r.t sub-exponential circuits.

Security: We achieve a stronger notion of security, which (in analogy with [63]) requires that the output of the simulator is indistinguishable also for \mathcal{PQT} ; in contrast, [6] only achieves polynomial-time indistinguishability. In essence, this means that anything an attacker can learn “on-line” (in poly-time) can be simulated “off-line” (in qpoly-time) in a way that is indistinguishable also “off-line”. In this language, [6] only achieves on-line indistinguishability.

Lemma 12. *Assume the existence of a one-way function f , such that there exists a \mathcal{PQT} machine that inverts f w.p. 1. Then, there exists a 1-round $(n.u.\mathcal{PPT}, \mathcal{PQT})$ -secure UC-puzzle.*

Proof: Let f be the one-way function. Define $\mathcal{P}_{pqt} = (\langle S, R \rangle, \mathcal{R})$ as follows

Protocol $\langle S, R \rangle$: $S \rightarrow R$: S picks $x \leftarrow \{0, 1\}^n$ and sends $y = f(x)$ to R

$S \leftrightarrow R$: a witness-hiding argument of knowledge of the statement that there exists x' such that $y = f(x')$

Relation: $\mathcal{R} = \{(x, y) | y = f(x)\}$

SOUNDNESS: This follows directly from the one-wayness of f and the witness-hiding property of the proof given by the sender.

STATISTICAL SIMULATABILITY: Simulation follows identically as in the case for Non-Uniform UC model. To output a witness, we require A' to compute the inverse of $y = f(x)$ for a random x . While emulating A , if A completes a puzzle-interaction, then A' tries to invert x to obtain a witness y such that $y = f(x)$ if one exists and halts outputting fail. Since f is invertible in \mathcal{PQT} , we have that A' succeeds whenever y is in the range of f and the simulation is identical to the real execution. It suffices to bound the failure probability of A' . Since A uses a proof-of-knowledge in the puzzle to prove that y is in the range of f , except with negligible probability, for every puzzle, it holds that y is in the range of f . \square

Remark 5. *If we further assume the existence of collision-resistant hash functions that are secure w.r.t \mathcal{PQT} , we can construct a UC-puzzle relying on the weaker*

assumption of the existence of a \mathcal{PPT} interactive Turing Machine M such that for every \mathcal{PPT} machine P , $\Pr[\text{OUTPUT}_M(\langle P, M \rangle(1^n)) = 1]$ is negligible and there exists a \mathcal{PQT} machine \tilde{P} such that $\Pr[\text{OUTPUT}_M(\langle P, M \rangle(1^n)) = 1]$ is negligibly close to 1. In fact, this weaker assumption can be shown to be necessary for realizing QPS-UC-security. However, both the construction and the lower-bound is beyond the scope of this thesis.

6.3 UC in the Common Reference String model

In the traditional Common Reference String (CRS) model, all parties have access to a common reference string that is chosen from a specified *trusted* distribution D . In the UC framework, this is captured via an ideal functionality $\mathcal{F}_{\text{CRS}}^D$ that samples a string τ from a pre-determined fixed distribution D and sets τ as the reference string. Canetti, Lindell, Ostrovsky and Sahai in [19] show that any functionality can be realized in the \mathcal{F}_{CRS} -hybrid assuming the existence of enhanced trapdoor permutations.

We show how to construct a puzzle relying on one-way functions. Since one-way functions can be based on stand-alone SH-OT, we obtain the following theorem.

Theorem 7. *Assume the existence of a $O(t)$ -round stand-alone secure SH-OT. Let G be a pseudo-random generator. Then, for every well-formed ideal functionality \mathcal{F} , there exists a $O(t)$ -round protocol π that realizes $\hat{\mathcal{F}}$ with UC-security in $\mathcal{F}_{\text{CRS}}^G$ -hybrid.*

In comparison with previous works, our construction for the CRS-model is optimal w.r.t complexity assumptions since stand-alone SH-OT is necessary. Since we construct a $O(t)$ -round protocol, our construction is also round-optimal.

We show how to construct a puzzle in the $\mathcal{F}_{\text{CRS}}^G$ -hybrid, where G is a pseudo-random generator. The puzzle for this model, $\mathcal{P}_{\text{CRS}} = (\langle S, R \rangle, \mathcal{R})$ is defined as follows. The protocol $\langle S, R \rangle$ in this model, simply requires S and R on input sid to request the reference string by sending sid to the ideal functionality $\mathcal{F}_{\text{CRS}}^G$. As in the quasi-polynomial model, we define the relation \mathcal{R} to include the set of tuples $(x, G(x))$. Soundness of the puzzle, follows from the pseudo-randomness (and thus, one-wayness) of G . In a concurrent puzzle execution the puzzle environment

Functionality \mathcal{F}_{CRS}^D

1. Upon activation with session id sid proceed as follows. Run the sampling algorithm D on a uniformly distributed random input ρ from $\{0,1\}^n$ to obtain a reference string $r = D(\rho)$. Store D, ρ, r and send (URS, sid, ρ) to the adversary.
2. When receiving input (CRS, sid) from some party P with session id sid' , send (CRS, sid, ρ) to that party if $sid = sid'$; otherwise ignore the message.

Figure 6.1: Uniform Reference String functionality

Z decides the sid that the receivers receive as input. Furthermore, since Z is allowed to interact only with the adversary, it can never obtain the reference string directly from \mathcal{F}_{CRS}^G . We obtain simulability, by allowing the simulator A' to set the reference string as $G(x)$ for its choice of x , and thus enable A' to obtain the witness x (to all puzzles) directly.

We remark that pseudo-random generator can be constructed relying on one-way functions[43] which in turn exist if SH-OT exists. Therefore, existence of SH-OT is necessary and sufficient for UC-security to be feasible in the CRS model.

6.4 UC in the Uniform Reference String model

When the distribution D in the CRS-functionality is fixed as the uniform distribution, we obtain the uniform reference string (URS) model. In [19], assuming further the existence of dense crypto-systems, they show how to realize any functionality in the URS-model. They require this additional assumption to embed the key of a public-key crypto-system in the reference string for ensuring non-malleability. We show that how to improve their construction by achieving the same without assuming the existence of dense-crypto systems and using only stand-alone SH-OT.

Let the URS-functionality be $\mathcal{F}_{URS} = \mathcal{F}_{CRS}^I$, where I is the identity function. Since, the \mathcal{F}_{CRS}^G -functionality implements the \mathcal{F}_{URS} -functionality when G is a pseudo-random generator, any protocol that realizes f in the \mathcal{F}_{CRS}^G also realizes the same functionality in the \mathcal{F}_{URS} -hybrid. This follows from the UC-composition Theorem [12]. Thus, without any additional assumptions, we obtain a protocol in the URS-model.

Theorem 8. *Assume the existence of a t -round stand-alone secure SH-OT. Then, for every well-formed ideal functionality \mathcal{F} , there exists a $O(t)$ -round protocol π that realizes $\hat{\mathcal{F}}$ with UC-security in \mathcal{F}_{URS} -hybrid.*

As in the CRS-model, our construction is optimal both in complexity-assumptions and round-complexity.

6.5 UC in the Key Registration model

The Key-Registration (KR) service introduced in [2], represents a method for all parties to obtain a public-key from a seed (representing the secret-key) which is kept private by the service. The service is modeled as an ideal functionality parameterized by a function $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$. Any party can register with the KR service and obtain a public key in return. In a registration process, the party obtains a public key $v = f(r)$ for some uniformly chosen r known only to the service. In addition, a corrupted party may provide the service with any arbitrary r and have its public-key set to r . In an execution, whenever a party asks the service for the public key of another party, the service returns one of the keys registered for that party. We provide the description of the KR-functionality from [2] in Figure 6.2.

In [2], they show feasibility of UC-security in the KR-hybrid assuming enhanced trapdoor permutation for *specific* one-way functions f .¹ We dispense the latter assumption and show how to achieve the same assuming only stand-alone secure SH-OT. We prove the following feasibility theorem for this model.

Theorem 9. *Assume the existence of a t -round SH-OT. Let f be a one-way function. Then, for every well-formed ideal functionality \mathcal{F} , there exists a $O(t)$ -round protocol π that realizes $\hat{\mathcal{F}}$ with UC-security in \mathcal{F}_{KR}^f -hybrid.*

Again, our construction is optimal both in complexity-assumptions and round-complexity.

¹They required one-way functions that mapped secret keys to corresponding public keys of a crypto-system

Functionality \mathcal{F}_{KR}^f

Upon activation with input sid and security parameter n , proceed as follows.

Initialize a set of R of strings empty.

Registration: When receiving a message $(\text{register}, sid)$ from a party P_i (which is either corrupted or uncorrupted), send $(\text{register}, sid, P_i)$ to the adversary A and receive a value p' . Then, if $p' \in R$ then let $p \leftarrow p'$. Else, choose $r \leftarrow \{0,1\}^n$, let $p \leftarrow f(r)$, and add p to R . Finally, record (P_i, p) and return (sid, p) to P_i and to A .

Registration by a corrupted party: When receiving a message $(\text{register}, sid, r)$ from a corrupted party P_i , record $(P_i, f(r))$. In this case, $f(r)$ is not added to R .

Retrieval: When receiving a message $(\text{retrieve}, sid, P_i, P_j)$ from P_j , send $(\text{retrieve}, sid, P_i, P_j)$ to A , and obtain a value p from A . If (P_i, p) is recorded then return (sid, P_i, p) to P_j . Else, return (sid, P_i, \perp) to P_j .

Figure 6.2: Key Registration functionality

We prove the theorem by constructing a puzzle. Define the puzzle $\mathcal{P}_{kr} = (\langle S, R \rangle, \mathcal{R})$ in \mathcal{F}_{KR}^f as follows. The protocol $\langle S, R \rangle$ requires the receiver R to obtain the public-key of the sender by sending the message $(\text{retrieve}, sid, S, R)$ to \mathcal{F}_{KR}^f . The relation \mathcal{R} is defined as in the CRS model with function f . Soundness of the puzzle follows from the one-wayness of f . To obtain simulation, the simulator of the puzzle A' , internally emulates the adversary A and emulates an interaction, by forwarding all messages sent by A to its puzzle-environment externally to A' 's environment. All messages sent to \mathcal{F}_{KR}^f are however treated differently so that A' obtains the secret-key corresponding to the public-keys of all senders controlled by the adversary A in a concurrent puzzle execution. More precisely, the adversary A sends two kinds of messages to \mathcal{F}_{KR}^f .

- A , controlling party P_i can register with \mathcal{F}_{KR}^f honestly. In this case, A' does not let \mathcal{F}_{KR}^f chose the public-key, but instead, chooses the public-key r and registers r with \mathcal{F}_{KR}^f . Internally, A' feeds the message $(P_i, f(r))$ to A .
- A , controlling party P_i chooses the public-key r and registers by sending r to \mathcal{F}_{KR}^f . In this case, A' stores r and forwards it to \mathcal{F}_{KR}^f .

Finally, A' is required to output a witness corresponding to every puzzle-interaction s . Recall that, s is the public-key of some sender controlled by A , and therefore, the witness is the secret-key corresponding to s . Since, A' knows all the secret-keys of senders controlled by A , it can output the witness accordingly.

6.6 UC in the Sunspots model

The *sunspots model* introduced by Canetti, Pass and Shelat [20], considers a variant of the CRS model. In contrast to the traditional CRS-model, here the ideal-functionality, that we call the \mathcal{F}_{sun} , sets the reference string by sampling uniformly from an *efficient* distribution D (that has sufficient min-entropy) which is initially *set* by the environment Z . In [20], they show how to securely realize any functionality in the \mathcal{F}_{sun} -hybrid. Their construction, however, requires every pair of parties to share two separate reference string in order to achieve non-malleability. We show how to remove this assumption, by constructing a protocol that is secure, when parties have access to only a *single* reference string with minimal min-entropy guarantees.

In [20], the sunspots model allows the environments to decide the distribution D for the \mathcal{F}_{sun} -functionality. This is done, in order to consider random sources computable in exponential time and derive (black-box) lower bounds. Since, we focus only on feasibility, we slightly simplify the model (as suggested in [20]) and instead directly let the adversary select the distribution D .

As in [20], we consider (μ, d, t) -conforming adversaries:

Definition. An adversary A is called (μ, d, t) -conforming if the following conditions hold:

1. Given security parameter 1^n , and upon receiving a message **(Activated-CRS, sid)** from \mathcal{F}_{sun} , A directly replies by sending back a message **(Distribution, sid, D)**, where the sampling algorithm D outputs reference strings of length n , has description size at most $d(n)$, and generates an output within $t(n)$ steps.
2. The distribution induced by the output of \mathcal{F}_{sun} in the execution by A (on input 1^n) has min-entropy at least $\mu(n)$ (over the random choices of both A and \mathcal{F}_{sun}).

First, we state our feasibility theorem for the Sunspots model.

Theorem 10. Assume the existence of t -round stand-alone secure SH-OT and collision-resistant hash-functions. Then, for every well-formed ideal functionality

Functionality \mathcal{F}_{sun}

1. Upon activation with session id sid proceed as follows. Send the message $(\text{Activated}, sid)$ to the adversary, and wait to receive back a message (n, sid, D) . Run the sampling algorithm D on a uniformly distributed random input ρ from $\{0, 1\}^n$ to obtain a reference string $r = D(\rho)$. Store D, ρ, r and send $(\text{CRS}, sid, r, \rho)$ to the adversary.
2. When receiving input (CRS, sid) from some party P with session id sid' , send (CRS, sid, r) to that party if $sid = sid'$; otherwise ignore the message.

Figure 6.3: The Sunspots functionality

\mathcal{F} , there exists a $O(t)$ -round protocol π in the \mathcal{F}_{sun} -hybrid that realizes $\hat{\mathcal{F}}$ with UC-security w.r.t (μ, d, t) -conforming adversaries where $\mu(n) - d(n) > n^\varepsilon$ for some $\varepsilon > 0$.

We remark that as in [20], we can achieve the same result for smaller min-entropies (i.e. $\mu(n) - d(n) > \text{poly}(\log n)$) by additionally assuming one-way functions with sub-exponential hardness. To prove the above theorem, it suffices to construct a $O(1)$ -round UC-puzzle in the \mathcal{F}_{sun} -hybrid model. To show feasibility with UC-security in the \mathcal{F}_{sun} -hybrid, we use a similar approach to the \mathcal{F}_{URS} -hybrid, by first considering a $\mathcal{F}_{\text{sun}}^G$ -hybrid model, where $\mathcal{F}_{\text{sun}, G}$ is the ideal-functionality identical to \mathcal{F}_{sun} , with the exception that, instead of running the sampling algorithm D on a uniformly distributed ρ , it runs D on input $G(x)$ for a uniformly random x , where G is a pseudo-random generator. Then similar to the URS-model, we conclude that the protocol constructed in the $\mathcal{F}_{\text{sun}}^G$ -hybrid also securely realizes the functionality in the \mathcal{F}_{sun} -hybrid.

We proceed towards constructing a puzzle in the $\mathcal{F}_{\text{sun}}^G$ -hybrid. Let $G : \{0, 1\}^{n^\delta} \rightarrow \{0, 1\}^*$ be a pseudo-random generator that expands a seed of length n^δ (for $\delta > 0$) to a stream of bits such that $d(n) + n^\delta + |G| < \mu(n)$. Such a $\delta(n)$ is guaranteed to exist since $\mu(n) - d(n) > n^\varepsilon$. Such a generator can in turn be constructed relying on any one-way function[43].

Our construction of the puzzle is essentially identical to the construction used in [20], the only difference is we rely on the use of statistically-hiding commitments in the construction. We mention that although we rely on a similar protocol, we show that a single reference string is enough, while [20] requires many strings. As in [20], we rely on *universal arguments* to construct the puzzle. Universal Arguments

(UARGs) [3], are a variant of CS-proofs introduced by Micali [55]. Informally, such proofs systems are used in order to provide “efficient” proofs to statements of the form $y = (M, x, t)$, where y is considered to be a true statement if M is a non-deterministic machine that accepts x within t steps. Such a system can be constructed based on collision-resistant hash functions. See [3] for more details.

Let $(V_1, P_1, V_2, P_2, V_3)$ be the respective verifier and prover algorithms for a public-coin UARG for the language $\{r | r \in \{0, 1\}^n \text{ and } \text{KOL}(r) < n^{\frac{\varepsilon+\delta}{2}}\}$, where $\text{KOL}(x)$ is the *Kolmogorov complexity* of a string x .² We describe a language of transcripts of universal arguments in which the prover’s messages are committed instead of sent to the verifier. (In the protocol, we make use of the fact that the universal argument we use is a public-coin system, so the Verifier can always choose a next message without seeing the Prover’s prior messages.). Let com be a *statistically-hiding* commitment scheme (which exists based on collision-resistant hash functions)

Protocol $\langle S, R \rangle$: S and R obtain the reference string r from the $\mathcal{F}_{\text{sun}}^G$ -functionality.

$S \rightarrow R$: Pick $m_1 \leftarrow V_1(r, n)$ and send to R .

$R \rightarrow S$: Pick $c_1 \leftarrow \{0, 1\}^l$ and send to S .

$S \rightarrow R$: Pick $m_2 \leftarrow V_2(r, n)$ and send to R .

$R \rightarrow S$: Pick $c_2 \leftarrow \{0, 1\}^l$ and send to S .

$$\text{Relation: } \mathcal{R} = \left\{ (\text{TRANS}, w) \middle| \begin{array}{l} \text{TRANS} = (r, v_1, c_1, v_2, c_2), \\ w = ((p_1, r_1), (p_2, r_2)), \\ \exists r_1, r_2 \quad c_1 \leftarrow \text{com}(p_1, r_1), c_2 \leftarrow \text{com}(p_2, r_2) \\ \text{and } V_3(s, v_1, p_1, v_2, p_2) = 1 \end{array} \right\}$$

SOUNDNESS: Suppose a \mathcal{PPT} receiver R^* is able to break the soundness by outputting the witness for a puzzle with probability p . We use R^* to construct another

²The Kolmogorov complexity of a string x is the size of the smallest Turing machine which produces x on its output tape, when run on the empty tape.

efficient algorithm P which breaks the soundness property of the universal argument system with probability $\text{poly}(p)$. The soundness of the universal argument system therefore implies that p must be negligible which implies the soundness of the puzzle. We show that P breaks the soundness of the universal argument w.p. $\text{poly}(p)$ on the statement that the reference string r sampled from $\mathcal{F}_{\text{sun}}^G$ -functionality has a “short” description. Since, G is pseudo-random, if p is non-negligible, then P breaks the soundness with non-negligible probability in the hybrid experiment when r is sampled from \mathcal{F}_{sun} -functionality. Since, D has min-entropy $\mu(n)$, w.p. at most $2^{-n^{\frac{\varepsilon-\delta}{2}}} = 2^{-O(n^\varepsilon)}$, r has a short description and therefore no computationally unbounded prover can succeed in the the universal argument with non-negligible probability. Thus, p is non-negligible.

More precisely, P upon receiving the verifier message v_1 , feeds v_1 to R and then internally simulates the rest of the puzzle until R outputs the witness. By hypothesis, this succeeds with probability p . Let p_1 be a decommitment to the first message sent by R . P forwards p_1 externally to the verifier and receives the next message v_2 . At this point, P rewinds R and feeds v_2 instead of the second message (simulated before) from the verifier and continues to simulate the rest of the puzzle. If R returns a valid decommitment p_2 for the second message sent by R , P forwards p_2 to the external verifier. We argue that with probability at least p^2 , the transcript (v_1, p_1, v_2, p_2) is an accepting transcript for the universal argument.

SIMULABILITY: We achieve statistical simulation similar to the puzzle in the URS-model, by allowing the simulator A' to set the reference string and obtain the witness, which is the description of D , G and x , whose combined size by construction is $n^\delta + O(1) + d(n) < n^{\frac{\varepsilon+\delta}{2}}$. Furthermore, while emulating a receiver in a puzzle with adversary A , instead of following the honest receiver’s code, runs the code of an honest prover (P_1, P_2) in the universal argument with witness (D, G, x) and sends a commitment to the messages generated by P_1 and P_2 . The witness output by A' for the puzzle are decommitments to the messages sent in the puzzle.

6.7 UC in the Tamper-Proof Hardware Model

In this section, we consider a model that incorporates the physical assumption that protocols can be run in an isolated environment. In particular, we consider the existence of tamper-proof hardware, which enables a party P_i to “create” and “give” to another party P_j a hardware token T_F implementing any desired functionality F , and P_j can then access the embedded functionality in a black-box manner. Here the token is tamper-proof meaning that any adversary having access to the token can do no more than observing the input/output characteristics of the token. Such tamper-proof hardware was first introduced by Katz [45] in the context of UC-security, and formalized as the wrap ideal functionality \mathcal{F}_{wrap} , which accepts two kinds of messages.

- In order for a party P_i to “create” a token and “give” it to another party P_j , it sends the message $(\text{create}, sid, P_i, P_j, M)$ to \mathcal{F}_{wrap} , where M is the interactive Turing machine implementing the functionality P_i wants to embed in the token. Upon receiving such a message, the ideal functionality stores internally the initial state $(P_i, P_j, M, 0, \emptyset)$ for this token, if there has not been a token from P_i to P_j .
- Once a token is created and given to P_j , it can then access the token (i.e. interacts with M) via \mathcal{F}_{wrap} . More precisely, \mathcal{F}_{wrap} chooses a fresh random tape r for M , and internally emulates the execution of $M(r)$, by “forwarding” messages from $M(r)$ externally to P_j . Furthermore, in the emulation, M is also allowed to receive messages from P_i (forwarded by \mathcal{F}_{wrap}). However, it can only send messages to P_j ; and it cannot communicate with any other parties except P_i and P_j .

We recall the formal description of \mathcal{F}_{wrap} for a two-round interactive protocol in [45] in figure 6.4.

Katz shows that UC secure computation is feasible in the \mathcal{F}_{wrap} -hybrid model relying on the DDH assumption. Using our generalized framework, we remove the need for specific number-theoretic assumption, and show how to construct by relying only on the existence of SH-OT.

Theorem 11. *Assume the existence of a t -round stand-alone secure SH-OT.*

Functionality \mathcal{F}_{wrap}

\mathcal{F}_{wrap} is parameterized by a polynomial p and an implicit security parameter n .

“Creation”: Upon receiving $(\text{create}, sid, P_i, P_j, M)$ from P_i , where P_j is another user in the system and M is an interactive Turing machine, do:

1. Send $(\text{create}, sid, P_i, P_j)$ to P_j .
2. If there is no tuple of the form $(P_i, P_j, *, *, *)$ stored, then store $(P_i, P_j, M, 0, \emptyset)$.

“Execution”: Upon receiving $(\text{run}, sid, P_i, msg)$ from P' , find the unique stored tuple $(P_i, P_j, M, k, state)$ (if no such tuple exists, then do nothing). Then do:

Case 1 ($i = 0$): Choose random $r \leftarrow \{0, 1\}^{p(n)}$. Run $M(msg, r)$ for at most $p(n)$ steps, and let out be the response (set $out = \perp$ if M does not respond in the allotted time). Send (sid, P_i, out) to P_j . Store $(P_i, P_j, M, 1, (msg, r))$ and erase $(P_i, P_j, M, i, state)$.

Case 2 ($i = 1$): Parse state as $(msg1, r)$. Run $M(msg1 \| msg, r)$ for at most $p(n)$ steps, and let out be the response (set $out = \perp$ if M does not respond in the allotted time). Send (sid, P_i, out) to P_j . Store $(P_i, P_j, M, 0, \emptyset)$ and erase $(P_i, P_j, M, i, state)$.

If during the execution, M sends a message msg' addressed to party P . (Assume, without loss of generality, that the identity of the receiver is encoded in the message.) Send (sid, P_i, msg') to P_i , if $P = P_i$, and ignore otherwise.

Figure 6.4: Wrap functionality

Then, for every well-formed ideal functionality \mathcal{F} , there exists a $O(t)$ -round protocol π that realizes $\hat{\mathcal{F}}$ with UC-security in the $\mathcal{F}_{\text{wrap}}$ -hybrid model.

To show the theorem, it suffices to construct a puzzle in the $\mathcal{F}_{\text{wrap}}$ -hybrid model. In all the previous models, the puzzle protocols $\langle S, R \rangle$ are executed in a “stateless” way, that is, whenever a party intends to challenge (acting as the sender of the puzzle) another, it spawns *independently* a new subroutine of S to generate the puzzle. In this model, we, however, consider a “stateful” puzzle, which requires a party to spawn a subroutine of S at the beginning of its execution, and use this subroutine to generate all the puzzles it needs throughout its lifetime. (Note that the receiver part of the puzzle protocol is still “stateless”.) It is stateful in the sense that the subroutine can keep states across multiple invocations, and hence the puzzle instances generated are not independent to each other, but correlated. More precisely, We define the puzzle $\mathcal{P}_{\text{wrap}} = (\langle S, R \rangle, \mathcal{R})$ for the $\mathcal{F}_{\text{wrap}}$ -hybrid model as follows. The interactive Turing machine S , proceeds in two phases.

- When it is first spawned and invoked on inputs the identity of the sender P_i and the session id sid —called the initialization phase—it uniformly picks a string $x \in \{0, 1\}^n$, computes its image y through the one-way function f , and stores (y, P_i, sid) as an internal state.
- Later when S is invoked on inputs the identity of the puzzle receiver P_j to challenge P_j —called the challenging phase— S checks whether this is the first time interacting with party P_j , if so, it “creates” and “give” P_j a token, which encapsulates the functionality M that gives a witness-hiding argument-of-knowledge of the statement that y is in the image set of f , by sending the message $(\text{create}, sid, P_i, P_j, M)$ to $\mathcal{F}_{\text{wrap}}$. To actually challenge P_j , S simply sends y as the puzzle to the receiver.

On the other hand, the interactive Turing machine R , upon receiving y from the sender, accesses M via $\mathcal{F}_{\text{wrap}}$. More precisely, it first sends the message $(\text{run}, sid, S, \varepsilon)$ to $\mathcal{F}_{\text{wrap}}$ (where ε is an empty string), and then receives from M a WHAOK of the statement that y is in the image set of f (forwarded by $\mathcal{F}_{\text{wrap}}$). Finally, the puzzle relation \mathcal{R} is simply $\{(x, y) \mid y = f(x)\}$.

The soundness of the puzzle follows directly from the one-wayness of the function f and the witness-hiding property of the protocol. Furthermore, to simulate a

concurrent puzzle execution with A and the environment Z , A' internally emulates an execution with A and acts as the \mathcal{F}_{wrap} functionality for A . Whenever A sends a message $(create, sid, P_i, P_j, M^*)$ to \mathcal{F}_{wrap} , A' obtains the message. Later to extract the witness of a puzzle y challenged by A (controlling P_i) to P_j , A' simply rewinds M^* in the witness-hiding argument-of-knowledge sub-protocol to extract the witness. Notice that since M^* is prohibited from receiving messages from other parties except P_j , it would never expect any new messages from parties other than P_j during rewinding. Therefore the extraction can be finished in isolation, without intervening the adversary A and the environment Z . Hence we achieve perfect simulation.

6.8 Stand-alone Model

The UC-framework enables to guarantee security even when multiple protocol instances are running concurrently. It might still be useful to capture within the UC framework also security properties that are not necessarily preserved under concurrent composition. This section aims at constructing protocols for one such milder setting that guarantees that during the execution of a protocol instance no other protocol instances are running concurrently. This model is usually referred to as the *stand-alone* or non-concurrent setting.

This model does not directly follow from our general UC-model. However, it is possible to alter the model following [12] and construct a puzzle to show feasibility. First we describe the model from [12] in more detail and then prove feasibility.

To model non-concurrent behavior, we first require synchronous communication (as opposed to just authenticated communication). Roughly speaking, in addition to authenticated delivery, here the computation proceeds in *rounds*, when in each round each party receives all the messages that were sent to it in the previous round, and generates outgoing messages for the next round. This is formally modeled through the \mathcal{F}_{syn} ideal-functionality that expects a list of \mathcal{P} of parties among which synchronization is to be provided and then proceeds to guide the execution in a lock-step manner by providing guaranteed and timely delivery of messages every round.

For non-concurrent behavior, we additionally require that the environment Z

Functionality \mathcal{F}_{syn}

\mathcal{F}_{syn} expects its SID to be of the form $sid = (\bar{P}, sid')$, where \bar{P} is a list of party identities among which synchronization is to be provided. It proceeds as follows.

1. At the first activation, initialize a round counter r as 1, and send a public delayed output (Init, sid) to all parties in \bar{P} .
2. Upon receiving input (Send, sid, M) from a party $P \in \bar{P}$, where $M = \{(m_i, R_i)\}$ is a set of pairs of messages m_i and recipient identities $R_i \in \bar{P}$, record (P, M, r) and output (sid, P, M, r) to the adversary.
3. Upon receiving message $(\text{Advance-Round}, sid, N)$ from the adversary, do: If there exist uncorrupted parties $P \in \bar{P}$ for which no record (P, M, r) exists then ignore this message. Else:
 - (a) Interpret N as list of messages sent by corrupted parties in the round. That is, $N = \{(S_i, R_i, m_i)\}$ where each $S_i, R_i \in \bar{P}$, m is a message, and S_i is corrupted. (S_i is taken as the sender of message m_i and R_i is the receiver.)
 - (b) Prepare for each party $P \in \bar{P}$ the list L_P^r of messages that were sent to it in round r .
 - (c) Increment the round number: $r \leftarrow r + 1$.
4. Upon receiving input $(\text{Receive}, sid)$ from some party $P \in \bar{P}$, output $(\text{Received}, sid, r, L_P^{r-1})$ to P . (Let $L_P^0 = \perp$.)

Figure 6.5: The Synchronous Communication functionality

and the adversary do not interact from the moment the first protocol message is sent until the moment where the last protocol message is delivered. Thus the stand-alone model is realized through the \mathcal{F}_{syn} -hybrid with restricted environments as described above.

As pointed out in [12], the protocol from [34] can be shown to be secure in this model. However, the construction in [34] requires $O(n)$ -rounds of communication. This is because only two parties are allowed to exchange messages according to a protocol in every round (the others send “dummy” messages). This is required to achieve non-malleability. We show how to achieve the same with $O(t)$ -rounds of communication just assuming existence of a t -round semi-honest SH-OT. Since oblivious-transfer is necessary for secure-computation, this shows that our construction is round-optimal.

To show feasibility, we construct a puzzle in the \mathcal{F}_{syn} -hybrid by relying on one-way functions. We need to alter the definition of the real-world experiment of the concurrent puzzle execution slightly to fit the stand-alone UC-model. Recall that (as in the CRS-model), the puzzle-environment is not allowed to interact with the \mathcal{F}_{syn} ideal-functionality. However, here we allow the environment to access the \mathcal{F}_{syn} -hybrid. Furthermore, we allow the simulator to rewind the environment to any previous state. Informally speaking, these changes does not affect security. Recall that in the puzzle lemma, to prove correctness, we incorporated the entire execution in a concurrent puzzle execution, where we considered puzzle-environments that internally emulated all the honest-parties execution (outside the puzzle executions) and the environment. However, in this model, since the environment is denied communication with the adversary during protocol execution, the puzzle-environment is only require to model honest parties communication. Since these executions and \mathcal{F}_{syn} are internally emulated in our final simulator and is allowed to rewind those executions, we can allow simulators for the concurrent puzzle execution that rewinds environments.

Theorem 12. *Assume the existence of t -round stand-alone secure SH-OT protocol. Then, for every well-formed ideal functionality \mathcal{F} , there exists a $O(t)$ -round protocol π that realizes $\hat{\mathcal{F}}$ with Stand Alone UC-security.*

As mentioned before, we construct a UC-puzzle in this model and obtain the theorem as corollary of Theorem 3. The puzzle we consider is identical to the one

consider for QPS-UC-Security, with the exception that, we use a standard one-way function instead of a one-way function that is invertable in \mathcal{PQT} . Let f be a one-way function. Define $\mathcal{P}_{\parallel}(\langle P_s, V_s \rangle, \mathcal{R})$ as follows:

Protocol $\langle S, R \rangle$: $S \rightarrow R$: S picks $x \leftarrow \{0, 1\}^n$ and sends $y = f(x)$ to R

$S \leftrightarrow R$: a witness-hiding argument of knowledge of the statement that there exists x' such that $y = f(x')$

Relation: $\mathcal{R} = \{(x, y) | y = f(x)\}$

SOUNDNESS: This follows directly from the one-wayness of f and the witness-hiding property of the proof given by the sender.

STATISTICAL SIMULATABILITY: To output a witness, we require A' to compute the inverse of $y = f(x)$ for a random x . As we are in the \mathcal{F}_{syn} -hybrid, the execution proceeds in rounds, where all parties receive and send a message in each round. After A completes a puzzle execution, A' temporarily stalls the main execution and tries to extract a witness for the puzzle.

Towards this, A' rewinds A to the round where it received the challenge for the \mathcal{WI} protocol in the puzzle, say round r , and feeds a new challenge. For emulating the environment messages and \mathcal{F}_{syn} in round r , A' simply rewinds the respective parties (as it is allowed to in this model). On receiving a response from A in the r^{th} round, if A' obtains a valid challenge-response pair, then using the special-sound extractor for the \mathcal{WI} protocol, A' extracts a witness. If the extractor fails A' halts outputting fail. If A aborts without giving a valid response, A' rewinds repeatedly until it obtains a valid challenge-response pair and then continues with the main execution.

We remark that even if A completes several puzzle interactions concurrently (or, more accurately, in parallel), A' can focus on extracting the witnesses one at a time, since the execution proceeds in rounds and rewinding in one interaction can not rush A' into extracting witnesses for other puzzle interactions (as is the case with concurrent zero-knowledge simulators; see next chapter).

Since the rewinding does not affect the main execution, unless the special-soundness extractor fails, the execution proceeds identically to the real-execution. But this happens only with negligible probability and hence the simulation by A' is statistically close to the real execution. Finally, to show that the simulator runs in polynomial time, we rely on the fact that if p is the probability that A completes the puzzle execution successfully, then A' needs to rewind it in expectation $1/p$ times. Overall, for every puzzle A' in expectation employs $p \times 1/p = O(1)$ rewinds to extract the witness. Since there are only polynomially many interactions, the expected running time of A' is polynomial.

CHAPTER 7

LOWER BOUNDS FOR NON-UNIFORM UC-SECURITY

In the previous section, we constructed a puzzle in the Non-Uniform UC model relying on the existence of evasive promise-problem in **BPP**. In this section, we show that this assumption is in fact necessary to achieve Non-Uniform UC security. More precisely, we show that if there exists a protocol $\Pi = (\Pi_{com}, \Pi_{decom})$ that realizes the bit commitment functionality with Non-Uniform UC security, then there exists an *evasive* promise problem $\Delta = (\Delta_Y, \Delta_N)$ in **BPP**.

To prove the lower-bound, we consider environments that communicate with the adversary. However, it should be appreciated that the same lower-bounds hold even in the case of just self-composition where the environment does not communicate with the adversary. This follows since, as pointed out by Lindell [54], the environment can always communicate with the adversary through the functionality—in our setting, if the environment wishes to send a message to the adversary, it can ask an honest party to commit and decommit that message to the adversary and the adversary can commit and decommit the message to an honest party in order to communicate a message to the environment. The proof idea is similar to [15], but we exploit the “reverse properties”.

Functionality \mathcal{F}_{com}

1. Upon receiving input $(\text{Commit}, sid, P_j, b)$ from P_i where $b \in \{0, 1\}$, internally record the tuple (P_i, P_j, b) and send the message (sid, P_i, P_j) to the adversary; When receiving (ok) from the adversary, output $(\text{Receipt}, sid, P_i)$ to P_j . Ignore all subsequent (Commit, \dots) inputs.
2. Upon receiving a value (Open, sid) from P_i , where a tuple (P_i, P_j, b) is recorded, send (b) to the adversary; When receiving (ok) from the adversary, output (Open, sid, b) to P_j .

Figure 7.1: Ideal Bit Commitment Functionality

Theorem 13. *If there exists a protocol Π that with securely realizes \mathcal{F}_{com} with Non-Uniform UC security, then there exists an evasive promise problem $\Delta = (\Delta_Y, \Delta_N)$ in **BPP**.*

Proof: Let $\Pi = (\Pi_{com}, \Pi_{decom})$ be the protocol that securely realizes the functionality \mathcal{F}_{com} . Consider the following execution with the environment \tilde{Z} . On activation it activates the adversary and proceeds as follows:

- First, it activates two parties C and R . It then picks a random bit b and initiates a session between C and R by selecting a session identified sid and sending $(\text{Commit}, sid, R, b)$ to C . On completion, it sends (Open, sid, b) to C .
- The \tilde{Z} interacts with the adversary as an honest receiver using Π_{com} and then followed by Π_{decom}
- Finally, \tilde{Z} outputs 1 if the adversary successfully commits and decommits to b .

Consider an adversary A , which upon activation, corrupts the receiver R and relays all messages received from C to \tilde{Z} . Since Π realizes \mathcal{F}_{com} with Non-Uniform UC security, there exists a simulator S^* such that for all environments Z , the following distributions are indistinguishable over $n \in \mathbb{N}$:

- $\{\text{EXEC}_{\Pi, A, Z}(n)\}_{n \in \mathbb{N}}$
- $\{\text{EXEC}_{\Pi_{\text{ideal}}, S^*, Z}^{\mathcal{F}_{com}}(n)\}_{n \in \mathbb{N}}$

where Π_{ideal} is the protocol, where the parties directly interact with \mathcal{F}_{com} using their inputs.

Towards obtaining an *evasive* promise problem in **BPP**, we first construct an interactive \mathcal{PPT} machine M that is “easy” to solve for n.u. \mathcal{PPT} machines but “hard” for \mathcal{PPT} machines. Then using M , we show how to construct the required language. Informally speaking, we say that M is easy to solve for P if after an execution between P and M , M outputs 1 with high probability and it is hard to solve if M outputs 1 with small probability.

The machine M on input 1^n , proceeds as follows: It incorporates \tilde{Z} , \mathcal{F}_{com} and the committer C and emulates the ideal-world experiment in the following manner:

- All messages intended for the adversary and receiver R and forwarded outside with the recipient information encoded appropriately.
- All messages received are fed internally to the respective parties as encoded in the message.
- Finally, M outputs what \tilde{Z} outputs in the internal emulation.

Claim 14. *There exists a n.u. \mathcal{PPT} machine \tilde{P} such that*

$$\Pr \left[\text{OUTPUT}_M(\langle \tilde{P}, M \rangle(1^n)) = 1 \right] \geq 1 - \nu(n)$$

where $\nu(\cdot)$ is a negligible function.

Proof: Consider the machine \tilde{P} internally incorporates S^* and forwards all messages from S^* outside with the recipient information encoded appropriately (just as M does). Since S^* is a non-uniform \mathcal{PPT} machine, so is \tilde{P} . We claim that this \tilde{P} satisfies the conditions of the claim.

Recall that in an execution with \tilde{Z} and A , \tilde{Z} outputs 1 if A commits and decommits to the bit b that it provided as input to C . Since A controls the receiver R and relays the messages back-and-forth between C and the environment, and C commits and decommits to b , it holds that

$$\Pr \left[\text{EXEC}_{\Pi, A, \tilde{Z}}(n) = 1 \right] = 1$$

Since S^* simulates A , we have that

$$\Pr \left[\text{EXEC}_{\Pi_{\text{ideal}}, S^*, \tilde{Z}}^{\mathcal{F}_{\text{com}}}(n) = 1 \right] \geq 1 - \nu(n)$$

for some negligible function $\nu(\cdot)$.

We conclude the proof of the claim by observing that the internal emulation by M when interacting with \tilde{P} , proceeds identically to an execution with \tilde{Z} and S^* in the ideal-world. Therefore,

$$\Pr \left[\text{OUTPUT}_M(\langle \tilde{P}, M \rangle(1^n)) = 1 \right] = \text{EXEC}_{\Pi_{\text{ideal}}, S^*, Z}^{\mathcal{F}_{\text{com}}}(n) \geq 1 - \nu(n)$$

□

Claim 15. *For every \mathcal{PPT} machine P , there exists a negligible function $\epsilon(\cdot)$ such that $\Pr [\text{OUTPUT}_M(\langle P, M \rangle(1^n)) = 1] \leq \frac{1}{2} + \epsilon(n)$.*

Proof: On a high-level, to prove this claim, we construct for every machine P interacting with M an adversary \tilde{A} that violates the security of Π realizing \mathcal{F}_{com} and thus arrive at a contradiction.

Consider the following execution with the environment \tilde{Z}' . On activation it activates the adversary and proceeds as follows:

- First, it activates two parties C and R . It then picks a random bit b and initiates a session between C and R by selecting a session identified sid and sending $(\text{Commit}, sid, R, b)$ to C .
- On completion, it sends (Open, sid, b) to C and random bit b^* to the adversary.
- Finally, \tilde{Z}' outputs 1 if C successfully commits and decommits the bit b^* to R .

Given any machine P that interacts with M , we construct an adversary A . The adversary A internally incorporates P and proceeds as follows:

- A corrupts the committer C .
- On receiving $(\text{Commit}, sid, R, b)$ from \tilde{Z}' , A ignores the message and feeds the message $(\text{Receipt}, sid, C)$ to P .
- All messages received by A from the external receiver R are fed internally to P , with the exception that A changes the sender from R to the environment \tilde{Z} (as internally emulated by M). Similarly all messages from P in the internal emulation with encoded recipient \tilde{Z} is forwarded externally R .
- On receiving b^* from \tilde{Z}' , A feeds (Open, sid, b^*) to P .

It follows from construction that

$$\Pr [\text{EXEC}_{\Pi, A, \tilde{Z}'}(n) = 1] = \Pr [\text{OUTPUT}_M(\langle P, M \rangle(1^n)) = 1]$$

Furthermore, there exists \tilde{S} such that

$$\left| \Pr [\text{EXEC}_{\Pi, A, \tilde{Z}'}(n) = 1] - \Pr [\text{EXEC}_{\Pi_{\text{ideal}}, \tilde{S}, \tilde{Z}'}^{\mathcal{F}_{\text{com}}}(n) = 1] \right| < \epsilon(n) \quad (7.1)$$

for some negligible function $\epsilon(\cdot)$.

However, observe that \tilde{S} controlling C needs to provide \mathcal{F}_{com} a bit b before \tilde{Z}' releases b^* and hence it can only succeed with probability at most a $\frac{1}{2}$. Therefore, it follows from Equation 7.1 that

$$\Pr [\text{OUTPUT}_M(\langle P, M \rangle(1^n)) = 1] \leq \frac{1}{2} + \epsilon(n)$$

□

We now turn towards constructing the promise problem.

THE PROMISE PROBLEM: Let P^* be the n.u. \mathcal{PPT} machine that solves M . We denote by P_x^* when the non-uniform advise to P^* is fixed to the string x . Define the language Δ as follows:

$$\begin{aligned}\Delta_Y &= \left\{ x \mid \Pr[\text{OUTPUT}_M(\langle P_x^*, M \rangle 1^n) = 1] \geq \frac{7}{8} \right\} \\ \Delta_N &= \left\{ x \mid \Pr[\text{OUTPUT}_M(\langle P_x^*, M \rangle 1^n) = 1] \leq \frac{2}{3} \right\}\end{aligned}$$

Claim 16. Δ is in **BPP**.

Proof: To show that Δ is in **BPP**, we construct a \mathcal{PPT} machine that checks membership in Δ . The machine A on input x , emulates $N = n^3$ independent executions of $\langle P_x^*, M \rangle$. Let X_i denote the output of M in the i^{th} iteration. A ACCEPTS x if

$$\frac{\sum_i^N X_i}{N} \geq \frac{7}{8} - \frac{1}{n}$$

and REJECTS otherwise. It follows from definition of X_i that

$$E[X_i] = \Pr[\text{OUTPUT}_{P_x^*}(\langle P_x^*, M \rangle (1^n)) = 1]$$

Hence, if $x \in \Delta_Y$, using the Chernoff-bound, it follows that:

$$\Pr\left[\left|\frac{\sum_i^N X_i}{N} - \frac{7}{8}\right| > \frac{1}{n}\right] \leq 2^{-\left(\frac{1}{n}\right)^2 N} = 2^{-n}$$

and A ACCEPTS x with probability at least $1 - 2^{-n}$. Similarly, if $x \in \Delta_N$,

$$\Pr\left[\left|\frac{\sum_i^N X_i}{N} - \frac{2}{3}\right| > \frac{1}{n}\right] \leq 2^{-\left(\frac{1}{n}\right)^2 N} = 2^{-n}$$

and A REJECTS x with probability at least $1 - 2^{-n}$. Therefore, A decides the promise-problem Δ . \square

Claim 17. For every \mathcal{PPT} machine A , there exists a negligible function $\nu(\cdot)$ such that, the probability that A on input 1^n outputs an element in the set $\{0, 1\}^n \setminus \Delta_N$ is at most $\nu(n)$.

Proof: Assume for contradiction, there exists a \mathcal{PPT} machine A and polynomial $p_1(\cdot)$ such that for infinitely many n , it holds that

$$\Pr[x \leftarrow A(1^n) : x \notin \Delta_N \cap \{0, 1\}^n] \geq \frac{1}{p_1(n)}$$

Using A , we construct a \mathcal{PPT} machine P' such that violates Claim 15

Consider the machine P' that on input 1^n proceeds as follows:

- In a preprocessing phase, P' runs the following sequence of steps $np_1(n)$ times.
 1. Run A on input 1^n . Let x be A 's output.
 2. Emulate $\langle P_x, M \rangle(1^n)$ n^3 times. Compute the fraction of times P_x convinces M , let this be p . If $p < \frac{2}{3} - \frac{1}{n}$ continue with next iteration. Otherwise, exit loop outputting x .
- Suppose, no x was output by the pre-processing phase, halt outputting \perp . Otherwise, run the code of P_x to interact with the external machine M .

We claim that P' convinces M with probability at least $\frac{2}{3} - \frac{2}{n}$ and this contradicts Claim 15 since P' is a \mathcal{PPT} machine.

It follows from description that if the pre-processing phase outputs any x , then the probability that P' convinces M is $\Pr[\text{OUTPUT}_M(\langle P_x, M \rangle(1^n)) \neq 1]$. Call x **good**, if $\Pr[\text{OUTPUT}_M(\langle P_x, M \rangle(1^n)) = 1] \geq \frac{2}{3}$, i.e. $x \in \{0, 1\}^n \setminus \Delta_N$. Next, we bound the probability that the pre-processing phase fails to output a **good** x . This happens if:

Event E_1 : The pre-processing phase outputs an x that is not **good**. If x is not **good** in a trial, then the pre-processing phase outputs x only if the check in Step 2 succeeds. Using the Chernoff-bound, this is happens with probability at most 2^{-n} . Using the union bound, we have that the probability that the pre-processing phase outputs an x that is not **good** is at most $np_1(n)2^{-n}$.

Event E_2 : A fails to output a **good** x in every iteration. Since every element outside Δ_N is **good** by definition, the probability that A does not output such an element in any of the trials is at most $\left(1 - \frac{1}{p_1(n)}\right)^{np_1(n)} \leq e^{-n}$.

Event E_3 : A outputs a good x , but the check in Step 2 fails. If x is good, then by Chernoff-bound we have that except with probability 2^{-n} , the check succeeds.

Observe that if neither of the events E_1 , E_2 or E_3 occurs, then the pre-processing phase outputs a good x . Therefore P' interacts with M using P_x for a good x and succeeds with probability at least $\frac{2}{3} - \frac{1}{n}$.

Using the union bound we have that the probability that pre-processing phase fails to output a good x is at most $np_1(n)2^{-n} + e^{-n} + 2^{-n} < \frac{1}{n}$ (for sufficiently large n). Using another union bound, we have that

$$\Pr[\text{OUTPUT}_M(\langle P', M \rangle(1^n)) \neq 1] > \frac{2}{3} - \frac{1}{n} - \frac{1}{n} = \frac{2}{3} - \frac{2}{n}$$

□

This concludes the proof of Theorem 13. □

7.1 On Existence of Evasive Sets

We just proved in the previous section that the existence of evasive sets is necessary and sufficient for achieving Non-Uniform UC-Security. Here we discuss the plausibility of this assumption based on concrete constructions.

EVASIVE SETS FROM UNIFORM HASH FUNCTIONS. The first construction we consider will rely on the existence of a uniform hash function that is collision resistant [5, 65]. Recall that hash functions are functions that are length compression. A uniform hash-function is one that is computable by a deterministic polynomial time algorithm.

Definition 23. A uniform hash function H is said to be collision-resistant if for every uniform \mathcal{PPT} algorithm A , there exists a negligible function $\nu(\cdot)$ such that, for all n , the probability that A on input 1^n outputs a pair $x \neq x' \in \{0, 1\}^n$ such that $H(x) = H(x')$ is at most $\nu(n)$.

Theorem 14. Assume the existence of a uniform hash function H that is collision resistant. Then there exists an evasive set that is decidable in polynomial time.

Proof: Let H be the collision-resistant hash function. Define

$$\Delta = \{(1^n, x, x') | x \neq x' \in \{0, 1\}^n \text{ and } H(x) = H(x')\}$$

Since any member in Δ yields a valid collision for H , an adversary that outputs a member of $\Delta \cap \{0, 1\}^n$ for infinitely many n with non-negligible probability, also violates the collision-resistant property of H . Therefore, Δ is an evasive-set. Furthermore, since H is computable in strict polynomial time, any member of Δ can also be checked in deterministic polynomial time. \square

EVASIVE SETS FROM HARD-ON-THE-AVERAGE LANGUAGE IN **NE**. We now provide a construction of complexity flavor. We show how to construct an evasive set by relying on a language decidable in **NE** (**NE** = **Ntime**($2^{O(n)}$)) that is *hard-on-the-average* for probabilistic exponential time machines (\mathcal{PET}).

Definition 24. A language L is *hard-on-the-average* for the class of machines \mathcal{C} , if for any machine $M \in \mathcal{C}$, there exists a $\delta < \frac{1}{6}$, such that for all sufficiently large n ,

$$\Pr[x \leftarrow \{0, 1\}^n : M(x) = L(x)] \leq \frac{1}{2} + \delta$$

Theorem 15. Assume the existence of hard-on-the-average languages for \mathcal{PET} in **NE**. Then there exists an evasive-set in **BPP**.

Proof: Let $L \in \mathbf{NE}$ be a hard-on-the-average for \mathcal{PET} machines. Since $L \in \mathbf{NE}$, we have that every x that is a member of L has a witness of length $2^{|x|}$. Set $k(n) = \lceil \frac{n}{3} + 1 \rceil$.

Define Δ as follows: the tuple $(1^n, (x_1, w_1), (x_2, w_2), \dots, (x_{k(n)}, w_{k(n)}))$ is in Δ if all x_i 's are distinct, for every $i \in [k(n)]$, $|x_i| = \lfloor \log n \rfloor$ and w_i is a witness that $L(x_i) = 1$.

Membership in Δ can be checked in polynomial time since the statement and witness are of length $O(\log n)$. To prove that it is evasive, assume for contradiction that there exists a uniform \mathcal{PPT} algorithm A and polynomial $p(\cdot)$ such that for infinitely many n , the following holds: On input 1^n , A outputs a member of Δ of the form $(1^n, \dots)$ with probability $p(n)$. Using A , we construct a \mathcal{PET} machine M that decides L on at least $\frac{2}{3}$ fraction of strings from $\{0, 1\}^n$ for infinitely many n .

Consider M that on input x ($|x| = n$) proceeds as follows: M runs A $n^2 p(n)$ times on input 1^n . If it obtains a member in Δ , it outputs 1 if x, w is part of the output and 0 otherwise. It follows that whenever A outputs a member in Δ , M answers correctly on at least $2 \frac{k(2^n)}{2^n} \geq \frac{2}{3} + 2^{-n+1}$ fraction of the strings in $\{0, 1\}^{2^n}$. Furthermore, the probability that A fails to output a member of Δ in all the $n^2 p(n)$ tries is at most e^{-n^2} . Therefore, using the union bound we have that

$$\Pr[x \leftarrow \{0, 1\}^n : M(x) = L(x)] \geq \frac{2}{3} + 2^{-n+1} - e^{-n^2} \geq \frac{2}{3}$$

Thus, L is not hard-on-the-average for \mathcal{PET} . \square

An equivalent formulation is to assume a unary language in \mathbf{NP} that is hard-on-the-average for \mathcal{PPT} machines.

EVASIVE SETS FROM WORST-CAST HARDNESS OF $\mathbf{NE} \cap \text{coNE}$. In this section, we construct an evasive set from a language in $\mathbf{NE} \cap \text{coNE}$ that is worst-cast hard for \mathcal{PET} machines. We require worst-case hardness in the following sense.

Definition 25. A language L is worst-case hard for \mathcal{PET} , if for all \mathcal{PPT} machines P , there exists a $N > 0$, such that for every $n > N$, there exists x_n such that

$$\Pr[P(x_n) = L(x_n)] < \frac{2}{3}$$

We remark that this is slightly stronger than the traditional worst-case hardness definitions which does not require that there be a hard input for every length n , but only for infinitely many n . Now, we proceed to construct an evasive set Δ from a language L that is worst-case hard for \mathcal{PET} .

Theorem 16. Assume the existence of language in $\mathbf{NE} \cap \text{coNE}$ that is worst-case hard for \mathcal{PET} . Then there exists an evasive-set in \mathbf{BPP} .

Proof: Let $L \in \mathbf{NE} \cap \text{coNE}$ that is worst-case hard for \mathcal{PET} machines. Since $L \in \mathbf{NE} \cap \text{coNE}$, we have that every x has either a witness of length $2^{|x|}$ proving membership in L or a witness of length $2^{|x|}$ proving membership of $\{0, 1\}^* \setminus L$. Let $k(n) = 2^{\lfloor \log n \rfloor}$.

Define Δ as follows: the tuple $(1^n, (x_1, w_1), (x_2, w_2), \dots, (x_{k(n)}, w_{k(n)}))$ is in Δ if all x_i 's are distinct, for every $i \in [n]$, $|x_i| = \lfloor \log n \rfloor$ and w_i is either a witness that $L(x_i) = 1$ or $L(x_i) = 0$.

Membership in Δ can be checked in polynomial time since the statement and witness are of length $O(\log n)$. To prove that it is evasive, assume for contradiction that there exists a uniform \mathcal{PPT} algorithm A and polynomial $p(\cdot)$ such that for infinitely many n , the following holds: On input 1^n , A outputs a member of Δ of the form $(1^n, \dots)$ with probability $p(n)$. Using A , we construct a \mathcal{PET} machine M that decides L in the worst-case and arrive at a contradiction.

Consider M that on input x ($|x| = n$) proceeds as follows: M runs A $np(n)$ times on input 1^n . If it obtains a member in Δ , it finds a witness for x from the output and checks outputs $L(x)$ using the witness. It follows that whenever A outputs a member in Δ , M decides L on all inputs from $\{0, 1\}^{2^n}$. Furthermore, the probability that A fails to output a member of Δ in all the $n^2 p(n)$ tries is at most e^{-n} . Therefore, using the union bound we have that for infinitely many n it holds that:

$$\forall x \in \{0, 1\}^n, \Pr[M(x) = L(x)] \geq 1 - e^{-n} \geq \frac{2}{3}$$

Thus, L is not worst-case hard for \mathcal{PET} . □

CHAPTER 8

EFFICIENT CONCURRENT ZERO-KNOWLEDGE

In the previous chapters, we provided constructions to realize any functionality that remains secure under arbitrary composition. The focus was to provide a unified framework to help construct protocols in the different models both with and without trusted setup. In this chapter, we consider a relatively simpler task of constructing protocols that are secure under *self-composition* (i.e. when several instances of the same protocol run concurrently) but focusing on obtaining round-efficient protocols without assuming any trusted set-up, i.e. in the plain model. In particular, we show how to obtain $O(1)$ -round zero-knowledge proofs and argument that are secure in this model of concurrent security. First, we begin with a more detailed discussion on zero-knowledge and concurrent security under self-composition.

Zero-knowledge interactive proofs [38] are paradoxical constructs that allow one player (called the Prover) to convince another player (called the Verifier) of the validity of a mathematical statement $x \in L$, while providing *zero additional knowledge* to the Verifier. This is formalized by requiring that the view of every “efficient” adversary verifier V^* interacting with the honest prover P be simulated by an “efficient” machine S (a.k.a. the *simulator*). The idea behind this definition is that whatever V^* might have learned from interacting with P , he could have actually learned by himself (by running the simulator S). As “efficient” adversaries normally are modelled as probabilistic polynomial-time machines (\mathcal{PPT}), the traditional definition of \mathcal{ZK} models both the verifier and the simulator as \mathcal{PPT} machines. In this paper, we investigate alternative models of efficient adversaries—in particular, as in [63], we model adversaries as probabilistic quasi-polynomial time machines (\mathcal{PQT}).

CONCURRENCY AND \mathcal{ZK} . The notion of concurrent \mathcal{ZK} , first introduced and achieved, by Dwork, Naor and Sahai [27] considers the execution of zero-knowledge proofs in an asynchronous setting and concurrent setting. More precisely, we consider a single adversary mounting a coordinated attack by acting as a verifier in many concurrent executions. Concurrent zero-knowledge proofs are significantly harder to construct (and analyze).

Since the original protocols by Dwork, Naor and Sahai (which relied on so called “timing assumptions”), various other protocols have been obtained based on different set-up assumptions (e.g., [29, 22, 16]). On the other hand, in the “plain” model without any set-up Canetti, Kilian, Petrank and Rosen [17] (building on earlier works by [48, 74]) show that concurrent \mathcal{ZK} proofs for non-trivial languages, with so called “black-box” simulators, require at least $\Omega(\frac{\log n}{\log \log n})$ number of communication rounds. Richardson and Kilian [73] constructed the first concurrent zero-knowledge argument in the standard model. Their protocol which uses a black-box simulator requires $O(n^\epsilon)$ number of rounds. Kilian and Petrank [47] later obtained a round complexity of $\tilde{O}(\log^2 n)$, and finally Prabhakaran, Rosen and Sahai [70] essentially closed the gap by obtaining a round complexity of $\tilde{O}(\log n)$.

All of the above results rely on the traditional modeling of adversaries as \mathcal{PPT} machines. Thus, it is feasible that there exists some super-polynomial, but “well-behaved”, model of adversaries that admits constant-round concurrent \mathcal{ZK} proofs.

CONCURRENT \mathcal{ZK} W.R.T SUPER-POLYNOMIAL ADVERSARIES. The lower bound of [48] shows that only languages decidable in probabilistic subexponential-time have 4-round concurrent black-box zero-knowledge arguments w.r.t to probabilistic subexponential-time adversaries. On the other hand, [63] constructs constant-round concurrent zero-knowledge arguments w.r.t \mathcal{PQT} verifiers (and consequently also simulators); however the soundness condition of those argument systems only holds w.r.t. \mathcal{PPT} adversaries—in fact, the simulator succeeds in its simulation by breaking the soundness condition of the argument system. Additionally, it is noted in [63] that there exist 3-round concurrent \mathcal{ZK} proofs w.r.t. exponential-time adversaries (as any witness indistinguishable proof is also zero-knowledge with respect to exponential-time verifiers). Finally, [73] claimed that a constant-round version of their protocol remains secure w.r.t \mathcal{PQT} adversaries, when considering a “benign” type of concurrent adversary (which never sends any invalid messages and has a fixed—i.e., non-adaptively chosen—scheduling), but as far as we know a proof of this has never appeared.

Thus, the above results leave open the question of whether there exist $r(n)$ -round concurrent black-box zero-knowledge proofs w.r.t super-polynomial, but sub-exponential, adversaries, as long as $4 < r(n) < \log n$. In particular,

Does there exist constant-round concurrent zero-knowledge arguments w.r.t. \mathcal{PQT} (or even sub-exponential time) adversaries?

8.1 Results

Our main result answers the above question in the affirmative. Let \mathcal{PQT} denote the class of probabilistic quasi-polynomial time machines, i.e., randomized machines that run in time $n^{\text{poly}(\log(n))}$. Let $\omega(\mathcal{PQT})$ denote the class of *probabilistic super quasi-polynomial time* machines, i.e. randomized machines that run in time $n^{\omega(\text{poly}(\log(n)))}$.

Theorem 17 (Main Theorem). *Assume the existence of claw-free permutations w.r.t \mathcal{PQT} . Then, every language in \mathbf{NP} has an $O(1)$ -round perfect concurrent black-box \mathcal{ZK} argument w.r.t \mathcal{PQT} .*

In addition, we show:

Theorem 18. *Assume the existence of one-way functions that are secure w.r.t $\omega(\mathcal{PQT})$ and collision-resistant hash functions that are secure w.r.t $\omega(\mathcal{PQT})$. Then, every language in \mathbf{NP} has an $O(1)$ -round concurrent computational black-box \mathcal{ZK} proof w.r.t \mathcal{PQT} .*

Theorem 19. *Assume the existence of one-way function that are secure w.r.t $\omega(\mathcal{PQT})$. Then, every language in \mathbf{NP} has an $O(1)$ -round concurrent computational black-box \mathcal{ZK} arguments w.r.t \mathcal{PQT} .*

Theorem 20. *There exists an $O(1)$ -round concurrent perfect \mathcal{ZK} proof w.r.t \mathcal{PQT} for Graph Non-Isomorphism and Quadratic Non-Residuosity*

We emphasize that in the above theorems, “ \mathcal{ZK} proofs and arguments w.r.t \mathcal{PQT} ” refer to proofs / arguments where both the soundness condition and the \mathcal{ZK} condition holds w.r.t to \mathcal{PQT} adversaries; in particular, for the \mathcal{ZK} property we also require that the distinguishability gap is smaller than the inverse of any quasi-polynomial function.

A NOTE ON EXPECTED RUNNING-TIME. In contrast to earlier work on concurrent zero-knowledge (e.g. [73, 47, 70]), our simulators run in *expected* \mathcal{PQT} . This is

inherent: by the work of Barak-Lindell [4] it follows that only languages decidable in \mathcal{PQT} have constant-round \mathcal{ZK} protocols w.r.t \mathcal{PQT} if requiring a strict \mathcal{PQT} simulator (let alone the question of concurrency). In particular, this shows that none of the previous simulation techniques can be extended to get constant-round protocols w.r.t \mathcal{PQT} (at least when requiring that the output of the simulation is also indistinguishable for \mathcal{PQT}).¹

ADDITIONAL RESULTS. Finally, we mention that our techniques apply also to concurrent \mathcal{ZK} proofs w.r.t \mathcal{PPT} . As a result we obtain the first concurrent *perfect* \mathcal{ZK} arguments/proofs w.r.t \mathcal{PPT} .

Theorem 21. *Assume the existence of claw-free permutations (w.r.t \mathcal{PPT}). Then, every language in \mathbf{NP} has an $O(n^\epsilon)$ -round perfect concurrent black-box \mathcal{ZK} argument w.r.t \mathcal{PPT} , for every $\epsilon > 0$.*

Theorem 22. *For every $\epsilon > 0$, there exists a $O(n^\epsilon)$ -round concurrent perfect \mathcal{ZK} proof for Graph Non-Isomorphism and Quadratic Non-Residuosity.*

As an additional contribution, we believe that both our protocols and their analysis provides the simplest proof of the existence of concurrent \mathcal{ZK} proofs (w.r.t \mathcal{PPT}).²

PQT v.s. PPT: WHAT IS RIGHT MODEL FOR ADVERSARIAL COMPUTATION? Recall that to show that \mathcal{ZK} is closed under sequential composition, the original definition of \mathcal{ZK} was extended to consider *non-uniform* \mathcal{PPT} adversaries [41]—in other words, in the context of \mathcal{ZK} the notion of non-uniform \mathcal{PPT} (for modeling adversaries) is more robust than simply \mathcal{PPT} . Additionally, security is guaranteed w.r.t a stronger class of adversaries. Of course, the extra price to pay is that all hardness assumptions now must hold also with respect to non-uniform \mathcal{PPT} .

In this paper we show that by considering an even stronger class of adversaries—namely \mathcal{PQT} —we get a notion that is even more robust; in particular, it is now possible to get constant-round concurrent \mathcal{ZK} protocols. Again, this requires us to

¹On the other hand, it might still be plausible that the technique of [73] can be extended to give constant-round protocols w.r.t \mathcal{PQT} , when allowing the indistinguishability gap to be a polynomial (or even some *fixed* quasi-polynomial) function.

²In a related work [68], joint with Dustin Tseng we provide a simple proof for existence of concurrent \mathcal{ZK} proofs with logarithmic round complexity.

rely on hardness assumptions against \mathcal{PQT} , but this seems like a weak strengthening of traditional hardness assumptions (especially since the known attacks on traditional conjectured hard functions require subexponential time).

A NOTE ON PLAUSIBLE DENIABILITY. The notion of \mathcal{ZK} is traditionally associated with *plausible deniability*—i.e., that the interaction leaves “no trace” which the verifier can use later to convince that the interaction took place. Intuitively, this holds since the verifier could have executed the simulator (on its self) to generate its view of the interaction. We mention, however, that since the traditional definition of \mathcal{ZK} allows the simulator to have an arbitrary (polynomial) overhead with respect to the verifier (who’s view it is supposed to simulate), the deniability guarantee offered by traditional \mathcal{ZK} proofs is weak: consider for instance a verifier with a running-time of $t = 2^{40}$ computational steps, and a simulator with running-time, say, t^3 ; although 2^{40} is very feasible, 2^{120} seems like a stretch! The example is not hypothetic—the “tightest” concurrent \mathcal{ZK} protocols [47, 70] indeed have a running-time of t^2 not counting the time need to emulate the verifier. Additionally, as demonstrated in [56], the traditional notion of \mathcal{ZK} does not guarantee that the running-time of the simulator is (even polynomially) related to the running-time of the verifier in the view it is outputting, but rather the *worst-case* running-time of the verifier; this makes deniability even harder to argue.³

Nevertheless, in this respect, \mathcal{ZK} w.r.t \mathcal{PQT} provides even worse guarantees (as the overhead is now allowed to be quasi-polynomial).

8.2 Techniques

The concurrent \mathcal{ZK} protocols of Richardson and Kilian (RK) [73], Kilian and Petrank (KP) [47] and Prabhakaran, Rosen and Sahai (PRS) [70] rely on the same principal idea: provide the simulator with multiple possibilities (called “slots”) to rewind the verifier. If a rewinding is successful, the simulator obtains a trapdoor that allows it to complete the execution that has been rewound. The RK simulator

³In a recent work [62], joint with Pandey, Sahai and Tseng we also show how to obtain precise concurrent \mathcal{ZK} proofs. Precise zero knowledge guarantees that the view of any verifier V can be simulated in time closely related to the *actual* (as opposed to the worst-case) time spent by V in the generated view.

is “adaptive” and dynamically decides when and where to rewind, while making sure there are not too many recursive rewinding (which would result in a large running-time). On a high-level, this is done by recursively invoking the simulator, but ensuring that the number of levels of the recursion stays small (in fact, constant). On the other hand, the KP (and PRS) simulator is “oblivious”; the simulator has a fixed rewinding scheduling, thereby ensuring a fixed (and bounded) running-time. The core of the argument is then to show that every execution has a slot that is rewound at least once.

Our approach is based on the approach taken by RK. As RK, we consider an adaptive simulator that makes recursive calls to itself, while ensuring that the depth of the recursion stays small. Our actual simulation procedure is, however, quite different. On a high-level, our approach will perform a straight-line simulation until a “good” slot has been found, and then continue rewinding that slot until a trapdoor has been found. Thus, in contrast to the previous approach, we can not bound the worst-case running-time of our simulator, instead we are forced to bound the expected running-time of the simulator.

The benefit of our approach is that 1) it enables us to achieve perfect simulation, and 2) our analysis works no matter how many slots we have and what the depth of recursion is. In fact, we can achieve both of these properties while still guaranteeing the same expected running-time as RK—namely $O(m^{O(\log_r m)})$, where r is the number of slots. As a consequence, when applied to constant-round protocols (and considering a logarithmic recursive depth) we get a quasi-polynomial running time. As already mentioned, for this application, it is inherent to have an *expected* quasi-polynomial running-time.

8.3 Open questions

We have demonstrated that constant-round concurrent \mathcal{ZK} is possible w.r.t \mathcal{PQT} adversaries. Our protocol currently uses 10 communication rounds⁴. A natural open question is to either improve the round-complexity or to strengthen the 4-round lower bound of [48]. Another question is to investigate the possibility of using

⁴To obtain a 10 round protocol, we require non-interactive commitment schemes, which can be constructed from one-way-permutations. If we assume only existence of one-way functions, we get a 11-round protocol.

an even weaker (but still super-polynomial) model of computation. Rosen [74] shows that only languages in probabilistic sub quasi-polynomial time have 7-round concurrent black-box zero-knowledge arguments when adversaries are modelled as probabilistic sub quasi-polynomial time machines; thus, such protocols would require more than 7-rounds.

8.4 Black-Box Concurrent Zero-Knowledge

In this section, we define black-box concurrent zero-knowledge CONCURRENT ZERO-KNOWLEDGE Let $\langle P, V \rangle$ be an interactive proof for a language L . Consider a concurrent adversary verifier V^* that, given an input instance $x \in L$ interacts with m independent copies of P concurrently, without any restrictions over the scheduling of the messages in the different interactions with P . Let $\text{OUTPUT}_2[\langle P, V^*(z) \rangle(x)]$ denote the random variable describing the output of the adversary V^* on common input x and auxiliary input z , in an interaction with P .

Definition 26 (Black-box concurrent zero-knowledge w.r.t \mathcal{C}). *Let $\langle P, V \rangle$ be an interactive proof system for a language L . We say that $\langle P, V \rangle$ is black-box concurrent zero-knowledge w.r.t \mathcal{C} if for every functions $q, m \in \mathcal{C}$, there exists a probabilistic algorithm $S_{q,m}$, such that for every concurrent non-uniform adversary V^* that on common input x and auxiliary input z has a running-time bounded by $q(|x|)$ and opens up $m(|x|)$ executions, $S_{q,m}(x, z)$ runs in time polynomial in $(q(|x|), m(|x|), |x|)$. Furthermore, the ensembles $\{S_{q,m}(x, z)\}_{x \in L, w \in R_L(x), z \in \{0,1\}^*}$ and $\{\text{view}_{V^*}[\langle P(w), V^*(z) \rangle(x)]\}_{x \in L, w \in R_L(x), z \in \{0,1\}^*}$ are computationally indistinguishable w.r.t \mathcal{C} over $x \in L$. We say that $\langle P, V \rangle$ is perfect concurrent zero-knowledge w.r.t \mathcal{C} , if the above ensembles are identical.*

8.5 Description of the protocol

Our concurrent \mathcal{ZK} protocol (also used in [68]) is a slight variant of the precise \mathcal{ZK} protocol of [65], which in turn is a modification of the Feige-Shamir protocol [30]. The protocol proceeds in the following two stages, on a common input statement $x \in \{0, 1\}^*$ and security parameter n ,

1. In Stage 1, the Verifier picks two random strings $s_1, s_2 \in \{0, 1\}^n$, and sends their image $c_1 = f(r_1)$, $c_2 = f(r_2)$ through a one-way function f to the Prover. The Verifier sends $\alpha_1, \dots, \alpha_r$, the first messages of r invocations of a \mathcal{WI} special-sound proof of the fact that c_1 and c_2 have been constructed properly (i.e., that they are in the image set of f). This is followed by r iterations so that in the j^{th} iteration, the Prover sends $\beta_j \leftarrow \{0, 1\}^{n^2}$, a random second message for the j^{th} proof and the Verifier sends the third message γ_j for the j^{th} proof.
2. In Stage 2, the Prover provides a WI proof of knowledge of the fact that either x is in the language, or (at least) one of c_1 and c_2 are in the image set of f .

More precisely, let $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a one-way function and let the witness relation $R_{L'}$, where $((x_1, x_2), (y_1, y_2)) \in R_{L'}$ if $f(x_1) = y_1$ or $f(x_2) = y_2$, characterize the language L' . Let the language $L \in \mathcal{NP}$. Protocol **ConcZKArg** for proving that $x \in L$ is depicted in Figure 8.1.

The soundness and the completeness of the protocol follows directly from the proof of Feige and Shamir [30]; in fact, the protocol is an instantiation of theirs. (Intuitively, to cheat in the protocol a prover must “know” an inverse to either c_1 or c_2 , which requires inverting the one-way function f).

On a very high-level the simulation follows that of Feige and Shamir [30]: the simulator will attempt to rewind one of the special-sound proofs—each such proof, i.e. the challenge(β) and the response(γ) is called a slot. If the simulator gets two accepting proof transcripts, the special-soundness property allows the simulator to extract a “fake” witness r_i such that $c_i = f(r_i)$. This witness can later be used in the second phase of the protocol.

Before we proceed to describing the actual simulator, we describe and analyze a simple simulator that simulates only a restricted class of adversaries. But first, we fix some notation that will be useful in describing simulation.

We refer to the pair of a prover-challenge, and the verifier answer in Stage 1 as a slot. We say that a slot **opens** when the verifier receives a prover challenge and that the same slot **closes**, when the verifier sends the answer to the prover challenge. Formally, the opening of a slot s is a partial view h of V^* immediately

Protocol **ConcZKArg**

Common Input: an instance x of a language L with witness relation R_L .

Auxiliary Input for Prover: a witness w , such that $(x, w) \in R_L(x)$.

Stage 1:

V uniformly chooses $r_1, r_2 \in \{0, 1\}^n$.

$V \rightarrow P$: $c_1 = f(r_1), c_2 = f(r_2)$. r first messages $\alpha_1, \dots, \alpha_r$ for \mathcal{WI} special-sound proof of the statement. (called the **start** message)

either there exists a value r_1 s.t $c_1 = f(r_1)$

or there exists a value r_2 s.t $c_2 = f(r_2)$

The proof of knowledge is with respect to the witness relation R'_L

For $j = 1$ to r do

$P \rightarrow V$: Second message $\beta_j \leftarrow \{0, 1\}^{n^2}$ for j^{th} \mathcal{WI} special-sound proof. (called the **opening** of slot j)

$V \rightarrow P$: Third message γ_j for j^{th} \mathcal{WI} special-sound proof. (called the **closing** of slot j)

Stage 2:

$P \leftrightarrow V$: a perfect- \mathcal{WI} argument of knowledge of the statement

either there exists values r'_1, r'_2 s.t either $c_1 = f(r'_1)$ or $c_2 = f(r'_2)$.

or $x \in L$

The argument of knowledge is with respect to the witness relation $R_{L \vee L'}(c_1, c_2, x) = \{(r'_1, r'_2, w) | (r'_1, r'_2) \in R_{L'}(c_1, c_2) \vee w \in R_L(x)\}$.

Figure 8.1: Concurrent Perfect \mathcal{ZK} Argument for **NP**

after which the slot opens; we may now identify a slot s by the view corresponding to its opening. Analogously, the closing of a slot s is a partial view h immediately after which s closes.

8.6 Warm-up: Simulating Static Non-Aborting Adversaries

In the context of concurrent zero-knowledge protocols, a “benign” form of adversary employs an arbitrary but fixed scheduling of messages. Such an adversary is referred to as a *static* adversary. A *non-aborting* adversary is one that never causes an honest prover to abort in an execution. In this section, we describe a simulator that simulates only *static non-aborting* adversaries.

We consider the protocol described in the previous section where r is set to 2, i.e. we have 2 slots for every session. The simulator is defined recursively in the following manner: On the recursive depth ℓ , from some partially view h of messages in an execution, the simulator feeds random Stage 1 messages to V^* . Whenever the simulator receives the closing of slot s corresponding to a session for which the simulator has not extracted a fake witness yet, it decides whether or not to rewind the slot as follows:

- If s is a prefix of the partial view h (i.e. it corresponds to a session that started at a higher recursive call), then the simulator continues simulation at depth ℓ .
- If the start message corresponding to the session containing s occurs after the partial view h , then it decides to rewind s depending on the number of new sessions that started between the challenge message and the response message of s . If the number of sessions is “small” (where small is defined based on the depth ℓ), the simulator begins rewinding the slot.

To rewind a slot s at depth ℓ , S first rewinds V^* up until the point where V^* expects the opening of s . Then, it sends a new challenge β^* for slot s and recursively invokes itself on recursive depth $\ell+1$ from partial view s . It continues the simulation until the slot s closes (i.e. obtains a response for the new challenge β^*). On obtaining a valid response γ^* , the simulator using the special-soundness extractor, extracts

a “fake” witness and continues its simulation (on depth ℓ). To generate Stage 2 messages, S relies on the “fake” witness extracted in the rewinds. If a “fake” witness has not yet been extracted it halts and outputs fail.

The basic idea behind the simulation is similar to [73]: if we define “small” appropriately we can ensure that some slot of every session is rewound and the running time is bounded. Let “small” at recursive depth ℓ be set as $\frac{m}{2^{\ell+1}}$, where m is the number of sessions. We explain below informally that the simulator always extracts a witness (i.e. never fails). The proof of indistinguishability follows using a standard hybrid argument and is presented for the actual simulator in the next section. For the warm-up simulator, we only analyze running time.

Proof Intuition: From the description, it follows that the simulator fails whenever the simulator needs to provide a Stage 2 message for a session for which it has not extracted a witness yet. We argue that this never happens. Suppose the simulation fails at depth ℓ starting from partial view h on reaching Stage 2 of session i . There are two cases depending on where the session started.

- Suppose session i started after partial view h , then it will hold that some slot of the session is rewound and witness extracted. Recall that the simulator rewinds every the slot of a session that starts after h , if fewer than $\frac{m}{2^{\ell+1}}$ new sessions start within it. Recall that the simulation at level $\ell > 0$ is a rewinding of a slot at level $\ell - 1$ and the adversary employs static scheduling; so, at most $\frac{m}{2^\ell}$ new sessions start in level ℓ . Since session i starts after h , both the slots must have completed before the simulation reaches Stage 2 and at least one of them have fewer than $\frac{m}{2^{\ell+1}}$ new sessions within it and the simulator would have chosen this slot to rewind. Finally, since the adversary is non-aborting, the rewinding of the slot always provides a second challenge-response for the slot and the simulator would have extracted a witness for that session. Thus, this case does not occur.
- If the session i started at a higher recursive call, say depth $\ell' < \ell$, we argue that the simulator would have failed at depth ℓ' before it entered depth ℓ . Recall that the simulation at any depth $\ell^* > 0$ is a rewinding of some slot that occurred at depth $\ell^* - 1$. Since the adversary employs a static scheduling, every message that occurs in depth ℓ^* must have occurred at depth $\ell^* - 1$. This implies that the simulation should have reached the Stage 2 of session

i at depth ℓ' . Furthermore, by assumption, it could not have extracted a witness for session i at depth ℓ . Hence, it could not have had a witness at depth ℓ' and failed there before entering depth ℓ .

Running-time analysis: To analyze the running time, we show inductively that, conditioned on the simulator never failing, the running time of the simulator at depth ℓ is bounded by $O(m)^{\log m+1-\ell}$. Setting, $\ell = 0$, we obtain that the running time of the simulator is $O(m)^{\log m}$.

- **Base case:** At depth $\log m$, since there are no more recursive calls and the simulator does not fail, the total running time of S is bounded by the maximum number of messages in any session and this is $O(m)$.
- **Induction step:** Suppose that the statement is true for depth ℓ . At depth $\ell - 1$, the running time can be computed by bounding the number of recursive calls made to depth ℓ and applying the induction hypothesis. There are at most m sessions and if the simulator does not fail, at most one recursive call is made for every session. Therefore, the total number of recursive calls made to depth ℓ is bounded by m . As every session has only 2 slots, the time spent in simulating the messages at depth $\ell - 1$ is bounded by $O(m)$. Therefore, the time spent at depth $\ell - 1$ is bounded by

$$\begin{aligned} O(m) + m(\text{Time spent at depth } \ell) &= O(m) + m \times O(m)^{\log m+1-\ell} \\ &= O(m)^{\log m+1-(\ell-1)} \end{aligned}$$

This concludes the induction step.

Thus, we have shown a protocol with 2 slots where all static non-aborting adversaries can be simulated.

SIMULATING GENERAL ADVERSARIES.

Next, we analyze why the simulator fails on general adversaries. The simulator fails at depth ℓ while simulating from some partial view h , if

Simulator fails to extract witness for a session that started after h . For static adversaries, we argued that some slot of every execution that starts

after h will be rewound. For this, we relied on the fact that the total number of new executions that start at depth ℓ is bounded and the adversary never aborts. When we consider general adversaries, neither of these properties hold. To resolve this problem, we can allow the simulator to repeatedly rewind a slot until it extracts a witness and cutting off whenever the adversary opens too many new executions or aborts on the slot. This ensures that every session that begins at a particular depth, some slot is chosen at the same depth to be recursively rewound and the simulator extracts a witness if the recursion goes to completion. Furthermore, we show that this modification causes the simulator to run in expected polynomial time rather than strict polynomial time (which is inevitable if we want to achieve perfect simulation[4]).

Simulator fails to extract witness for a session that starts in h . For static non-aborting adversaries, every message that occurs at depth ℓ occurs at depth $\ell - 1$ and hence the simulator can fail on an execution only at the depth it started. For general adversaries this statement no longer holds. However, it suffices to modify the simulator's procedure only in the case when it reaches the Stage 2 of a session that starts in h , say depth $\ell^* < \ell$. There are two cases depending on where the challenge of the last (second) slot was received and we show how to modify the simulator in each of these cases.

- If the challenge message of the last slot did not start at the depth where the execution started, i.e. at a depth $\ell' > \ell^*$, then we store the challenge-response pair obtained from the last slot and restart simulation at depth $\ell^* + 1$. This modification allows the simulator to make progress w.r.t session i at depth ℓ^* since the next time it reaches Stage 2 of session i , a second challenge-response pair for the last slot must occur and the simulator can extract a witness with this and the one stored (as they share the same start message from depth ℓ^*).
- If the challenge-message of the last slot occurred at the depth where the execution started, i.e. depth ℓ^* , then the previous modification does not help as the challenge-message remains the same for every simulation at depth $\ell^* + 1$. To avoid this case, we take a different approach. We add a third slot and still require that the simulator rewinds some slot among

the first 2 slots. This modification ensures that this case does not occur as the simulator is guaranteed to extract a witness for session i before it reaches the last (or 3rd) slot.

We show in the next section that essentially by incorporating these modifications, we can construct a simulator that works for all adversaries.

We now proceed to formally describe our simulator and prove its correctness.

8.7 The Simulator

Our simulator is defined recursively in the following manner. Given the view h' of V^* , we call a prefix h of h' ℓ -good if the number of new executions that open in h' after h is at most $\frac{m}{(r-1)^\ell}$ (recall that m is an upper bound on the number of executions started by V^*). Given a partial view h' after which we have the closing of a slot s , we say that the slot is ℓ -good in h if s is a d -good prefix of h' (Recall that s stands for the prefix after which slot s opens). Our simulator maintains a collection of partial views in $\bar{\mathcal{H}}$ so that if two challenge-response pair occurs for any slot, it can use the special-soundness extractor to extract a witness.

Now, on recursive level $\ell > 0$, starting from a view h , S^{V^*} feeds messages to V^* until a slot s of an execution that started after h closes and the slot is $\ell + 1$ -good for the current view h' ; whenever this happens, it rewinds V^* back to the point when s opened, and invokes itself recursively at level $\ell + 1$. It continues rewinding until it gets m partial transcripts, each time appending the partial transcript to $\bar{\mathcal{H}}$. If at any instant, the current view h' and any view h^* in $\bar{\mathcal{H}}$ contain two challenge-response pairs for the same slot (with same start message for which a witness has not been extracted yet), then the simulator applies the special soundness extractor on the two transcripts; if the extractor outputs a valid witness corresponding to the j^{th} proof proved using $\langle P, V \rangle$, the witness is stored. Furthermore, at each recursive level $\ell \geq 1$ (i.e., on all recursive levels except the first one), if h is not a ℓ -good prefix of the current view h' (i.e., if the number of new executions that start exceeds $\frac{m}{(r-1)^\ell}$ or V^* aborts on the slot s that opened at h , the recursive procedure aborts). Finally, whenever V^* is expecting a Stage 2 message, S^{V^*} checks 1) if a witness has been extracted; then, it simply uses the witness to generate the Stage

2 message, or 2) if a witness has not been extracted and the execution began at a prefix of h , then it returns the view to the level where the execution began. Otherwise S^{V^*} halts outputting fail.

A formal description of our simulator can be found in Figure 8.2. Let $d = \lceil \log_{r-1} m \rceil$, the maximum depth of recursion. By construction we have that the simulation does not go beyond d recursive levels.

PROCEDURE $S^{V^*}(1^n, \ell, h, \bar{\mathcal{H}})$:

On input the recursive level ℓ and the partial view h of V^* , and collection of partial views $\bar{\mathcal{H}}$ proceeds as follows. Let $h' = h$.

Repeat the following:

- If $\ell > 0$ and h' is the closing of the slot opened at h , return h .
- If $\ell > 0$ and the partial view h is not d -good, restart simulation at current depth from h .
- If h' is the closing of a slot s corresponding to a session that started after h and s is $\ell + 1$ -good for h' : let $i = 0$; repeat the following until $i = m$:
 - Let $h^* = S^{V^*}(1^n, \ell + 1, s, \bar{\mathcal{H}})$. If h^* is the closing of the last slot of a session that opened at a prefix of h , return h^* .
 - Otherwise, if $h^* \neq \perp$, let $i = i + 1$, append h^* to $\bar{\mathcal{H}}$.
- If V^* is expecting to hear a Stage 1 message, \tilde{S}^{V^*} samples a random message of length $l(n)$, and updates h' accordingly.
- If V^* is expecting to hear a Stage 2 message in the j^{th} proof, \tilde{S}^{V^*} checks if there exists a view h^* in $\bar{\mathcal{H}}$ such that the special soundness extractor on the pair of transcripts h' and h^* outputs a witness r for the j^{th} session. If so, it uses r to generate the Stage 2 message, updates h' accordingly and continues. If not, it returns h' if session j started before h and otherwise halts outputting fail.

If in the simulation, S queries V^* more than 2^n times, it proceed as follows: Let h denote the view reached in the “main-line” simulation (i.e., at depth 0 of the recursion). Continue the simulation in a “straight-line” fashion from h by using a brute-force search to find a fake witness each time Stage 2 of a session i is reached.

Figure 8.2: Simulator for Perfect Concurrent Zero-Knowledge Argument $\langle P, V \rangle$

8.8 Analysis of the Simulator

To prove correctness of the simulator, we show that the output of the simulator is correctly distributed and its expected running-time is bounded. We first prove in Claim 18 that the simulator never aborts. Using Claim 18, we show in Proposition 3 that the output distribution of the simulator is correct. In Proposition 4, we show that the expected running time of the simulation is at most $\text{poly}(m^d r^d)$. Throughout this proof we assume without loss of generality the adversary verifier V^* is deterministic (as it can always get its random coins as part of the auxiliary input).

SIMULATION NEVER FAILS

Claim 18. *For every $x \in L$, $S^{V^*}(x, z)$ never halts outputting fail.*

Proof: As a first step, we show that for every partial view h and depth ℓ , if the simulation at depth ℓ starts simulating from view h and reaches the opening of the last slot corresponding to a session i that starts after h , then it must be the case that the simulator extracted a witness corresponding to session i .

Fix a particular history h , depth ℓ and session i that starts after h and completes $r-1$ slots. Furthermore, let us assume that the simulator has extracted the witness of all sessions that complete $r-1$ session before session i does. We show inductively that the simulator extracts a witness for session i .

Recall that if at depth ℓ , the current view is not ℓ -good then the simulation is cut off. So, if the simulation reaches the opening of the last or r^{th} slot of session i , then at this instant, the current view is ℓ -good and there is some slot s in the first $r-1$ slots that has fewer than $\frac{m}{(r-1)^{\ell+1}}$ new sessions started within, i.e. s is $\ell+1$ -good. This slot is chosen by the simulator to be recursively rewound at depth $\ell+1$. Furthermore, the simulator obtains m partial views from depth $\ell+1$. We claim that in this case, the simulator should have obtained a view stored in $\bar{\mathcal{H}}$ so that the current view h' along with $\bar{\mathcal{H}}$ would yield a witness for session i through the special-soundness extractor. This is because, if any of the m views returned from depth $\ell+1$ does not contain a second challenge-response pair for slot s , then it must end in the closing of the last slot of a session $j \neq i$ for which a witness has not yet been extracted. Furthermore, session j must have started after h , since otherwise the simulator returns the view to a higher recursive call.

Now, we claim that, no two views among the m views returned can end on the last slot of the same session j . It follows from our inductive hypothesis that session j could not have completed $r - 1$ slots before opening of s ; in this case, it was assumed that a witness was extracted. Therefore, the view returned contains the closing of the $r - 1^{th}$ slot. This in turn implies that the view also contains the challenge-response pair of the last slot; this is stored in $\bar{\mathcal{H}}$. This means that, if at any instant afterward in the recursive simulation from s if session j completes r slots, the current view contains a new challenge-response pair for the last slot of session j that is different from the one stored in $\bar{\mathcal{H}}$; In this case, the simulator would have applied the special-soundness extractor on the current view with the stored view in $\bar{\mathcal{H}}$ to extract a witness for session j and continued simulation. Therefore, no two views returned in the recursive simulation from s can end in the same session.

Finally, observe that there are at most $m - 1$ sessions other than session i ; so, if m views are returned it must be the case that one view contains a second challenge-response pair for slot s and a witness extracted for session i .

To conclude the proof, we observe that if the simulator fails on session i at depth ℓ simulating from h , then it must be the case that session i starts after h , since otherwise, the simulator returns the view to a higher-recursive call where the session started. However, if session i starts after h and completes $r - 1$ slots, as proved above in our preliminary step, it must have extracted a witness for session i and thus cannot fail. \square

INDISTINGUISHABILITY OF THE SIMULATION

Proposition 3. *The following ensembles are identical*

- $\{\text{view}_{V^*}(\langle P(w), V^*(z) \rangle(x))\}_{x \in L, w \in R_L(x), z \in \{0,1\}^*}$
- $\{S^{V^*}(x, z)\}_{x \in L, w \in R_L(x), z \in \{0,1\}^*}$

Proof: Consider the following hybrid simulator \tilde{S}^{V^*} that receives the real witness w to the statement x . \tilde{S}^{V^*} on input x, w , and z proceeds just like S^{V^*} in order to generate the prover messages in Stage 1, but proceeds as the honest prover using the witness w in order to generate messages in Stage 2 (instead of using the “fake” witness as S^{V^*} would have). Using the same proof as in Claim 18, we can

show that $\tilde{S}^{V^*}(x, (w, z))$ never aborts outputting \perp . Furthermore, as the prover messages in Stage 1 are chosen uniformly and \tilde{S}^{V^*} behaves like an honest prover in Stage 2. Therefore, we obtain the following claim.

Claim 19. *The following ensembles are identical*

- $\{\text{view}_{V^*}(\langle P(w), V^*(z) \rangle(x))\}_{x \in L, w \in R_L(x), z \in \{0,1\}^*}$
- $\{\tilde{S}^{V^*}(x, (w, z))\}_{x \in L, w \in R_L(x), z \in \{0,1\}^*}$

Next we compare \tilde{S}^{V^*} with S^{V^*} . The proof of the proposition follows using the standard hybrid argument by combining Claim 19 with the following claim.

Claim 20. *The following ensembles are identical*

- $\{\tilde{S}^{V^*}(x, (w, z))\}_{x \in L, w \in R_L(x), z \in \{0,1\}^*}$
- $\{S^{V^*}(x, z)\}_{x \in L, w \in R_L(x), z \in \{0,1\}^*}$

Proof: The proof of this claim essentially follows from the perfect- \mathcal{WI} property of Stage 2 of the protocols, since the only difference between the simulators \tilde{S}^{V^*} and S^{V^*} is the choice of witness used. For completeness, we provide a proof below.

To prove the claim we will rely on the fact that the running time of the simulator is bounded. This holds since S stops executing SOLVE whenever it performs more than 2^n queries and continues the simulation in a straight-line fashion, extracting “fake” witnesses using brute-force search. Assume, for contradiction, that the claim is false, i.e. there exists a deterministic verifier V^* (we assume w.l.o.g that V^* is deterministic, as its random-tape can be fixed) such that the ensembles are not identical.

We consider several hybrid simulators, S_i for $i = 0$ to N , where N is an upper-bound on the running time of the simulator. S_i receives the real witness w to the statement x and behaves exactly like S , with the exception that Stage 2 messages in the first i proofs are generated using the honest prover strategy (and the witness w). By construction, $S_0 = \tilde{S}$ and $S_N = S$. Since, by assumption, the outputs of S_0 and S_N are not identically distributed, there must exist some j such that the output of S_j and S_{j+1} are different. Furthermore, since S_j proceeds exactly as S_{j+1} in the first j sessions, and also the same in Stage 1 of the $j + 1$ 'th session, there

exists a partial view v —which defines an instance $x' \in L \vee L'$ for Stage 2 of the $j+1$ 'th session—such that outputs of S_j and S_{j+1} are not identical conditioned on the event that S_j and S_{j+1} feed V^* the view v . Since the only difference between the view of V^* in S_j and S_{j+1} is the choice of the witness used for the statement x' used in Stage 2 of the $j+1$ 'th session, we contradict the perfect- \mathcal{WI} property of Stage 2. \square \square

RUNNING-TIME OF S

We consider the hybrid simulator \tilde{S}^{V^*} constructed in proof of Proposition 3. It follows by the same proof as in Claim 20 that the running time distributions of \tilde{S} and S are identical. Therefore, it suffices to analyze the expected running time of \tilde{S} .

Proposition 4. *For all $x \in L, z \in \{0, 1\}^*$, and all V^* such that $V^*(x, z)$ opens up at most m sessions, $E[\text{TIME}_{\tilde{S}^{V^*}(x, z)}] \leq \text{poly}(m^d r^d)$*

Proof: Recall that in the simulation by $\tilde{S}^{V^*}(x, z)$, if at any point the simulator queries more than 2^n queries to V^* , it instead continues in a straight-line simulation using a brute-force search. By linearity of expectation, the expected running time of S is

$$\text{poly}(E[\# \text{ queries made to } V^*]) + E[\text{time spent in straight-line simulation}]$$

In Claim 21 below, we show that expected time spent in straight-line simulation is negligible. In Claim 22 we show that the expected number of queries made to V^* is at most $m^{2(d+1-\ell)}(2r)^{d+1-\ell}$. The proof of the proposition follows.

Claim 21. *The expected time spent by \tilde{S}^{V^*} in straight-line simulation is less than 1.*

Proof: The straight-line simulation takes at most $p(2^n)$ steps for some function $p \in \mathcal{PQT}$ since it takes $O(2^n)$ steps to extract a “fake” witness. Recall that, the simulator runs the brute-force search only if it picks the same challenge (β) twice. Since, the simulator is cut-off after 2^n steps, it could have picked at most 2^n challenges. Therefore, by the union bound, the probability that it obtains the same challenge twice is at most $\frac{2^n}{2^{n^2}}$. Thus, the expected time spent by S^{V^*} in straight-line simulation is at most $\frac{2^n}{2^{n^2}}p(2^n) < 1$. \square

Claim 22. For all partial view h , depth ℓ , such that $S^{V^*}(1^n, \ell, h, \bar{\mathcal{H}})$ never outputs fail, $E[\# \text{ queries by } S^{V^*}(1^n, \ell, h, \bar{\mathcal{H}})] \leq (2m^2r)^{\mathsf{d}-\ell+1}$

Proof: We prove the claim by reverse induction on ℓ . To simplify notation let $\alpha(\ell) = (2m^2r)^{\mathsf{d}-\ell+1}$. When $\ell = \mathsf{d}$ the claim follows since SOLVE does not perform any recursive calls and the number of queries made by SOLVE can be at most the total number of messages, which is mr .

Assume the claim is true for $\ell = \ell' + 1$. We show that it holds also for $\ell = \ell'$. Fix a particular history h and views $\bar{\mathcal{H}}$ such that $S^{V^*}(1^n, \ell', h, \bar{\mathcal{H}})$ never outputs fail. We show that

$$E[\# \text{ queries by } S^{V^*}(1^n, \ell', h, \bar{\mathcal{H}})] \leq (2m^2r)^{\mathsf{d}-\ell'+1} = \alpha(\ell')$$

Towards this goal we introduce some additional notation. Given a view h' extending the view h ,

- Let $q^\ell(h'; \hat{s})$ denote the probability that the view h' occurs in the “main-line” session of $S^{V^*}(1^n, \ell, h)$ and that slot \hat{s} opens immediately after h' .
- Let $\Gamma_{\hat{s}}$ denote the set of views such that $q^\ell(h'; \hat{s}) > 0$.

We bound the number of queries made by $S^{V^*}(1^n, \ell, h)$ as the sum of the queries it makes on depth ℓ , and the queries made by recursive calls. The number of queries made by the simulator on depth ℓ is at most the total number of messages in a session, i.e. mr . The number of queries made on recursive calls is computed by summing the queries made by recursive calls on over every slot \hat{s} and taking expectation over every view h' (such that $q^\ell(h'; \hat{s}) > 0$).

More precisely,

$$E[\# \text{ queries by } S^{V^*}(1^n, \ell, h)] \leq mr + \sum_{\hat{s}} \sum_{h' \in \Gamma_{\hat{s}}} q^\ell(h'; \hat{s}) E(h'; \hat{s})$$

where $E(h'; \hat{s})$ denotes the expected number of queries made by the simulator from the view h' on \hat{s} . There are two steps involved in computing $E(h'; \hat{s})$. The first step involves finding the expected number of times SOLVE is run on a slot and the second step, using the induction hypothesis, computing a bound for $E(h'; \hat{s})$.

Step 1: Recall that the simulator at depth ℓ is restarted either if the current view h' is not ℓ -good or if the simulation reaches the Stage 2 of a session that starts

at a prefix of h . We show below that, in expectation the former causes S^{V^*} to restart at depth $\ell+1$ at most $O(1)$ times. Then we argue that the latter can occur without S^{V^*} returning to a higher recursive call at most m times. Therefore, using linearity of expectation, we get that the expected number of times S^{V^*} is restarted before it obtains a second challenge-response pair for the slot \hat{s} is at most $O(m)$.

First, we consider the probabilities of the following events (all conditioned on S^{V^*} never outputs \perp). Given a view h' from where slot \hat{s} opens, let p^ℓ denote the probability that S^{V^*} rewinds slot \hat{s} from h' , i.e. p^ℓ is the probability that in the simulation from h' at depth ℓ , V^* completes \hat{s} with an accepting proof while opening fewer than $\frac{m}{(r-1)^d - \ell + 1}$ new sessions within the slot \hat{s} . Let y^ℓ denote the probability that when executing the simulator at depth ℓ , S^{V^*} either cuts off the simulation or returns the current view h' . We clearly have that $p^\ell \leq 1 - y^\ell$ (note that equality does not necessarily hold since it might return h' to a higher recursive call). Recall that S^{V^*} generates random Stage 1 messages, and uses the same (real) witness to generate Stage 2 messages, independent of the depth of the recursion. Conditioned on S^{V^*} never failing or returning to a higher recursive call, we can conclude that the view of V^* in the “main-line” simulation by S^{V^*} on depth ℓ is identically distributed to its view on depth $\ell+1$. Therefore, the probability it aborts or opens too many new sessions at both the depths are identical. However, it could stop and return to a higher recursive call more often in the simulation at depth $\ell+1$ than at depth ℓ . Thus, we have that $y^\ell \leq y^{\ell+1}$.

Therefore, the expected number of times S^{V^*} recursively executes \hat{s} at depth $\ell+1$, before it either obtains a second challenge-response pair for the slot or returns to a higher recursive call is at most $\frac{1}{1-y^{\ell+1}} \leq \frac{1}{1-y^\ell} \leq \frac{1}{p^\ell}$. Since, S^{V^*} rewinds \hat{s} from \hat{h} only with probability p^ℓ , the expected number of restarts at depth $\ell+1$ from h' before which the rewinding is not cut-off is at most $p^\ell \frac{1}{p^\ell} = 1$. Since it is run until m views are obtained, we have that by linearity of expectation, the expected number of times the simulation at depth $\ell+1$ is carried out for each slot is bounded by m .

Step 2: From the induction hypothesis, we know that the expected number of queries made by S^{V^*} at depth $\ell'+1$ is at most $\alpha(\ell'+1)$. Therefore, if S^{V^*} is run u times on a slot, the expected total number of queries made by SOLVE is bounded

by $u\alpha(\ell' + 1)$. We conclude that

$$\begin{aligned}
 E(h'; \hat{s}) &\leq \sum_{u \in \mathbb{N}} \Pr[u \text{ recursive calls are made by } S^{V^*} \text{ from } h'] u\alpha(\ell' + 1) \\
 &= \alpha(\ell' + 1) \sum_{u \in \mathbb{N}} u \cdot \Pr[u \text{ recursive calls are made by } S^{V^*} \text{ from } h'] \\
 &\leq m\alpha(\ell' + 1)
 \end{aligned}$$

Therefore, $E[\# \text{ queries by } S^{V^*}(1^n, \ell', h)] \leq$

$$\begin{aligned}
 mr + \sum_{\hat{s}} \sum_{h' \in \Gamma_{\hat{s}}} q^{\ell'}(h'; \hat{s}) E(h'; \hat{s}) &\leq mr + \sum_{\hat{s}} m\alpha(\ell' + 1) \sum_{h' \in \Gamma_{\hat{s}}} q^{\ell'}(h'; \hat{s}) \\
 &\leq mr + \sum_{\hat{s}} m\alpha(\ell' + 1) \leq mr + (mr)m\alpha(\ell' + 1) \leq \alpha(\ell')
 \end{aligned}$$

This completes the induction step and concludes the proof of Claim 4. \square \square

8.9 Concurrent Computational \mathcal{ZK} Proof for \mathbf{NP}

In the previous section, we constructed perfect concurrent \mathcal{ZK} proofs assuming the existence of claw-free permutations. In this section, we consider concurrent computational \mathcal{ZK} proofs.

Theorem 23 (restated). *Assume the existence of one-way functions that are secure w.r.t $\omega(\mathcal{PQT})$ and collision-resistant hash function that are secure w.r.t $\omega(\mathcal{PQT})$. Then, every language language in \mathbf{NP} has a constant-round concurrent computational black-box \mathcal{ZK} proof w.r.t \mathcal{PQT} .*

Theorem 18 relies on a slight variant of the \mathcal{ZK} proof of [56] (and is an instantiation of the protocol of [70]). This protocol is described in Figure 8.3. We assume the existence of honest-verifier \mathcal{ZK} proofs that are secure w.r.t $\omega(\mathcal{PQT})$. Such proofs exists if one-way functions that are secure w.r.t $\omega(\mathcal{PQT})$ exists. Furthermore, we require constant round statistically hiding commitments that are computationally binding w.r.t $\omega(\mathcal{PQT})$ adversaries. Such commitment schemes can be constructed from collision resistant hash functions that are secure w.r.t $\omega(\mathcal{PQT})$ [23, 42]. Given these assumptions with some subtle changes, our proof from the previous section works for this protocol as well.

Let $h(n) \in \omega(n^{\text{poly}(\log n)})$ be such that there exists OWFs and CRHs secure w.r.t $(h(n))^2$.

DESCRIPTION OF THE SIMULATOR: We modify the simulator described in Section 8.7 to work for this protocol. More precisely, we change the procedure SOLVE as follows. To generate the prover message for Stage 1 of any execution, it picks challenges uniformly at random as before, but to simulate the Stage 2 of an execution, on input the “fake witness” $w = \bar{r}$, proceeds as follows⁵:

1. S generates a “random-looking” execution (m_1, \bar{r}', m_2) of the “parallelized” GMW protocol, where the verifier query $\bar{r}' = \bar{r}$. (This property of the GMW protocol is sometimes called special honest-verifier \mathcal{ZK} .)⁶
2. S feeds m_1 to V^* .
3. If V^* decommits to \bar{r} , S feeds m_2 to V^* .
4. If V^* succeeds in decommit to a different value than \bar{r} , S outputs fail_{bin} and halts.

In the event that the simulation runs for more than $h(n)$ steps, S halts with output $\text{fail}_{\text{time-out}}$.

ANALYSIS OF THE SIMULATOR

We analyze the correctness and running time of the simulator S . In the rest of the proof, we write negligible for negligible w.r.t \mathcal{PQT} .

First, we prove the following claim regarding S .

Claim 23. *The probability that S outputs fail_{bin} is negligible w.r.t \mathcal{PQT} .*

Proof: Recall that S outputs fail_{bin} whenever the verifier decommits to a value that is different from the one extracted from the Stage 1 proofs. Assume for contradiction, S^{V^*} outputs fail_{bin} with probability $p(n) = \frac{1}{n^{\log^d n}}$ for infinitely many n . We show that S can be used to violate the computational-binding property of COM and arrive at a contradiction.

⁵This is similar to the simulator constructed in [32]

⁶Note that this is possible since it is easy to commit to a coloring such that the two vertices on a *particular* (predetermined) edge have different colors.

Protocol CompZKProof

Common Input: an instance x of a language L with witness relation R_L .

Auxiliary Input for Prover: a witness w , such that $(x, w) \in R_L(x)$.

Stage 1:

V uniformly chooses $\bar{r} = r_1, r_2, \dots, r_n \in \{0, 1\}^n$, $s \in \{0, 1\}^{poly(n)}$.

$V \rightarrow P$: $c = \text{COM}(\bar{r}; s)$, where COM is a statistically hiding commitment, which has the property that the committer must communicate at least m bits in order to commit to m strings.

$V \rightarrow P$: r first messages $\alpha_1, \dots, \alpha_r$ for \mathcal{WI} special-sound proofs of the statement. (called the **start** message)

there exists values \bar{r}', s' s.t $c = \text{COM}(\bar{r}'; s')$

The proof of knowledge is with respect to the witness relation $R'_L(c) = \{(v, s) | c = \text{COM}(v; s)\}$.

For $j = 1$ to r do

$P \rightarrow V$: Second message $\beta_j \leftarrow \{0, 1\}^{n^2}$ for j^{th} \mathcal{WI} special-sound proof. (called the **opening** of slot j)

$V \rightarrow P$: Third message γ_j for j^{th} \mathcal{WI} special-sound proof. (called the **closing** of slot j)

Stage 2:

$P \leftrightarrow V$: P and V engage in n parallel executions of the GMW's (3-round) Graph 3-Coloring protocol, where V uses the strings r_1, \dots, r_n as its challenges:

1. $P \rightarrow V$: n (random) first messages of the *GMW* proof system for the statement x .
2. $V \leftarrow P$: V decommits to $\bar{r} = r_1, \dots, r_n$.
3. $P \rightarrow V$: For $i = 1..n$, P computes the answer (i.e., the 3rd message of the GMW proof system) to the challenge r_i and sends all the answers to V .

Figure 8.3: Computational \mathcal{ZK} Proof for \mathbf{NP}

Consider an efficient cheating committer C^* that runs S^{V^*} internally. C^* picks a random COM committed by V^* in the simulation by S , and forwards the COM to an outside honest receiver. This is possible, despite the fact that S (and therefore C^*) internally rewinds A , because COM is a 2-round (non-interactive) commitment scheme. Then C^* also forwards the correct COM with probability $p(n)$, of which it obtains two valid openings: one extracted by S at the end of Stage 1, and one given by V^* in Stage 2. Since there are at most $h(n)$ proofs simulated by S , C^* picks the right one with probability at least $\frac{1}{h(n)}$. Since S^{V^*} outputs fail_{bin} with probability $p(n)$, the probability that C^* succeeds is at least $p(n)\frac{1}{h(n)}$. This contradicts the computational binding property of COM (w.r.t $(h(n))^2$). \square

Next, we consider two hybrid simulators \tilde{S} and \hat{S} that receives the real witness w to the statement x . \tilde{S} on input x, w , proceeds just like S in order to generate the prover messages in Stage 1 in each execution, but proceeds as the honest prover using the witness w instead of using the “fake” witness to generate messages in Stage 2. Additionally, \tilde{S} aborts outputting \perp just as S does.⁷ However, \tilde{S} does not fail if the verifier decommits to a different value other than the one extracted (i.e. never outputs fail_{bin}) or if its running time exceeds $h(n)$ (i.e. never outputs $\text{fail}_{\text{time-out}}$). The hybrid simulator \hat{S} proceeds identical to \tilde{S} , with the only exception being, it halts outputting $\text{fail}_{\text{time-out}}$ whenever it runs more than $h(n)$ steps (where $n = |x|$). It follows identically to Claim 18 and Proposition 4 that \tilde{S} never aborts outputting \perp and its expected running time is bounded by $n^{\log^c n}$ for some constant c . It also follows from Claim 19 that the output of \tilde{S} and the view of verifier in a real-interaction are identical. The running-time and correctness of S follows from the next two claims.

Claim 24. *For any $x \in L, w \in R_L(x), z \in \{0, 1\}^*$, the expected running time of \hat{S} on input $(x, (w, z))$ is bounded by $n^{\log^c n}$ for some constant $c \in \mathbb{N}$, where $n = |x|$ and the following two ensembles are statistically close w.r.t \mathcal{PQT} .*

- $\{\tilde{S}(x, (w, z))\}_{x \in L, w \in R_L(x), z \in \{0, 1\}^*}$
- $\{\hat{S}(x, (w, z))\}_{x \in L, w \in R_L(x), z \in \{0, 1\}^*}$

Proof: Since \hat{S} proceeds identically to \tilde{S} with the exception that it outputs $\text{fail}_{\text{time-out}}$ whenever it runs more than $h(n)$ steps we have that the expected running time of \hat{S} is bounded by the expected running time of \tilde{S} which is $n^{\log^c n}$. It

⁷This happens when it has not extracted a “fake” witness and reaches Stage 2 of that session.

then follows using Markov-inequality that the probability that \widehat{S} outputs $\text{fail}_{\text{time-out}}$ is at most $\frac{n^{\log^c n}}{h(n)}$. Therefore, the outputs of \widetilde{S} and \widehat{S} are statistically-close w.r.t \mathcal{PQT} . \square

Claim 25. *For any $x \in L, w \in R_L(x), z \in \{0, 1\}^*$, the expected running time of S on input $(x, (w, z))$ is bounded by $n^{2\log^d n}$, where $n = |x|$, d is some constant and the following two ensembles are indistinguishable w.r.t \mathcal{PQT} over $x \in L$.*

- $\{\widehat{S}(x, (w, z))\}_{x \in L, w \in R_L(x), z \in \{0, 1\}^*}$
- $\{S^{V^*}(x, z)\}_{x \in L, w \in R_L(x), z \in \{0, 1\}^*}$

Proof:

First we prove that the expected running time of S is bounded by $n^{2\log^c n}$. In a run by S or \widehat{S} , the maximum number of proofs made as part of the Stage 2 of an execution is bounded by the running time which is $h(n)$.

Suppose, for contradiction, the expected running time of S is bigger than $n^{2\log^c n}$. We will consider the hybrid simulators, S_i for $i = 1$ to $h(n)$ as defined above. Therefore, there must exist a j such that the expected running time of $S_{j(n)}$ and $S_{j(n)+1}$ differ by at least $\frac{n^{2\log^c n} - n^{\log^c n}}{h(n)} \geq \frac{n^{\log^c n}}{h(n)}$.

We construct a verifier V' that violates the special honest-verifier \mathcal{ZK} property of the protocol in Stage 2. More precisely, we will construct V' , such that the ensembles $\text{OUTPUT}_{V'}(\langle P(w), V' \rangle(x))$ and $\text{OUTPUT}_{V'}(\langle P(r), V' \rangle(x))$ are distinguishable, where w is the real witness, i.e. $w \in R_L(x)$ and r is a fake witness.

The verifier V' does the following. It runs a simulator S^* that behaves like $S_{j(n)+1}$, except that it feeds the messages from the outside prover internally, when the simulator needs to generate the $(j(n) + 1)$ proof for the Stage 2 of some execution. Furthermore, V' computes the running time t of the simulator. Finally, V' tosses some random coins and outputs 1 with probability $\frac{t}{h(n)}$. By construction, the simulator S^* behaves like $S_{j(n)}$ when the external prover uses the “fake” witness and like $S_{j(n)+1}$ when it uses the real witness. Furthermore, V outputs 1 with probability $\frac{T}{h(n)}$ where T is the expected running time of the simulation. Since, the difference in the expected running time of $S_{j(n)}$ and $S_{j(n)+1}$ is at least $\frac{n^{\log^c n}}{h(n)}$,

we have that

$$\begin{aligned} & |\Pr[\text{OUTPUT}_{V'}(\langle P(w), V' \rangle(x)) = 1] - \Pr[\text{OUTPUT}_{V'}(\langle P(r), V' \rangle(x)) = 1]| \\ & \quad > \frac{n^{\log^c n}}{h(n)} = \frac{1}{\omega(n^{poly(\log n)})} \end{aligned}$$

which contradicts the special honest-verifier \mathcal{ZK} property of the protocol in Stage 2.

Next, we prove indistinguishability of \widehat{S} and S . Towards this, we consider a different sequence of hybrids to prove the indistinguishability of the output of the simulation. We describe two sequences of hybrid simulators S_i and S_i^+ , $0 \leq i \leq h(n)$ that receive the witnesses of the statements. We order the sequence of proofs in an output of S by the order in which Stage 1 is completed. The hybrid simulator S_i proceeds as follows.

1. Run the simulator S with verifier V^* *in its entirety*. Output fail and fail_{bin} if S outputs fail and fail_{bin} respectively. Otherwise, let \mathcal{V} be the view output by S .
2. Let \mathcal{V}_i be the prefix of \mathcal{V} up until the Stage 1 of the i^{th} proof is completed. Simulate an execution with V^* starting from view \mathcal{V}_i in a *straight-line manner*.
 - Continue the simulation of the first i proofs in the same manner as S , i.e. using the “fake” witness extracted by S . This can be done in a straight-line manner for the first i proofs since the “fake” witnesses extracted are still useful. However, similar to S , if V^* decommits to a string different from the fake witness, S_i outputs fail_{bin} .
 - Continue the simulation of the $i + 1^{st}$ and later proofs by the following the honest prover strategy using the given real witness.
3. Output the view generated.

We also define S_i^+ that proceeds identically to S_i except that in Step 2, it simulates the i^{th} proof using the honest prover strategy using the real witness. We start with a claim bounding the failing probability of S_i .

Subclaim 4. *For all i , the probability that S_i and S_i^+ outputs fail or fail_{bin} is negligible w.r.t \mathcal{PQT} .*

Proof: Recall that S_i and S_i^+ output fail and fail_{bin} if S does and we know from above and 23 this happens with probability negligible w.r.t \mathcal{PQT} . Furthermore, they output fail_{bin} if V^* decommits to a different value in any of the first i proofs. Using the same proof as in Claim 23, we can show that this happens with probability negligible w.r.t \mathcal{PQT} . \square

By Claim 4, the output of S_0 is statistically close to the real view with V^* (they only differ when S_0 aborts, which occurs with negligible probability). The output of S_m , on the other hand, is identical to the output of simulator S . Indistinguishability follows from the next two claims:

Subclaim 5. *For any $i : \mathbb{N} \rightarrow \mathbb{N}$, the ensembles*

- $\{S_{i(|x|)}(x, w, z)\}_{x \in L, w \in R_L(x), z \in \{0,1\}^*}$ and
- $\{S_{i(|x|)}^+(x, w, z)\}_{x \in L, w \in R_L(x), z \in \{0,1\}^*}$

are computationally indistinguishable w.r.t \mathcal{PQT} over $x \in L$.

Proof: S_i and S_i^+ differs only in how the i^{th} execution is simulated (i.e. real or fake witness), which is done in a straight line fashion by both hybrids. Furthermore, conditioned on not outputting fail_{bin} , the verifier always reveals the correct decommitment. Therefore, they are computationally indistinguishable by the witness indistinguishability property that follows from the special honest-verifier \mathcal{ZK} property of the Stage 2 proof. \square

Subclaim 6. *For any $i : \mathbb{N} \rightarrow \mathbb{N}$, the ensembles*

- $\{S_{i(|x|)}^+(x, z)\}_{x \in L, w \in R_L(x), z \in \{0,1\}^*}$ and
- $\{S_{i(|x|)-1}(x, z)\}_{x \in L, w \in R_L(x), z \in \{0,1\}^*}$

are statistically close w.r.t \mathcal{PQT} over $x \in L$.

Proof: Ignoring the fact that S_i^+ and S_{i-1} may abort, their outputs are identical. This is because S_i^+ differs from S_{i-1} only in that when generating the output view, from the end of the Stage 1 of the $i - 1^{st}$ proof until the end of the Stage 1 of the i^{th} proof, S_i^+ employs additional *rewinds*. However, these rewinds do not extract

any new “fake witnesses” for use in the output view, and do not skew the output distribution because the randomness used in the main thread is independent of the rewinding. Since both machines abort at most with probability negligible w.r.t \mathcal{PQT} by Claim 4, their outputs are statistically close w.r.t \mathcal{PQT} . \square \square

8.10 Concurrent \mathcal{ZK} Arguments from OWFs

Using essentially the same protocol we get Theorem 19.

Theorem 24 (restated). *Assume the existence of one-way function that are secure w.r.t \mathcal{PQT} . Then, every language in \mathbf{NP} has an $O(1)$ -round concurrent computational black-box \mathcal{ZK} arguments w.r.t \mathcal{PQT} .*

Proof: The protocol is obtained by using a computational \mathcal{WI} protocol w.r.t \mathcal{PQT} instead of the perfect \mathcal{WI} protocol in Stage 2 described in Section 8.5. Such proofs can be constructed based on the existence of OWF secure for \mathcal{PQT} . We consider the same simulator as in the previous section with the exception of how the prover messages are generated, for which we rely on the original simulator from Section 8.7. Then, following the same proof from the last section the theorem holds. \square

8.11 Concurrent Perfect \mathcal{ZK} Proofs for languages in \mathbf{NP}

We provide *unconditional* constructions of *perfect \mathcal{ZK} proofs* for certain specific languages. Our constructions are essentially identical to the protocols in [56]. We here present only a construction of **GraphNonIso**, but as in [56], the same paradigm works also for QNR. The construction proceeds in the following steps:

1. As in [56], we first recast (a variant, due to Benaloh [10], of) Goldreich, Micali and Wigderson’s protocol [35] for Graph Non-Isomorphism as an instance of the Feige-Shamir protocol.
2. We then essentially rely on the same construction paradigm as in our previous constructions; namely, we repeat the special-sound proof of knowledge in Stage 1 r times.

8.12 Unconditional \mathcal{WI} Proof of Knowledge for Specific Languages

We provide an example of a *perfect- \mathcal{WI} proof of knowledge* for a GraphNonIso . As mentioned above, this protocol will then be used in order to construct a perfect \mathcal{ZK} proof for GraphNonIso .

Consider the language 1of2GraphIso of triplets of graphs G_0, G_1, H , such that H isomorphic to either G_0 or G_1 , and the corresponding witness relation $R_{\text{1of2GraphIso}}$ which describes the two isomorphism. The protocol (which is a variant of a protocol implicit in [35] and the protocol of Benaloh [10]) is depicted in Figure 8.4 is a 3-round special-sound \mathcal{WI} proof for $R_{\text{1of2GraphIso}}$.

Protocol 1of2GraphIsoProof

Common Input: an instance G_0, G_1, H of the language 1of2GraphIso .

Auxiliary Input for Prover: a witness w , such that $((G_0, G_1, H), w) \in R_{\text{1of2GraphIso}}(x)$.

P uniformly selects a bit i , and lets C_i be a random isomorphic copy of G_0 and C_{1-i} be a random isomorphic copy of G_1 .

$P \rightarrow V: C_0, C_1$.

$V \rightarrow P$: a random bit b .

$P \rightarrow V$:

1. If $b = 0$, P sends the permutation from C_i, C_{1-i} to G_0, G_1 .
2. If $b = 1$, P sends the permutation from H to of C_i, C_{1-i} .

V checks the validity of the permutations received.

Figure 8.4: \mathcal{WI} Proof of Knowledge for 1OF2GRAPHISO

Soundness and Completeness follow directly using the same proof as in [35]. Perfect- \mathcal{WI} follows from the fact that protocol 1of2GraphIsoProof is honest-verifier perfect zero-knowledge (see [35]). By using parallel repetition and an appropriate representation of the graphs, we thus obtains 3-round perfect- \mathcal{WI} special-sound proof system (P, V) for 1of2GraphIso with witness relation $R_{\text{1of2GraphIso}}$. Furthermore, the verifier query in (P, V) for a statement $x \in \{0, 1\}^n$ is of length n^2 .

8.13 Concurrent Perfect \mathcal{ZK} Proof for Graph Non-Isomorphism

Let GraphNonIso denote the language of non-isomorphic graphs.

Theorem 25 (restated). *There exists a constant-round concurrent perfect \mathcal{ZK} proof w.r.t \mathcal{PQT} for GraphNonIso .*

Proof: Let $1\text{of}2\text{GraphIso}$ and $R_{1\text{of}2\text{GraphIso}}$ be defined as in Section 8.12. Consider the protocol depicted in Figure 8.5 for proving that $x \in \text{GraphNonIso}$. Soundness and Completeness of the protocol follows as in [35]. Perfect concurrent simulation follows from Proposition 3 and is obtained by using the simulator constructed in Section 8.7. Furthermore, the expected running time of the simulator is $n^{poly(\log n)}$ if r is set to some constant (≥ 3). \square

Protocol PerfectZKGraphNonIso
Common Input: an instance G_0, G_1 of the language GraphNonIso .
Stage 1:
<p>V uniformly chooses a bit i and let H be a random isomorphic copy of G_i.</p> <p>$V \rightarrow P: H$.</p> <p>$V \rightarrow P: r$ first messages $\alpha_1, \dots, \alpha_r$ for \mathcal{WI} Proof of Knowledge of the statement (G_0, G_1, H) w.r.t \mathbf{NP}-relation $R_{1\text{of}2\text{GraphIso}}$.</p> <p>For $j = 1$ to r do</p> <ul style="list-style-type: none"> $P \rightarrow V$: Second message $\beta_j \leftarrow \{0, 1\}^{n^2}$ for j^{th} \mathcal{WI} Proof. $V \rightarrow P$: Third message γ_j for j^{th} \mathcal{WI} Proof.
Stage 2:
<p>$P \rightarrow V$: The bit i' such that H is isomorphic to $G_{i'}$.</p> <p>V accepts if $i' = i$.</p>

Figure 8.5: Perfect Concurrent \mathcal{ZK} Proof for GraphNonIso

Theorem 26 (restated). *For every $\varepsilon > 0$, there exists a $O(n^\varepsilon)$ -round concurrent perfect \mathcal{ZK} proof for GraphNonIso .*

Proof: This is directly obtained by relying on an n^ε rounds version of the previous protocol (instead of constant number of rounds). \square

BIBLIOGRAPHY

- [1] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, pages 106–115, 2001.
- [2] Boaz Barak, Ran Canetti, Jesper Buus Nielsen, and Rafael Pass. Universally composable protocols with relaxed set-up assumptions. In *FOCS*, pages 186–195, 2004.
- [3] Boaz Barak and Oded Goldreich. Universal arguments and their applications. In *IEEE Conference on Computational Complexity*, pages 194–203, 2002.
- [4] Boaz Barak and Yehuda Lindell. Strict polynomial-time in simulation and extraction. In *STOC '02*, pages 484–493, 2002.
- [5] Boaz Barak and Rafael Pass. On the possibility of one-message weak zero-knowledge. In *TCC*, pages 121–132, 2004.
- [6] Boaz Barak and Amit Sahai. How to play almost any mental game over the net - concurrent composition via super-polynomial simulation. In *FOCS*, pages 543–552, 2005.
- [7] Donald Beaver. Foundations of secure interactive computing. In *CRYPTO*, pages 377–391, 1991.
- [8] Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols (extended abstract). In *STOC*, pages 503–513, 1990.
- [9] Mihir Bellare, Markus Jakobsson, and Moti Yung. Round-optimal zero-knowledge arguments based on any one-way function. In *EUROCRYPT*, pages 280–305, 1997.
- [10] Josh Cohen Benaloh. Cryptographic capsules: A disjunctive primitive for interactive protocols. In *CRYPTO*, pages 213–222, 1986.
- [11] M. Blum. How to prove a theorem so that no one else can claim it. In *International Congress of Mathematicians*, 1986.
- [12] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145, 2001.

- [13] Ran Canetti. Obtaining universally composable security: Towards the bare bones of trust. In *ASIACRYPT*, pages 88–112, 2007.
- [14] Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. Universally composable security with global setup. In *TCC*, pages 61–85, 2007.
- [15] Ran Canetti and Marc Fischlin. Universally composable commitments. In *CRYPTO '01*, pages 19–40, 2001.
- [16] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *STOC '00*, pages 235–244, 2000.
- [17] Ran Canetti, Joe Kilian, Erez Petrank, and Alon Rosen. Black-box concurrent zero-knowledge requires $\tilde{\omega}(\log n)$ rounds. In *STOC '01*, pages 570–579, 2001.
- [18] Ran Canetti, Eyal Kushilevitz, and Yehuda Lindell. On the limitations of universally composable two-party computation without set-up assumptions. In *EUROCRYPT*, pages 68–86, 2003.
- [19] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *STOC*, pages 494–503, 2002.
- [20] Ran Canetti, Rafael Pass, and Abhi Shelat. Cryptography from sunspots: How to use an imperfect reference string. In *FOCS*, pages 249–259, 2007.
- [21] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *CRYPTO*, pages 174–187, 1994.
- [22] Ivan Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *EUROCRYPT*, pages 418–430, 2000.
- [23] Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. Statistical secrecy and multibit commitments. *IEEE Transactions on Information Theory*, 44(3):1143–1151, 1998.
- [24] Yevgeniy Dodis and Silvio Micali. Parallel reducibility for information-theoretically secure computation. In *CRYPTO*, pages 74–92, 2000.
- [25] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

- [26] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *CRYPTO*, pages 139–147, 1992.
- [27] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. *J. ACM*, 51(6):851–898, 2004.
- [28] Cynthia Dwork, Moni Naor, and Hoeteck Wee. Pebbling and proofs of work. In *CRYPTO*, pages 37–54, 2005.
- [29] Cynthia Dwork and Amit Sahai. Concurrent zero-knowledge: Reducing the need for timing constraints. In *CRYPTO*, pages 442–457, 1998.
- [30] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *STOC '90*, pages 416–426, 1990.
- [31] Oded Goldreich. *Foundations of Cryptography — Basic Tools*. Cambridge University Press, 2001.
- [32] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, 1996.
- [33] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.
- [34] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC '87*, pages 218–229, 1987.
- [35] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.
- [36] Shafi Goldwasser and Leonid A. Levin. Fair computation of general functions in presence of immoral majority. In *CRYPTO*, pages 77–93, 1990.
- [37] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.
- [38] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

- [39] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comput.*, 17(2):281–308, 1988.
- [40] Vipul Goyal. Constant round non-malleable protocols using one way functions. In *STOC*, 2011.
- [41] Jens Groth and Rafail Ostrovsky. Cryptography in the multi-string model. In *CRYPTO*, pages 323–341, 2007.
- [42] Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In *CRYPTO*, pages 201–215, 1996.
- [43] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.
- [44] Yael Tauman Kalai, Yehuda Lindell, and Manoj Prabhakaran. Concurrent general composition of secure protocols in the timing model. In *STOC ’05*, pages 644–653, 2005.
- [45] Jonathan Katz. Universally composable multi-party computation using tamper-proof hardware. In *EUROCRYPT*, pages 115–128, 2007.
- [46] Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Round efficiency of multi-party computation with a dishonest majority. In *EUROCRYPT*, pages 578–595, 2003.
- [47] Joe Kilian and Erez Petrank. Concurrent and resettable zero-knowledge in poly-loalgoritm rounds. In *STOC ’01*, pages 560–569, 2001.
- [48] Joe Kilian, Erez Petrank, and Charles Rackoff. Lower bounds for zero knowledge on the internet. In *FOCS ’98*, pages 484–492, 1998.
- [49] Huijia Lin and Rafael Pass. Non-malleability amplification. In *STOC*, pages 189–198, 2009.
- [50] Huijia Lin and Rafael Pass. Constant-round non-malleable commitments from any one-way function. In *STOC*, 2011.
- [51] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. Con-

- current non-malleable commitments from any one-way function. In *TCC*, pages 571–588, 2008.
- [52] Yehuda Lindell. Bounded-concurrent secure two-party computation without setup assumptions. In *STOC '03*, pages 683–692, 2003.
 - [53] Yehuda Lindell. General composition and universal composability in secure multi-party computation. In *FOCS*, pages 394–403, 2003.
 - [54] Yehuda Lindell. Lower bounds for concurrent self composition. In *TCC '04*, pages 203–222, 2004.
 - [55] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000.
 - [56] Silvio Micali and Rafael Pass. Local zero knowledge. In *STOC '06*, pages 306–315, 2006.
 - [57] Silvio Micali, Rafael Pass, and Alon Rosen. Input-indistinguishable computation. In *FOCS*, pages 367–378, 2006.
 - [58] Silvio Micali and Phillip Rogaway. Secure computation (abstract). In *CRYPTO*, pages 392–404, 1991.
 - [59] Moni Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.
 - [60] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Perfect zero-knowledge arguments for p using any one-way permutation. *J. Cryptology*, 11(2):87–108, 1998.
 - [61] Rafail Ostrovsky, Giuseppe Persiano, and Ivan Visconti. Concurrent non-malleable witness indistinguishability and its applications. *Electronic Colloquium on Computational Complexity (ECCC)*, 13(095), 2006.
 - [62] Omkant Pandey, Rafael Pass, Amit Sahai, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Precise concurrent zero knowledge. In *EUROCRYPT*, pages 397–414, 2008.
 - [63] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In *EUROCRYPT*, pages 160–176, 2003.

- [64] Rafael Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. In *STOC*, pages 232–241, 2004.
- [65] Rafael Pass. A precise computational approach to knowledge. PhD Thesis, 2004.
- [66] Rafael Pass and Alon Rosen. Bounded-concurrent secure two-party computation in a constant number of rounds. In *FOCS*, pages 404–, 2003.
- [67] Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In *STOC '05*, pages 533–542, 2005.
- [68] Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Unconditional characterizations of concurrent zero knowledge. Manuscript, 2008.
- [69] Birgit Pfitzmann and Michael Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *IEEE Symposium on Security and Privacy*, pages 184–, 2001.
- [70] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *FOCS '02*, pages 366–375, 2002.
- [71] Manoj Prabhakaran and Amit Sahai. New notions of security: achieving universal composability without trusted setup. In *STOC*, pages 242–251, 2004.
- [72] M. O. Rabin. How to exchange secrets by oblivious transfer. *IEEE Transactions on Reliability*, 1981.
- [73] Ransom Richardson and Joe Kilian. On the concurrent composition of zero-knowledge proofs. In *Eurocrypt '99*, pages 415–432, 1999.
- [74] Alon Rosen. A note on the round-complexity of concurrent zero-knowledge. In *CRYPTO*, pages 451–468, 2000.
- [75] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553, 1999.
- [76] Luca Trevisan and Salil P. Vadhan. Extracting randomness from samplable distributions. In *FOCS*, pages 32–42, 2000.

- [77] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS '82*, pages 80–91, 1982.