— A merged version of this work and the work of [BGW12] appears in the proceedings of the *Theory of Cryptography Conference* - TCC 2013 —

On the (In)security of the Fiat-Shamir Paradigm, Revisited

Dana Dachman-Soled
Microsoft Research

Abhishek Jain
UCLA

Yael Tauman Kalai
Microsoft Research

Adriana López-Alt
New York University

Abstract

The Fiat-Shamir paradigm [CRYPTO'86] is a heuristic for converting 3-round identification schemes into signature schemes, and more generally, for collapsing rounds in public-coin interactive protocols. This heuristic is very popular both in theory and in practice, and many researchers have studied its security (and insecurity).

In this work, we continue this study. As our main result, we show that for many well studied interactive *proofs* (and arguments) the soundness of the Fiat-Shamir heuristic cannot be proven via a black-box reduction to *any falsifiable assumption*. Previously, the insecurity of this paradigm was exemplified only when applied to interactive *arguments* (as opposed to proofs).

Using similar techniques, we also show a black-box impossibility result for Micali's CS-proofs [FOCS'94]. Namely, we prove that there exist PCPs such that for "sufficiently hard" NP languages, Micali's CS-proof cannot be proven sound via black-box reduction to any falsifiable assumption.

These results are obtained by extending the impossibility of two-message zero knowledge protocols due to Goldreich and Oren [J. Cryptology'94].

1 Introduction

In 1986, Fiat and Shamir [FS86] proposed a general method for eliminating interaction from any public coin protocol by replacing the verifier with a hash function. Initially, this heuristic was proposed for the sake of transforming three-round public-coin identification (ID) schemes into digital signature schemes, as known ID schemes (in which a sender *interactively* identifies himself to a receiver) were significantly simpler and more efficient than known signature schemes. Indeed, this so called *Fiat-Shamir heuristic*, quickly gained popularity both in theory and in practice, as it yields efficient and easy to implement digital signature schemes.

The Fiat-Shamir heuristic also had important applications outside the regime of ID and signature schemes. For example, it was used by Micali in his construction of CS-proofs [Mic94]. More generally, the importance of the Fiat-Shamir heuristic stems from the fact that the latency of sending messages back and forth has been shown to often be the bottleneck in the running of cryptographic protocols [MNPS04, BDNP08].

The main question is: *Is the Fiat-Shamir heuristic sound*? Indeed, due to the popularity and importance of the Fiat-Shamir heuristic, many researchers have tried to prove or disprove its security. In particular, Pointcheval and Stern [PS96] proved that the Fiat-Shamir heuristic is secure in the so called *random oracle model* (formalized in [BR93]) – when the hash function is modeled by a random oracle. In contrast, Barak [Bar01] gave an example of a constant-round public-coin protocol such that when the Fiat-Shamir heuristic is applied to it, the resulting 2-round protocol is not sound. In a followup work, Goldwasser and Kalai [GK03], gave such an example for a 3-round public-coin ID scheme. However, both these negative results are for protocols that are only *computationally sound*, also known as *arguments*.

The main question that remained open is: *Is the Fiat-Shamir heuristic sound when applied to proofs?* Many researchers have investigated this question. However, despite numerous efforts, there is still no satisfying answer. We refer the reader to Section 1.2 for details on these related works.

1.1 Our Results

In this work, we continue the study of the Fiat-Shamir paradigm and obtain several new results. Below, we state our results very roughly, followed by a more detailed explanation.

1. We show that for many well-studied public-coin interactive *proofs* (and specifically, three-round ID schemes), the soundness of the Fiat-Shamir heuristic cannot be proven via a black-box reduction to *any falsifiable assumption*.

2. Using similar techniques, we show a black-box impossibility result for proving soundness of Micali's CS-proofs [Mic94] based on any falsifiable assumptions. In contrast to [GW11], our result also holds for *non-adaptive* cheating provers, who choose the instance before seeing the verifier's message.

Roughly speaking, an assumption is said to be falsifiable [Nao03] if it can be modeled as an interactive game between the adversary and a challenger who can efficiently decide if the adversary won the game. This includes essentially all standard assumptions used in the cryptographic literature, such as one-way functions, trapdoor permutations, RSA, DDH, LWE, etc.

We now elaborate on our results.

Black-box impossibility result for the Fiat-Shamir paradigm. As our main result, we show that for any sub-exponentially hard language L, if a public-coin interactive proof (or argument) system for L is *honest-verifier zero-knowledge* (HVZK) against sub-exponential size distinguishers, then the Fiat-Shamir heuristic applied to this protocol cannot be proven sound via a black-box reduction to a falsifiable assumption.

Moreover, we argue that many of the known protocols are honest-verifier zero-knowledge against sub-exponential size distinguishers. For example, the well known 3-round (Σ-)protocol for quadratic residuosity (QR) due to Blum [Blu81], is *perfect* HVZK . Thus, our result implies that the Fiat-Shamir heuristic cannot be proven sound (via a black-box reduction to a falsifiable assumption) when applied to Blum's QR-protocol, assuming it is hard to break the QR assumption in sub-exponential time. Similarly, Blum's protocol for Graph Hamiltonicity [Blu87] is HVZK against sub-exponential size distinguishers assuming the underlying commitment scheme is hiding against sub-exponential size adversaries. Since Graph Hamiltonicity is an NP-complete language, our result implies that the Fiat-Shamir heuristic cannot be proven sound (via a black-box reduction to a falsifiable assumption) when applied to Blum's Graph Hamiltonicity protocol, assuming the underlying commitment scheme is hiding against sub-exponential size adversaries and the existance of NP languages that are sub-exponentially hard.

More generally, we show that for any $T(\kappa)$-hard NP language L, for any public-coin interactive protocol for L with messages of length $O(\log T)$, if the protocol is T-honest-verifier zero-knowledge (T-HVZK) then the soundness of the Fiat-Shamir heuristic applied to this protocol cannot be proven via a black-box reduction to a T-hard falsifiable assumption. Very roughly, we say that protocol is T-HVZK if an *honest* interaction between the prover and the verifier can be simulated in time $\mathrm{poly}(T)$ such that no distinguisher running in time $\mathrm{poly}(T)$ can distinguish between a simulated view from the real view. We refer the reader to Section 2 for the formal definitions.

Note that many protocols (such as the two protocols mentioned above; see [CD09, AJL+12] for more examples) have constant soundness error and are therefore repeated many times in parallel to reduce the soundness error. To save on communication, it is desirable to repeat the protocol only $\ell = \mathrm{poly}\log(\kappa)$ times since this already achieves negligible soundness error. However, our main result implies that if the language L (being proved) is quasi-polynomially hard, and the protocol is $\kappa^{\log \kappa}$-HVZK (which is the case for many of the protocols for which we would like to apply the Fiat-Shamir heuristic), then the Fiat-Shamir transformation cannot be proven sound via a black-box reduction to a (super-polynomially hard) falsifiable assumption.

On the connection between zero-knowledge and Fiat-Shamir. We note that the connection between zero-knowledge and the (in)security of Fiat-Shamir paradigm was already made in prior works. In particular, Dwork et al. [DNRS99] showed that if a public-coin interactive protocol is "weakly" zero-knowledge (where the ZK property is weakened by changing the order of quantifiers in the standard ZK definition, but requiring the simulator and distinguisher to be polynomial time) then the Fiat-Shamir heuristic applied to this protocol is not sound. We note however, that known public-coin protocols where Fiat-Shamir heuristic would typically be applied, are *not* known to satisfy their zero-knowledge property. In contrast, we only require the protocol to be honest-verifier zero-knowledge w.r.t. sub-exponential adversaries, and show that this property is satisfied by many well-known protocols (under some assumptions). We refer the reader to Section 1.2 for a more detailed comparison of our results with prior works.

Separating CS-proofs from falsifiable assumptions. We use the above results to obtain a black-box impossibility result for CS-proofs, as defined by Micali [Mic94]. A CS-proof is a 2-round argument for any NP statement,[1] where the communication complexity depends only on the security parameter κ, and is independent of the instance length. CS-proofs are obtained by applying the Fiat-Shamir heuristic to Kilian's 4-round succinct argument [Kil92]. We show that there exist PCPs (with zero-knowledge properties), such that when using these PCPs, Kilian's protocol is perfect honest-verifier zero-knowledge. Thus, we conclude that the resulting CS-proofs cannot be proven secure via a black-box reduction to a falsifiable assumption if the underlying language is sub-exponentially hard. We elaborate on this result in Section 5.

On 2-round zero-knowledge. Finally, we note that the technique underlying our black-box impossibility results is an extension of the negative result for 2-round zero-knowledge due to Goldreich and Oren [GO94]. Specifically, we show that for any T-hard language L, there does not exist a 2-round argument system for L that is (a) (auxiliary-input) T-zero-knowledge (i.e., where the view of every PPT cheating verifier can be simulated in time poly(T) so that no poly(T)-time distinguisher can distinguish between the real view and the simulated view), and (b) soundness can be proven via a black-box reduction to a T-hard falsifiable assumption. We believe this result is of independent interest, and elaborate on it in Section 3. We refer the reader to Section 1.2 for a comparison of our result with the work of Goldreich and Oren [GO94] and the work of Pass [Pas03].

A note on our black-box impossibility results. We clarify that our results are different from most black-box impossibility results in the literature (e.g., [IR89, Sim98, GKM+00, GMR01, RTV04, BMG07, BMG09]) which show that some primitive A (e.g., key-agreement) cannot be constructed by making black-box use of another primitive B (e.g., one-way functions). In contrast, our black-box separation results follow the paradigm of showing that the security of a concrete scheme cannot be based on a large class of concrete assumptions when the reduction treats the attacker as a black-box. Known examples of such results include [Cor02, DOPS04, HH09, GW11, Pas11, DHT12].

1.2 Related Work

Fiat-Shamir paradigm. Dwork et al. [DNRS99] (and independently, Hada and Tanaka [HT98]) observed the relation between the existence of 3-round zero-knowledge proofs and the (in)security of the Fiat-Shamir paradigm. Very roughly, the intuition is that if a given 3-round public-coin protocol is zero-knowledge, then for *every* verifier, there exists a simulator that can simulate the verifier's view. Observe that the Fiat-Shamir paradigm uses a hash function as the program of the verifier. Thus, one may simply use the zero-knowledge simulator as the cheating prover for the resultant 2-round protocol. Indeed, this idea is more general and extends to any k-round public-coin zero-knowledge proof (or argument), and hence already implies the insecurity of the Fiat-Shamir heuristic when applied to the *sequential* repetition of many basic 3-round zero-knowledge protocols, such as Blum's protocol for Graph Hamiltonicity [Blu87].

In fact, Dwork et al. [DNRS99] showed that the above intuition also holds for much weaker definitions of zero-knowledge. However, in all of their definitions, the simulator (and the distinguisher) were always required to be *efficient*. Indeed, this is crucial for their approach. This stands in sharp contrast to our approach, where we consider super-polynomial time simulators and distinguishers. Nevertheless, we find it instructive to compare our approach with theirs.

[1] More generally, CS-proofs are 2-round arguments for any language in NEXP, and were initially defined in [Mic94] for languages in EXP.

Following the work of [DNRS99], Barak [Bar01] presented a constant-round public-coin zero-knowledge argument for which the Fiat-Shamir transformation is not sound. Later, Goldwasser and Kalai [GK03] presented such an example for a 3-round argument system. We note that both of these negative results are for *arguments*. In contrast, our results are also applicable to interactive *proofs*.

In the context of interactive proofs, Barak, Lindell and Vadhan [BLV03] presented a security definition for the Fiat-Shamir hash function, namely, *entropy-preserving hash functions*, which if realized, would imply the security of the Fiat-Shamir paradigm applied to any constant-round public-coin interactive proof system. Barak et al. gave plausible conjectures under which this notion could be realized, but left open the problem of realizing it under standard complexity assumptions. Recently, Dodis, Ristenpart and Vadhan [DRV12] showed that under specific *non-falsifiable* assumptions regarding the existence of robust randomness condensers for seed-dependent sources, the definition of [BLV03] can be realized. Our results show that the security notion of [BLV03] cannot be realized under falsifiable assumptions, using standard (i.e. black-box) proof techniques.

Two-round zero-knowledge. Goldreich and Oren [GO94] showed the impossibility of constructing 2-round zero-knowledge proofs (and arguments) for languages outside BPP. More generally, their results extend in a straightforward manner to show the impossibility of constructing 2-round T-zero-knowledge[2] proofs or arguments for T-hard languages, that are sound against cheating provers running in time $\text{poly}(T(\kappa))$. Our black-box impossibility result for 2-round zero-knowledge (see Theorem 3.1) can be viewed as an extension of their results since we also rule out achieving soundness against *polynomial-time* cheating provers (based on T-hard falsifiable assumptions). Indeed this difference is crucial for obtaining our negative results. It is also interesting to compare Theorem 3.1 with the work of Pass [Pas03], who gave a positive result for 2-round zero-knowledge arguments where the underlying assumption is T'-hard for T' that is strictly more than the running time of the distinguisher (and strictly less than the running time of the simulator). Thus, Theorem 3.1 can be seen as essentially tight.

Succinct non-interactive arguments. The work of Gentry and Wichs [GW11] gives black-box impossibility results for constructing *succinct* non-interactive arguments from falsifiable assumptions. In particular, their results also imply the impossibility of proving the soundness of Micali's CS-proofs via black-box reduction to a falsifiable assumption, against *adaptive* adversaries who may choose the instances *after* the verifier's first message is fixed. Indeed, ruling out *non-adaptive* adversaries (in the general context of succinct non-interactive arguments) was left as an open problem in [GW11]. As such, our negative result on CS-proofs provides a partial answer to their question.

We also note that our techniques for proving Theorem 3.1 are similar to [GW11]. In particular, we encounter some similar technical challenges as in [GW11], such as dealing with reductions that "lie" about the security parameter.

1.3 Concurrent Work

In a concurrent and independent work, Bitansky, Garg and Wichs [BGW12] give a similar but incomparable result to ours. Both works show that the soundness of the Fiat-Shamir heuristic cannot be proven via black-box reductions to a large class of "standard" assumptions. However, the two works approach the problem from two different technical angles: [BGW12] focuses on the

[2]Recall that in T-zero-knowledge protocols the simulator and the distinguisher run in time $\text{poly}(T(\kappa))$.

impossibility of entropy-preserving hash functions [BLV03] and the implied Fiat-Shamir impossibility, whereas we show the impossibility of black-box reductions for proof systems satisfying a natural honest-verifier ZK requirement. The different techniques lead to two main differences in the end results: The work of [BGW12] gives an impossibility result for a "universal" Fiat-Shamir compiler that must preserve soundness when applied to *any* "3-round public-coin proof". In contrast, we not only show that a universal compiler cannot exist, but also show impossibility results for several specific proof systems. On the other hand, [BGW12] shows a separation from a larger class of assumptions, consisting of "cryptographic games with an unbounded challenger", whereas we show a separation from falsifiable assumptions where the challenger is efficient. A merged version of the two works [BDSK+13] will appear at TCC 2013.

1.4 Organization

The rest of this paper is organized as follows. We start by setting up notation and recalling some important definitions in Section 2. In Section 3, we give a black-box impossibility result for two-round zero-knowledge arguments. Then, building on this result, we present our black-box impossibility result for the Fiat-Shamir paradigm in Section 4, and for CS proofs in Section 5.

Notation Throughout this paper, we often use κ to denote the security parameter. A function $\mu(\cdot)$ is called *negligible* if for all polynomials p and all sufficiently large κ, $\mu(\kappa) \leq 1/p(\kappa)$. We use $\text{negl}(\cdot)$ to denote a negligible function. We say that an event $E = E(\kappa)$ occurs with overwhelming probability if it occurs with probability $1 - \text{negl}(\kappa)$.

2 Preliminaries

Definition 2.1. *Two distribution families $\mathcal{X} = \{\mathcal{X}_\kappa\}_{\kappa\in\mathbb{N}}$ and $\mathcal{Y} = \{\mathcal{Y}_\kappa\}_{\kappa\in\mathbb{N}}$ are said to be T-*indistinguishable *(denoted by $\mathcal{X} \overset{T}{\approx} \mathcal{Y}$) if for every circuit family $D = \{D_\kappa\}_{\kappa\in\mathbb{N}}$ of size $\text{poly}(T(\kappa))$,*

$$\text{Adv}_D^{\mathcal{X},\mathcal{Y}}(S) \overset{\text{def}}{=} |\Pr[D(x) = 1] - \Pr[D(y) = 1]| = \text{negl}(T(\kappa)),$$

where the probabilities are over $x \leftarrow \mathcal{X}_\kappa$ and over $y \leftarrow \mathcal{Y}_\kappa$.

2.1 Hard Languages

Definition 2.2. *For any $T = T(\kappa)$, an* NP *language L is said to be T-*hard *if there exist two distribution families $\mathcal{X} = \{\mathcal{X}_\kappa\}_{\kappa\in\mathbb{N}}$ and $\bar{\mathcal{X}} = \{\bar{\mathcal{X}}_\kappa\}_{\kappa\in\mathbb{N}}$, and a PPT sampling algorithm* Samp *such that:*

- *For every $\kappa \in \mathbb{N}$ the support of $\mathcal{X}_\kappa$ is in L and the support of $\bar{\mathcal{X}}_\kappa$ is in $\bar{L}$.*
- *The distributions $\mathcal{X}$ and $\bar{\mathcal{X}}$ are $T(\kappa)$-indistinguishable.*
- *The support of the sampling algorithm* Samp *consists of elements (x, w) such that $R(x, w) = 1$, and its projection to the first coordinate yields the distribution $\mathcal{X} = \{\mathcal{X}_\kappa\}_{\kappa\in N}$.*

  Note that since Samp *is efficient, the distribution family $\mathcal{X}$ is efficiently sampleable. There are no constraints on the size of the instances in $\mathcal{X}_\kappa$ or $\bar{\mathcal{X}}_\kappa$, however since $\mathcal{X}$ is efficiently sampleable each $x \leftarrow \mathcal{X}_\kappa$ is of size at most $\text{poly}(\kappa)$.*

An NP *language is said to be* sub-exponentially hard *if it is* 2^κ*-hard.*[3]

2.2 Zero-Knowledge Arguments

Definition 2.3. *A 2-round argument system for an* NP *language* L *with corresponding relation* R *is a tuple of efficient algorithms* $\Pi = (\mathcal{P}, \mathcal{V} = (\mathcal{V}_1, \mathcal{V}_2))$ *with syntax:*

- $ch = \mathcal{V}_1(1^\kappa, x; r)$*: On input a security parameter* 1^κ*, an instance* x*, and randomness* r*, the verifier* $\mathcal{V}_1$ *outputs a challenge* ch*.*
- $\pi \leftarrow \mathcal{P}(1^\kappa, x, w, ch)$*: On input a security parameter* 1^κ*, an instance* x *together with a corresponding witness* w*, and a challenge* ch*, the prover* $\mathcal{P}$ *outputs an argument* π*. This algorithm may be randomized.*
- $0/1 = \mathcal{V}_2(1^\kappa, x, r, \pi)$*: On input a security parameter* 1^κ*, an instance* x*, the randomness* r *(which is the same randomness as that of* $\mathcal{V}_1$*), and an argument* π*, the verifier* $\mathcal{V}_2$ *verifies whether the argument* π *is correct. We call the output of* $\mathcal{V}_2$*, the output of* $\mathcal{V}$*.*

We require Π *to satisfy the following properties:*

Completeness: *For all* $(x, w) \in R$ *and for all* $\kappa \in \mathbb{N}$*,*

$$\Pr\left[\mathcal{V}_2(1^\kappa, x, r, \pi) = 1 \;\middle|\; \begin{array}{l} ch = \mathcal{V}_1(1^\kappa, x; r) \\ \pi \leftarrow \mathcal{P}(1^\kappa, x, w, ch) \end{array}\right] = 1 - \mathrm{negl}(\kappa),$$

where the probability is over the randomness r *and over the randomness of the prover* $\mathcal{P}$*.*

Soundness: *For all poly-size (cheating provers)* $\mathcal{P}^* = (\mathcal{P}_1^*, \mathcal{P}_2^*)$*,*

$$\Pr\left[\mathcal{V}_2(1^\kappa, x, r, \pi) = 1 \;\wedge\; \bar{x} \notin L \;\middle|\; \begin{array}{l} \bar{x} \leftarrow \mathcal{P}_1^*(1^\kappa) \\ ch = \mathcal{V}_1(1^\kappa, \bar{x}; r) \\ \pi \leftarrow \mathcal{P}_2^*(1^\kappa, \bar{x}, ch) \end{array}\right] = \mathrm{negl}(\kappa),$$

where the probability is over the randomness r*.*

Definition 2.4. *For any* $T = T(\kappa)$*, we say that a 2-round argument system* $\Pi = (\mathcal{P}, \mathcal{V})$ *for an* NP *language* L *is* (auxiliary-input) T-zero-knowledge *if for every poly-size circuit* $\mathcal{V}^*$ *there exists a simulator* $\mathcal{S}_{\mathcal{V}^*}(1^\kappa)$ *of size* $\mathrm{poly}(T(\kappa))$ *such that for every* $\kappa \in \mathbb{N}$*, every instance* $x \in L$ *of length at most* $\mathrm{poly}(\kappa)$ *with a corresponding witness* w*, and every auxiliary input* $z \in \{0,1\}^{\mathrm{poly}(\kappa)}$*, it holds that for every non-uniform distinguisher* $D = \{D_\kappa\}$ *of size* $\mathrm{poly}(T(\kappa))$

$$\left|\Pr[D\left((\mathcal{P}(w), \mathcal{V}^*(z))(1^\kappa, x)\right) = 1] - \Pr[D\left(\mathcal{S}_{\mathcal{V}^*}(1^\kappa, x, z)\right) = 1]\right| \leq \mathrm{negl}(T(\kappa)),$$

where $(\mathcal{P}(w), \mathcal{V}^*(z))(1^\kappa, x)$ *denotes the view of the verifier* $\mathcal{V}^*$ *after interacting with the honest prover on input security parameter* κ*, statement* $x \in L$*, auxiliary input* z*, and* $\mathcal{S}_{\mathcal{V}^*}(1^\kappa, x, z)$ *denotes the output of the simulator* $\mathcal{S}_{\mathcal{V}^*}$ *on input* $(1^\kappa, x, z)$*.*

[3] Note that it should be hard for a $\mathrm{poly}(2^\kappa)$-time distinguisher to distinguish between elements in $\mathcal{X}_\kappa$ and elements in $\bar{\mathcal{X}}_\kappa$, where these elements can be much longer than κ, and can be of length κ^ϵ for any constant $\epsilon > 0$ (thus, capturing the *sub*-exponential hardness).

2.3 Falsifiable Assumptions and Black-Box Reductions

In what follows, we recall the notion of falsifiable assumptions as defined by Naor [Nao03]. We follow the formalization of Gentry and Wichs [GW11].

Definition 2.5. *For $T = T(\kappa)$, a* T-hard falsifiable assumption *consists of a* PPT *interactive challenger $\mathcal{C}(1^\kappa)$ that runs in time* $\mathrm{poly}(\kappa)$ *and a constant $\delta \in [0, 1)$. The challenger $\mathcal{C}$ interacts with a machine $\mathcal{A}$ and may output a special symbol* win*. If this occurs, $\mathcal{A}$ is said to win $\mathcal{C}$. For any adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ over $\mathcal{C}$ is defined as:*

$$\mathrm{Adv}_{\mathcal{A}}^{(\mathcal{C},\delta)}(T) = \left|\Pr[\mathcal{A}(1^\kappa) \text{ wins } \mathcal{C}(1^\kappa)] - \delta\right|,$$

where the probability is taken over the random coins of $\mathcal{A}$ and $\mathcal{C}$. The assumption associated with the tuple $(\mathcal{C}, \delta)$ states that for every (non-uniform) adversary $\mathcal{A}(1^\kappa)$ running in time $\mathrm{poly}(T(\kappa))$*,*

$$\mathrm{Adv}_{\mathcal{A}}^{(\mathcal{C},\delta)}(T) = \mathrm{negl}(T(\kappa)).$$

If the advantage of $\mathcal{A}$ is non-negligible in $T(\kappa)$ then $\mathcal{A}$ is said to break *the assumption.*

Definition 2.6. *Let Π be a 2-round argument system. We say that the soundness of Π can be proven via a* black-box reduction *to a $T(\kappa)$-hard falsifiable assumption, if there is an oracle-access machine $\mathcal{R}^{(\cdot)}$ such that for every (possibly inefficient) Π-adversary $\mathcal{P}^*$, the machine $\mathcal{R}^{\mathcal{P}^*}$ runs in time* $\mathrm{poly}(T(\kappa))$ *and breaks the assumption.*

3 Black-Box Impossibilty for 2-Round Zero Knowledge

In this section, we give a black-box impossibility result for 2-round zero-knowledge arguments. Our theorem extends the negative result of Goldreich and Oren [GO94], and can be seen as essentially tight, in view of the positive result of Pass [Pas03]. We refer the reader to Section 1.2 for a comparison of our result with [GO94] and [Pas03].

We now state our main technical theorem:

Theorem 3.1. *For any $T(\kappa)$ and any T-hard language L, there does not exist a 2-round argument system Π for L such that:*

- *Π is (auxiliary-input) T-zero-knowledge, and*
- *the soundness of Π can be proven via a black-box reduction to a T-hard falsifiable assumption,*

unless the assumption is false.

Theorem 3.1, which we believe to be of independent interest, is also the starting point for our impossibility results for the Fiat-Shamir paradigm (see Section 4) and for CS proofs (see Section 5).

Proof Idea. Consider a 2-round argument system Π for a T-hard language L that is (auxiliary-input) T-zero-knowledge. We prove, by contradiction, that the soundness of Π cannot be proven via a black-box reduction to a T-hard falsifiable assumption. Let n be a security parameter and suppose that there exists a $\mathrm{poly}(T(n))$-time black-box reduction $\mathcal{R}$ such that given black-box oracle access to any cheating prover $\mathcal{P}^*$, uses this oracle to break a $T(n)$-hard falsifiable assumption. By

the definition of a $T(n)$-hard falsifiable assumption (see Definition 2.5) and the definition of a black-box reduction (see Definition 2.6), we know the reduction $\mathcal{R}$ runs in time $\text{poly}(T(n))$.

By naturally extending Goldreich and Oren's 2-round zero-knowledge impossibility result [GO94], we first prove that the T-zero-knowledge simulator $\mathcal{S}$ always produces an accepting transcript, even when given a statement $x \in \bar{L}$ (see Lemma 3.2). Thus, we may view $\mathcal{S}$ as a cheating prover. This means that $\mathcal{R}$ breaks the assumption when given oracle access to $\mathcal{S}$ (and $\mathcal{S}$ is given $x \in \bar{L}$). For brevity, we say that $\mathcal{R}^{\mathcal{S}(x\in\bar{L})}$ breaks the assumption. However, we must be careful because the reduction $\mathcal{R}$ may "lie" about the security parameter and run $\mathcal{S}$ with security parameter $\kappa \neq n$. Throughout the proof will denote by n the security parameter of the underlying falsifiable assumption, and denote by κ the security parameter that the reduction uses when calling $\mathcal{S}$ (though the reduction $\mathcal{R}$ may call $\mathcal{S}$ many times with different security parameters). Note that the bound on the running time of $\mathcal{R}$ means $\kappa \leq T(n)$.

Our approach is to show that oracle access to $\mathcal{S}(x \in \bar{L}_\kappa)$ can be simulated in time $\text{poly}(T(n))$ regardless of the value of κ. If $\kappa \leq n$ then $\mathcal{S}(x \in \bar{L}_\kappa)$ runs in time $\text{poly}(T(k)) \leq \text{poly}(T(n))$ and we are done. However, if $\kappa > n$ then we show that if $\mathcal{R}^{\mathcal{S}(x\in\bar{L}_\kappa)}$ breaks the assumption then so does $\mathcal{R}^{\mathcal{P}(x\in L_\kappa, w)}$, where w is a valid witness for $x \in L_\kappa$ and $\mathcal{P}$ is the honest prover. Since $\mathcal{P}(x \in L_\kappa, w)$ runs in time $\text{poly}(\kappa) \leq \text{poly}(T(n))$, this means we can simulate $\mathcal{S}(x \in \bar{L}_\kappa)$ in time $\text{poly}(T(n))$.

We now proceed to give the formal proof.

Proof. As a first step in the proof of Theorem 3.1, we prove the following lemma.

Lemma 3.2. *For any $T(\kappa)$, any T-hard language L (w.r.t. distributions $\mathcal{X}, \bar{\mathcal{X}}$), and any 2-round argument system $\Pi = (\mathcal{P}, \mathcal{V})$ for L that is (auxiliary-input) T-zero-knowledge, the following holds. Let $\mathcal{V}^*$ be the verifier that on auxiliary input z simply outputs z as its challenge ch. Let $\mathcal{S}_{\mathcal{V}^*}$ be the corresponding (auxiliary-input) zero-knowledge simulator. Then, for every (possibly cheating) verifier $\hat{\mathcal{V}}$ of size $\text{poly}(T(\kappa))$,*

$$\{\mathsf{View}_\kappa\} \overset{T}{\approx} \{\mathsf{SimView}_\kappa\}$$

where View_κ and $\mathsf{SimView}_\kappa$ are defined as follows.

$$\mathsf{View}_\kappa = (1^\kappa, x, r, \pi)$$

where $(x, w) \leftarrow \mathsf{Samp}(1^\kappa)$, r is randomly chosen, $ch = \hat{\mathcal{V}}(1^\kappa, x; r)$, and $\pi = \mathcal{P}(1^\kappa, x, w, ch)$. Similarly,

$$\mathsf{SimView}_\kappa = (1^\kappa, \bar{x}, r, \bar{\pi})$$

where $\bar{x} \leftarrow \bar{\mathcal{X}}_\kappa$, r is randomly chosen, $ch = \hat{\mathcal{V}}(1^\kappa, \bar{x}; r)$, and $\pi \leftarrow \mathcal{S}_{\mathcal{V}^*}(1^\kappa, \bar{x}, ch)$, where $\mathcal{V}^*$ is the cheating verifier who takes ch as auxiliary input, and simply sends ch to the prover.

We note that the proof of this lemma builds upon ideas of Goldreich and Oren [GO94].

Proof. Let $\hat{\mathcal{V}}$ be any (possibly cheating) PPT verifier for Π. We first argue that by the (auxiliary-input) T-zero-knowledge property of Π w.r.t. non-uniform distinguishers, it must be the case that

$$\{\mathsf{View}_\kappa\} \overset{T}{\approx} \{\mathsf{SimView}'_\kappa\} \tag{3.1}$$

where $\mathsf{SimView}'_\kappa$ is defined exactly as $\mathsf{SimView}_\kappa$, with the only difference being that $x \leftarrow \mathcal{X}_\kappa$. Namely, $\mathsf{SimView}'_\kappa = (1^\kappa, x, r, \bar{\pi})$, where $x \leftarrow \mathcal{X}_\kappa$, r is randomly chosen, $ch = \hat{\mathcal{V}}(1^\kappa, x; r)$, and

$\bar{\pi} \leftarrow \mathcal{S}_{\mathcal{V}^*}(1^\kappa, x, ch)$. We note that Equation (3.1) actually holds for every x in the language (as opposed to only for x distributed according to $\mathcal{X}_\kappa$).

This is the case since otherwise there exists a (non-uniform) distinguisher $\mathcal{D}$ that runs in time $\text{poly}(T(\kappa))$, and given a transcript $(1^\kappa, x, ch, \pi)$, and auxiliary input r such that $ch = \hat{\mathcal{V}}(1^\kappa, x; r)$, distinguishes between a simulated transcript $(1^\kappa, x, ch, \pi) \leftarrow \mathcal{S}_{\mathcal{V}^*}(1^\kappa, x, ch)$ and a real transcript $(1^\kappa, x, ch, \pi) \leftarrow (\mathcal{P}(w), \mathcal{V}^*(ch))(1^\kappa, x)$ with non-negligible probability (in $T(\kappa)$), contradicting the T-zero-knowledge property.

We now claim that the fact that the language is T-hard w.r.t. $\mathcal{X}$ and $\bar{\mathcal{X}}$, implies that

$$\{\mathsf{SimView}'_\kappa\} \overset{T}{\approx} \{\mathsf{SimView}_\kappa\}. \tag{3.2}$$

This is the case, since if there exists a $\text{poly}(T(\kappa))$-size distinguisher $\mathcal{D}$ that distinguishes between these two views with non-negligible probability (in $T(\kappa)$), then there exists a $\text{poly}(T(\kappa))$-size adversary $\mathcal{A}$ that distinguishes between $x \leftarrow \mathcal{X}_\kappa$ and $\bar{x} \leftarrow \bar{\mathcal{X}}_\kappa$ with non-negligible probability (in $T(\kappa)$), contradicting the T-hardness of L.

Upon receiving an instance z, where $z \leftarrow \mathcal{X}_\kappa$ or $z \leftarrow \bar{\mathcal{X}}_\kappa$, the adversary $\mathcal{A}$ chooses r at random, computes $ch = \hat{\mathcal{V}}(1^\kappa, z; r)$, computes $\pi \leftarrow \mathcal{S}_{\mathcal{V}^*}(1^\kappa, z, ch)$, and outputs $\mathcal{D}(1^\kappa, z, r, \pi)$. By the definition of $\mathcal{D}$, the adversary $\mathcal{A}$ indeed distinguishes between $x \leftarrow \mathcal{X}_\kappa$ and $\bar{x} \leftarrow \bar{\mathcal{X}}_\kappa$ with non-negligible probability (in $T(\kappa)$), contradicting the T-hardness of L.

Equations (3.1) and (3.2) together imply that

$$\{\mathsf{View}_\kappa\} \overset{T}{\approx} \{\mathsf{SimView}_\kappa\}$$

as desired. □

Proof of Theorem 3.1: Fix any $T = T(\kappa)$ and any T-hard language L (w.r.t. distributions $\mathcal{X}, \bar{\mathcal{X}}$). Let $\Pi = (\mathcal{P}, \mathcal{V})$ be any 2-round argument system for L that is (auxiliary-input) T-zero-knowledge. Suppose that there exists a $\text{poly}(T)$-time black-box reduction $\mathcal{R}$, that has black-box oracle access to a cheating prover $\mathcal{P}^*$, and uses this oracle to break the underlying T-hard falsifiable assumption.

By the definition of a T-hard falsifiable assumption (see Definition 2.5) and by the definition of a black-box reduction (see Definition 2.6), the reduction $\mathcal{R}$ interacts with a challenger $\mathcal{C}(1^n)$, it runs in time $\text{poly}(T(n))$, and it uses oracle access to $\mathcal{P}^*$ to win the challenger's game. Throughout this proof we denote by n the security parameter of the underlying falsifiable assumption, and we denote by κ the security parameter that the reduction uses when calling $\mathcal{P}^*$ (though the reduction $\mathcal{R}$ may call $\mathcal{P}^*$ many times with different security parameters).

Loosely speaking, we prove that oracle access to the cheating prover $\mathcal{P}^*$ can be simulated in time $\text{poly}(T(n))$, which implies that the underlying falsifiable assumption can be broken in time $\text{poly}(T(n))$, thus contradicting T-hardness of the assumption.

To this end, let $\mathcal{P}^* = (\mathcal{P}_1^*, \mathcal{P}_2^*)$ be the cheating prover, defined as follows: Upon receiving a security parameter 1^κ the prover $\mathcal{P}_1^*(1^\kappa)$ samples $\bar{x} \leftarrow \bar{\mathcal{X}}_\kappa$, and upon receiving a challenge ch the prover $\mathcal{P}_2^*(1^\kappa, \bar{x}, ch)$ runs $\mathcal{S}_{\mathcal{V}^*}(1^\kappa, \bar{x}, ch)$ and returns the proof π output by $\mathcal{S}_{\mathcal{V}^*}$. Lemma 3.2, together with the completeness property of Π (see Definition 2.3), implies that $\mathcal{P}^*$ provides an accepting transcript with overwhelming probability (in κ).

At first sight, the reader may think that since Π is T-zero-knowledge, $\mathcal{P}^*$ can be trivially simulated in time $\text{poly}(T(n))$. However, note that $\mathcal{R}$ may query $\mathcal{P}^*$ on any security parameter κ of its choice. In particular, since $\mathcal{R}$ runs in time $\text{poly}(T(n))$, it may choose to query $\mathcal{P}^*$ with security

parameter $\kappa = T(n)$, in which case $\mathcal{P}^*$ will run in time $T(T(n))$, which may be significantly larger than $T(n)$. For example, if $T(n)$ is exponential in n then $T(T(n))$ is doubly exponential in n.

In order to prove that $\mathcal{P}^*$ can be simulated in time $\text{poly}(T(n))$, we distinguish between simulating $\mathcal{P}^*$ for (small) security parameters $\kappa \leq n$, and simulating $\mathcal{P}^*$ for (large) security parameters $\kappa > n$. For the former ($\kappa \leq n$), $\mathcal{P}^*$ can be perfectly simulated in time $\text{poly}(T(n))$, since $\mathcal{P}^*$ runs in time $\text{poly}(T(\kappa)) \leq \text{poly}(T(n))$.

We next use Lemma 3.2 to argue that $\mathcal{P}^*$ can also be simulated in time $\text{poly}(T(n))$ for $\kappa > n$. In this case, the simulation will not be perfect, but rather computational, w.r.t. $\text{poly}(T(n))$-time distinguishers. In particular, we will argue that if $\mathcal{R}$ breaks the assumption with oracle access to $\mathcal{P}^*$, it will also break the assumption with oracle access to the simulated version of $\mathcal{P}^*$.

Let $\mathcal{P}$ be a simulated version of $\mathcal{P}^*$, defined as follows: Upon receiving a security parameter 1^κ, the prover $\mathcal{P}$ samples $(x, w) \leftarrow \mathsf{Samp}(1^\kappa)$, and sends x to $\mathcal{R}$; and upon receiving a challenge ch (corresponding to x) from the reduction $\mathcal{R}$, the prover $\mathcal{P}$ sends $\pi \leftarrow \mathcal{P}_2(1^\kappa, x, w, ch)$ to $\mathcal{R}$.

Note that $\mathcal{P}(1^\kappa)$ runs in time $\text{poly}(\kappa)$. Furthermore, the fact that the reduction $\mathcal{R}$ runs in time at most $\text{poly}(T(n))$ implies that $\kappa \leq \text{poly}(T(n))$. Thus $\mathcal{P}$ runs in time at most $\text{poly}(T(n))$, as desired. It remains to argue that if $\mathcal{R}^{\mathcal{P}^*}$ wins the challenger's game with probability ϵ then $\mathcal{R}^{\mathcal{P}}$ wins the challenger's game with probability $\epsilon/2$. We will prove the following stronger claim.

Claim 3.3. *For any oracle machine $\mathcal{M}(1^n)$ running in time $\text{poly}(T(n))$,*

$$\mathcal{M}^{\mathcal{P}^*}(1^n) \overset{T}{\approx} \mathcal{M}^{\mathcal{P}}(1^n),$$

assuming $\mathcal{M}(1^n)$ queries its oracle only with security parameters $\kappa > n$.

Note that this claim immediately implies that $\mathcal{R}^{\mathcal{P}}$ wins the challenger's game with probability at least $\epsilon/2$ (and it actually implies that $\mathcal{R}^{\mathcal{P}}$ wins with probability $\epsilon - \text{negl}(T(n))$). To this end, consider the $\text{poly}(T(n))$-time oracle machine $\mathcal{M}$, that simulates the interaction between $\mathcal{R}$ and the challenger of the underlying falsifiable assumption "in his belly", and outputs 1 if and only if $\mathcal{R}$ wins the challenger. Note that if $\mathcal{M}$ has oracle access to $\mathcal{P}^*$ it will output 1 with probability ϵ, and therefore Claim 3.3 implies that if $\mathcal{M}$ has oracle access to $\mathcal{P}$ it must output 1 with probability at least $\epsilon - \text{negl}(T(n))$. Thus it remains to prove Claim 3.3, which follows from Lemma 3.2 together with a standard hybrid argument.

Proof of Claim 3.3. Fix a $\text{poly}(T(n))$-time oracle machine $\mathcal{M}(1^n)$. Let $q \leq \text{poly}(T(n))$ be an upper bound on the number of times that $\mathcal{M}$ invokes its oracle. We define a sequence of $q+1$ oracles: $H_0, H_1, \ldots, H_q$, where H_i is defined to be the oracle that behaves exactly like $\mathcal{P}^*$ during the first i runs of Π, and behaves exactly like $\mathcal{P}$ during the remaining runs of Π. Note that $H_0 = \mathcal{P}$ and $H_q = \mathcal{P}^*$. A standard hybrid argument implies that it suffices to prove that for every $i \in [q]$,

$$\mathcal{M}^{H_i}(1^n) \overset{T}{\approx} \mathcal{M}^{H_{i-1}}(1^n). \tag{3.3}$$

To this end, fix any $i \in [q]$, and suppose for the sake of contradiction that (3.3) does not hold. Namely, suppose that there exists a (non-uniform) $\text{poly}(T(n))$-time distinguisher $\mathcal{D}$ such that

$$\left| \Pr[\mathcal{D}(\mathcal{M}^{H_i}(1^n)) = 1] - \Pr[\mathcal{D}(\mathcal{M}^{H_{i-1}}(1^n)) = 1] \right| \geq \delta(n),$$

where $\delta(n)$ is a non-negligible function of $T(n)$. We construct a (non-uniform) $\text{poly}(T(n))$-size distinguisher that contradicts Lemma 3.2.

To this end, recall that both the oracles H_i and H_{i-1} behave exactly like $\mathcal{P}^*$ for the first $i-1$ proofs, and behave exactly like $\mathcal{P}$ from the $i+1$'st proof and onwards. Therefore, there exists some fixing of the first $i-1$ interactions between $\mathcal{M}$ and $\mathcal{P}^*$, such that conditioned on this fixing, $\mathcal{D}$ still succeeds in distinguishing between $\mathcal{M}^{H_i}(1^n)$ and $\mathcal{M}^{H_{i-1}}(1^n)$. Namely, denote by st the state of $\mathcal{M}$ after invoking the $\mathcal{P}^*$ oracle $i-1$ times. Then there exists a state $st \in \{0,1\}^{\mathrm{poly}(T(n))}$, and there exists a $\mathrm{poly}(T(n))$-time oracle machine $\mathcal{M}_{st}$ that invokes it oracle at most $q-i+1$ times, such that

$$\left|\Pr[\mathcal{D}(\mathcal{M}_{st}^{H})=1]-\Pr[\mathcal{D}(\mathcal{M}_{st}^{\mathcal{P}})=1]\right| \geq \delta(n),$$

where H behaves exactly like $\mathcal{P}^*$ during the first proof, and behaves exactly like $\mathcal{P}$ during the rest of the proofs.

Note that $\mathcal{P}$ runs in time $\mathrm{poly}(\kappa) \leq \mathrm{poly}(T(n))$, and thus $\mathcal{M}_{st}$ can simulate this oracle on its own. Namely, there exists a $\mathrm{poly}(T(n))$-time oracle machine $\mathcal{M}_{st}$ as above, that invokes its oracle only once, and simulates the rest of the oracle calls on its own. In other words, there exists a $\mathrm{poly}(T(n))$-time oracle machine $\mathcal{M}_{st}$ such that

$$\left|\Pr[\mathcal{D}(\mathcal{M}_{st}^{\mathcal{P}^*})=1]-\Pr[\mathcal{D}(\mathcal{M}_{st}^{\mathcal{P}})=1]\right| \geq \delta(n), \tag{3.4}$$

where $\mathcal{M}_{st}$ queries its oracle with security parameter $\kappa > n$. Note that the view of $\mathcal{M}_{st}^{\mathcal{P}^*}$ is exactly $\mathsf{SimView}_\kappa$ with $\hat{\mathcal{V}} = \mathcal{M}_{st}$. Similarly the view of $\mathcal{M}_{st}^{\mathcal{P}}$ is exactly View_κ with $\hat{\mathcal{V}} = \mathcal{M}_{st}$. Moreover, the distinguisher $\mathcal{D}$ runs in time $\mathrm{poly}(T(n)) \leq \mathrm{poly}(T(\kappa))$, and distinguishes with probability $\delta(n)$, which is non-negligible in $T(n)$, and thus is non-negligible in $T(\kappa)$. Therefore, Equation (3.4) contradicts Lemma 3.2.

□
□

4 Black-Box Impossibility for Fiat-Shamir Paradigm

In this section we give black-box impossibility results for the soundness of the Fiat-Shamir paradigm. Our results are applicable to both proofs and arguments. Moreover, our results are applicable both in the original setting of 3-round identification protocols, and more generally are applicable to constant round public-coin protocols (with certain properties).

For the sake of simplicity of notation, we present our results for the case of 4-round public-coin protocols. We note that although our techniques generalize to constant-round protocols, the case of 4-rounds already covers many interesting applications of the Fiat-Shamir paradigm, as presented in Section 4.1 and Section 5.

Let us first recall the Fiat-Shamir paradigm when applied to 4-round public-coin protocols. Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a 4-round public-coin proof (or argument) system for an NP language L. We denote its transcripts by $(h, \alpha, \beta, \gamma)$, where h, β are the messages sent by the verifier, and α, γ are the messages sent by the prover. Let $\{\mathcal{H}_\kappa^{\mathsf{FS}}\}_{\kappa \in \mathcal{K}}$ be an ensemble of hash functions. We define Π^{FS} to be the resulting 2-message protocol obtained by applying the Fiat-Shamir transformation to Π using $h^{\mathsf{FS}} \leftarrow \mathcal{H}_\kappa^{\mathsf{FS}}$. More formally, $\Pi^{\mathsf{FS}} = (\mathcal{P}^{\mathsf{FS}}, \mathcal{V}^{\mathsf{FS}})$ is presented in Figure 4.1.

Next we define *special honest-verifier (auxiliary-input) T-zero-knowledge*. We will later show the black-box impossibility results for protocols which have this property.

Definition 4.1. *For any $T = T(\kappa)$, we say that a 4-round public-coin proof (or argument) system $\Pi = (\mathcal{P}, \mathcal{V})$ for an NP language L is* (auxiliary-input) special honest-verifier T-zero-knowledge *if*

Protocol $\Pi^{\mathsf{FS}}(1^\kappa, x)$ for Language L

Prover's Input: Statement x and a witness w for $x \in L$.

Verifier's Input: Statement x.

$\mathcal{V}^{\mathsf{FS}} \to \mathcal{P}^{\mathsf{FS}}$: The verifier $\mathcal{V}^{\mathsf{FS}}$ chooses a random first message h for the protocol Π and chooses $h^{\mathsf{FS}} \leftarrow \mathcal{H}_\kappa^{\mathsf{FS}}$. It sends (h^{FS}, h) to the prover $\mathcal{P}^{\mathsf{FS}}$.

$\mathcal{P}^{\mathsf{FS}} \to \mathcal{V}^{\mathsf{FS}}$: The prover $\mathcal{P}^{\mathsf{FS}}$ simulates an execution with the prover $\mathcal{P}$ of Π in the following way:

- Choose a random tape for $\mathcal{P}$ and continue the emulation of $(\mathcal{P}, \mathcal{V})$ by running $\mathcal{P}$ on first message h. Let α be the second message sent by $\mathcal{P}$ in Π.
- Compute $h^{\mathsf{FS}}(\alpha) = \beta$.
- Continue the emulation of $\mathcal{P}$ assuming $\mathcal{P}$ received β as the third message from $\mathcal{V}^{\mathsf{FS}}$. Let γ be the fourth message sent by $\mathcal{P}$.

Send (α, β, γ) to the verifier $\mathcal{V}^{\mathsf{FS}}$.

Verification: The verifier $\mathcal{V}^{\mathsf{FS}}$ accepts iff:

- $h^{\mathsf{FS}}(\alpha) = \beta$.
- $\mathcal{V}$ accepts the transcript $(h, \alpha, \beta, \gamma)$.

Figure 4.1: The 2-round Collapsed Protocol Π^{FS}

there exists a simulator $\mathcal{S}(1^\kappa)$ *of size* $\mathrm{poly}(T(\kappa))$ *such that for every* $\kappa \in \mathbb{N}$*, every instance* $x \in L$ *of length at most* $\mathrm{poly}(\kappa)$ *with a corresponding witness* w*, every auxiliary input* $z \in \{0,1\}^{\mathrm{poly}(\kappa)}$*, and every random tape* (h, β) *of the verifier it holds that for every non-uniform distinguisher* $D = \{D_\kappa\}$ *of size* $\mathrm{poly}(T(\kappa))$

$$\left| \Pr[D\left((\mathcal{P}(w), \mathcal{V}(z, h, \beta))(1^\kappa, x)\right) = 1] - \Pr[D\left(\mathcal{S}(1^\kappa, x, z, h, \beta)\right) = 1] \right| \leq \mathrm{negl}(T(\kappa)),$$

where $(\mathcal{P}(w), \mathcal{V}(z, h, \beta))(1^\kappa, x)$ *denotes the view of the honest verifier* $\mathcal{V}$ *after interacting with the honest prover on input security parameter* κ*, statement* $x \in L$*, auxiliary input* z*, and random tape* (h, β)*, and* $\mathcal{S}(1^\kappa, x, z, h, \beta)$ *denotes the output of the simulator* $\mathcal{S}$ *on the corresponding inputs.*

We note that special honest verifier zero knowledge differs from honest verifier zero knowledge since the simulator must successfully simulate the view of the honest verifier for *every* given random tape (h, β).

We now state the main theorem of this section:

Theorem 4.2. *For any* $T(\kappa)$ *and any* T*-hard language* L*, let* Π *be a 4-round public-coin proof (or argument) system for* $\mathcal{L}$ *with* $2^{|\beta|} \leq T(\kappa)$ *which is special honest verifier (auxiliary input)* T*-zero knowledge. Then, the soundness of* Π^{FS} *cannot be proven via a black-box reduction to a* T*-hard falsifiable assumption (unless the assumption is false).*

Note that many public-coin proof (or argument) systems (such as those discussed in Section 4.1) consist of ℓ parallel repetitions of a basic protocol where the length of the verifier's message is a constant number of bits (or may depend logarithmically on the size of the instance x). To save on communication, it is desirable to repeat the protocol only $\ell = \text{poly}\log(\kappa)$ times, since this already achieves negligible soundness error. For such protocols, Theorem 4.2 implies that if the language L is quasi-polyomially hard, then the Fiat-Shamir transformation applied to this protocol cannot be proven sound via a black-box reduction to a falsifiable assumption.

Given Theorem 4.2, one may hypothesize that the Fiat-Shamir transformation, when applied to protocols of the type discussed above, can in fact be proven secure (via a black-box reduction to a falsifiable assumption) when the number of parallel repetitions is increased to $\ell = \text{poly}(\kappa)$. However, we show that this is not the case; for many protocols of interest, the impossibility result holds even when the number of repetitions ℓ, is greater than the hardness of the language.

Corollary 4.3. *Let L be a sub-exponentially hard language and let Π be a 4-round public-coin proof (or argument) system for L with the following properties:*

- *The length of the third message, β, is polynomial in the security parameter, κ, and is independent of the length of the instance, x.*
- *Π is special honest verifier (auxiliary input) $2^{|\beta|}$-zero knowledge.*

Then, the soundness of Π^{FS} cannot be proven via a black-box reduction to a $2^{|\beta|}$-hard falsifiable assumption (unless the assumption is false).

Corollary 4.3 follows from Theorem 4.2, as follows. Recall that a language is said to be sub-exponentially hard if it is T-hard for $T(\kappa) = 2^{\kappa}$ (see Definition 2.2). Namely, if there exist distributions $\mathcal{X}_\kappa$ and $\bar{\mathcal{X}}_\kappa$ over strings of length $\text{poly}(\kappa)$ that are 2^{κ}-indistinguishable, where $\mathcal{X}_\kappa$ is a distribution over instances in the language and $\bar{\mathcal{X}}_\kappa$ is a distribution over instances outside the language. Note that the length of these instances can be much larger than κ, and can be of length $\kappa^{1/\epsilon}$ for any constant $\epsilon > 0$.

We argue that any sub-exponentially hard language is also $2^{p(\kappa)}$-hard, for any polynomial p. This follows by simply taking $\mathcal{X}'_\kappa = \mathcal{X}_{p(\kappa)}$ and by taking $\bar{\mathcal{X}'}_\kappa = \bar{\mathcal{X}}_{p(\kappa)}$. Using this observation, Corollary 4.3 follows immediately from Theorem 4.2 by choosing $T(\kappa) = 2^{p(\kappa)}$ such that $|\beta| = p(\kappa)$.

4.1 Applications of Theorem 4.2 and Corollary 4.3

Typically (or at least traditionally), the Fiat-Shamir paradigm is applied to 3-round identification schemes, or more generally to what are called Σ-protocols. All these protocols are special honest-verifier zero-knowledge (see Definition 4.1). Therefore, Theorem 4.2 and Corollary 4.3 imply (black-box) negative results for the Fiat-Shamir paradigm when applied to any such protocol. In what follows we give two specific examples, keeping in mind that there are many other natural examples that we do not mention.

Perfect Zero-Knowledge Protocol for Quadratic Residuosity. Recall the language L_{QR} of quadratic residues.

$$L_{\mathsf{QR}} = \{(N, y) \mid \exists x \in \mathbb{Z}_N^* \text{ s.t. } y = x^2 \mod N\}$$

This language is assumed to be hard w.r.t. distributions $\mathcal{X}_\kappa$ and $\bar{\mathcal{X}}_\kappa$, defined as follows. In both distributions, N is sampled by sampling two random κ-bit primes p and q, and setting $N = pq$; in

$\mathcal{X}_\kappa$, the element y is a random quadratic residue, and in $\bar{\mathcal{X}}_\kappa$ the element y is a random quadratic non-residue with Jacobi symbol 1.

Recall the well-known *perfect* zero-knowledge Σ-protocol for quadratic residuosity with soundness $1/2$ [Blu81]. We denote by $\Pi^{\ell\text{-}\mathsf{QR}}$ the perfect special honest-verifier zero-knowledge protocol consisting of ℓ parallel executions of the basic Σ-protocol. We denote by $\Pi^{\mathsf{FS}(\ell\text{-}\mathsf{QR})}$ the protocol obtained when applying the Fiat-Shamir paradigm to $\Pi^{\ell\text{-}\mathsf{QR}}$. By applying Corollary 4.3, we obtain the following theorem:

Theorem 4.4. *For any $\ell = \ell(\kappa) = \mathrm{poly}(\kappa)$, if L_{QR} is sub-exponentially hard then the soundness of $\Pi^{\mathsf{FS}(\ell\text{-}\mathsf{QR})}$ cannot be proven via a black-box reduction to a falsifiable assumption (unless the assumption is false).*

Blum's Zero-Knowledge Protocol for NP. Recall the well-known Σ-protocol for NP of Blum [Blu87], based on the NP-complete problem of Graph Hamiltonicity, with soundness $1/2$. We denote by $\Pi^{\ell\text{-}\mathsf{Blum}}$ the special honest-verifier zero-knowledge protocol consisting of ℓ parallel executions of the basic Σ-protocol. Note that $\Pi^{\ell\text{-}\mathsf{Blum}}$ is special honest-verifier 2^ℓ-zero-knowledge, if the hiding property of the commitment scheme holds against 2^ℓ-size adversaries.[4]

We denote by $\Pi^{\mathsf{FS}(\ell\text{-}\mathsf{Blum})}$ the protocol obtained when applying the Fiat-Shamir paradigm to $\Pi^{\ell\text{-}\mathsf{Blum}}$. By applying Corollary 4.3, we obtain the following theorem:

Theorem 4.5. *For any $\ell = \ell(\kappa) = \mathrm{poly}(\kappa)$, if there exist NP languages L which are sub-exponentially hard, and if $\Pi^{\mathsf{FS}(\ell\text{-}\mathsf{Blum})}$ is instantiated with a commitment scheme whose hiding property holds against 2^ℓ-size adversaries, then the soundness of $\Pi^{\mathsf{FS}(\ell\text{-}\mathsf{Blum})}$ cannot be proven via a black-box reduction to a falsifiable assumption (unless the assumption is false).*

As noted above, one can apply Theorem 4.2 or Corollary 4.3 to many other Σ protocols (such as the ones based on the DDH assumption or on the N'th residuosity assumption), and obtain (black-box) negative results for the soundness of the resulting protocols obtained by applying the Fiat-Shamir paradigm.

The case of interactive arguments. We note that if the underlying NP language is not sub-exponentially hard, and instead only quasi-polynomially hard (e.g., when $T = \kappa^{\log \kappa}$), then our negative results only hold when the length of the verifier's message is logarithmic in the hardness of the language.

We note, however, that for many Σ-protocols which have the (standard) structure, where the first message sent by the prover is a commitment, the second message sent by the verifier is some query, and the third message sent by the prover is some decommitment information (such as in Blum's protocols presented above), then the following holds: If the protocol Π is for a T-hard language, and if the underlying commitment scheme is a (public-coin) commitment scheme whose binding property can be broken in time T but the hiding property holds against time $\mathrm{poly}(T)$-adversaries, then for any $\ell = \mathrm{poly}(\kappa)$, the soundness of $\Pi^{\mathsf{FS}(\ell)}$ cannot be proven via a black-box reduction to a T-hard falsifiable assumption. This follows from the fact that we can prove that $\Pi^{\mathsf{FS}(\ell)}$ is T-zero-knowledge by constructing a simulator that runs in time T and breaks the binding property of the underlying commitment scheme.

[4] Recall that for a protocol to be special honest-verifier 2^ℓ-zero knowledge, the simulated view needs to be 2^ℓ-indistinguishable from the real view (see Definition 4.1).

4.2 Proof of Theorem 4.2

Theorem 4.2 follows from the following lemma and from Theorem 3.1:

Lemma 4.6. *Let Π be a 4-round public-coin proof or argument system for a $T(\kappa)$-hard language L with the following properties:*

- *The length of the third message, β, satisfies $2^{|\beta|} \leq T$.*
- *Π is special honest verifier (auxiliary input) T-zero knowledge.*

Then Π^{FS} is (auxiliary-input) T-zero-knowledge.

In the remainder of this section we prove Lemma 4.6.

Proof of Lemma 4.6. Let $\mathcal{S}$ be the special honest verifier T-zero-knowledge simulator for Π. Let $\mathcal{V}^*$ be any non-uniform PPT adversary for Π^{FS}. We construct a T-zero-knowledge simulator $\mathcal{S}^{\mathsf{FS}}$ that simulates the view of $\mathcal{V}^*$, as follows:

Simulator $\mathcal{S}^{\mathsf{FS}}$: On input a security parameter 1^κ, a statement $x \in L$ of length $\text{poly}(\kappa)$, and auxiliary input $z \in \{0,1\}^{\text{poly}(\kappa)}$, the simulator $\mathcal{S}^{\mathsf{FS}}$ proceeds as follows:

1. $\mathcal{S}^{\mathsf{FS}}$ begins an internal emulation of Π^{FS} with $\mathcal{V}^*$.
2. Upon receiving (h^{FS}, h) from $\mathcal{V}^*$, $\mathcal{S}^{\mathsf{FS}}$ repeats the following at most $T^2 \geq 2^{2|\beta|}$ times:
   - Sample a transcript $(h, \alpha, \beta, \gamma)$ by choosing β uniformly at random and running $\mathcal{S}(1^\kappa, x, z, h, \beta)$.
   - If $\beta = h^{\mathsf{FS}}(\alpha)$, continue the emulation of Π^{FS} by returning the message (α, β, γ) to $\mathcal{V}^*$, on behalf of the honest prover.
3. If $\mathcal{S}^{\mathsf{FS}}$ does not find a matching transcript after $2^{2|\beta|}$ iterations, it halts and outputs $\perp$.
4. Otherwise, $\mathcal{S}^{\mathsf{FS}}$ halts and outputs $\mathsf{View}_{\mathcal{V}^*}$, the view of $\mathcal{V}^*$.

We claim that for every instance $x \in L$ of length at most $\text{poly}(\kappa)$ with a corresponding witness w, and every auxiliary input $z \in \{0,1\}^{\text{poly}(\kappa)}$, it holds that for every non-uniform distinguisher $D = \{D_\kappa\}$ of size $\text{poly}(T(\kappa))$

$$\left| \Pr[D\left((\mathcal{P}^{\mathsf{FS}}(w), \mathcal{V}^*(z))(1^\kappa, x)\right) = 1] - \Pr[D\left(\mathcal{S}^{\mathsf{FS}}_{\mathcal{V}^*}(1^\kappa, x, z)\right) = 1] \right| \leq \text{negl}(T(\kappa)).$$

This will immediately imply that Π^{FS} is auxiliary-input T-zero-knowledge.

Assume towards contradiction that for infinitely many $\kappa \in \mathbb{N}$ there exists some instance $x \in L$ of length at most $\text{poly}(\kappa)$, some auxiliary input $z \in \{0,1\}^{\text{poly}(\kappa)}$, and some non-uniform distinguisher D of size $\text{poly}(T(\kappa))$ distinguishing the above with advantage $\frac{1}{p(T(\kappa))}$, for some polynomial $p(\cdot)$. We show that for corresponding $\kappa \in \mathbb{N}$, instances $x \in L$ of length at most $\text{poly}(\kappa)$, auxiliary inputs $z \in \{0,1\}^{\text{poly}(\kappa)}$, and for some random tape (h, β), there exists a non-uniform distinguisher D' of size $\text{poly}(T(\kappa))$ such that

$$\left| \Pr[D'\left((\mathcal{P}(w), \mathcal{V}(z, h, \beta))(1^\kappa, x)\right) = 1] - \Pr[D'\left(\mathcal{S}(1^\kappa, x, z, h, \beta)\right) = 1] \right| \geq \frac{1}{p'(T(\kappa))},$$

for some polynomial $p'(\cdot)$. This contradicts the honest verifier T-zero-knowledge property of Π.

Instead of constructing D' as above, we construct a distinguisher D^* that for some $\overline{\mathsf{val}} = ((h_1, \beta_1), \ldots, (h_{T^2}, \beta_{T^2}))$ distinguishes with non-negligible probability $\frac{1}{p^*(T(\kappa))} = \frac{1}{2p(T(\kappa))}$ (for infinitely many κ's) between

$$\mathcal{REAL}(1^\kappa, x, w, z, \overline{\mathsf{val}}) \stackrel{\mathsf{def}}{=} ((\mathcal{P}(w), \mathcal{V}(z, h_1, \beta_1))(1^\kappa, x), \ldots, (\mathcal{P}(w), \mathcal{V}(z, h_{T^2}, \beta_{T^2}))(1^\kappa, x))$$

and

$$\mathcal{SIM}(1^\kappa, x, z, \overline{\mathsf{val}}) \stackrel{\mathsf{def}}{=} (\mathcal{S}(1^\kappa, x, z, h_1, \beta_1), \ldots, \mathcal{S}(1^\kappa, x, z, h_{T^2}, \beta_{T^2}))\,.$$

Since Π is special honest verifier zero-knowledge even against *non-uniform* distinguishers, it is straightforward to show via a standard hybrid argument that the existence of D^* implies the existence of D' as above.

We now proceed to construct D^*, given D as above. Consider the following distinguisher D^*:

D^***:** On input a security parameter 1^κ, a statement $x \in \mathcal{L}$ of size $\mathrm{poly}(\kappa)$, and an auxiliary input $z \in \{0,1\}^{\mathrm{poly}(\kappa)}$, the distinguisher D^* does the following:

1. Sample a random tape r for $\mathcal{V}^*$ and compute $(h^{\mathsf{FS}}, h) \leftarrow \mathcal{V}^*(1^\kappa, x, z; r)$. Additionally, sample $(\beta_1^*, \ldots, \beta_{T^2}^*)$ uniformly at random and set $\overline{\mathsf{val}} = \big((h, \beta_1^*), \ldots, (h, \beta_{T^2}^*)\big)$.
2. Submit $\overline{\mathsf{val}}$ to the external challenger and receive views

   $$(\mathsf{View}_1, \ldots, \mathsf{View}_{T^2}) = \big((h, \alpha_1^*, \beta_1^*, \gamma_1^*), \ldots, (h, \alpha_{T^2}^*, \beta_{T^2}^*, \gamma_{T^2}^*)\big)$$

   drawn from either $\mathcal{REAL}(1^\kappa, x, w, z, \overline{\mathsf{val}})$ or $\mathcal{SIM}(1^\kappa, x, z, \overline{\mathsf{val}})$.
3. Find the first i such that $\beta_i^* = h^{\mathsf{FS}}(\alpha_i^*)$.
4. If there is no such i, then halt and output $\perp$.
5. Otherwise, set the view of $\mathcal{V}^*$ to be $\mathsf{View}_{\mathcal{V}^*} = (\alpha_i^*, \beta_i^*, \gamma_i^*, r)$, and output $D(\mathsf{View}_{\mathcal{V}^*})$.

It is clear by the description above that D^* has size $\mathrm{poly}(T)$.

We next claim that

$$\big|\Pr[D^*(\mathcal{REAL}(1^\kappa, x, w, z, \overline{\mathsf{val}})) = 1] - \Pr[D^*(\mathcal{SIM}(1^\kappa, x, z, \overline{\mathsf{val}})) = 1]\big| \geq \frac{1}{2p(T(\kappa))}.$$

It is straightforward to see that when $(\mathsf{View}_1, \ldots, \mathsf{View}_{T^2})$ are drawn from $\mathcal{SIM}(1^\kappa, x, z, \overline{\mathsf{val}})$, the distribution of $\mathsf{View}_{\mathcal{V}^*}$ generated by D^* is identical to the output distribution of $\mathcal{S}_{\mathcal{V}^*}^{\mathsf{FS}}$.

On the other hand, we show that when $(\mathsf{View}_1, \ldots, \mathsf{View}_{T^2})$ is drawn from $\mathcal{REAL}(1^\kappa, x, w, z, \overline{\mathsf{val}})$ the distribution of $\mathsf{View}_{\mathcal{V}^*}$ generated by D^* is $\frac{1}{2p(T(\kappa))}$-close to the view of $\mathcal{V}^*$ when running $(\mathcal{P}^{\mathsf{FS}}(w), \mathcal{V}^*(z))(1^\kappa, x)$.

Since we have assumed that D distinguishes the output of $\mathcal{S}_{\mathcal{V}^*}^{\mathsf{FS}}$ and $(\mathcal{P}^{\mathsf{FS}}(w), \mathcal{V}^*(z))(1^\kappa, x)$ with advantage at least $\frac{1}{p(T(\kappa))}$, this immediately implies that D^* distinguishes with advantage at least $\frac{1}{2p(T(\kappa))}$.

Note that if we allow D^* to request arbitrarily many views from $(\mathcal{P}(w), \mathcal{V}(z, h, \beta^*))(1^\kappa, x))$, where β^* is chosen independently and uniformly at random, until it obtains a view $\mathsf{View} =$

$(\alpha^*, \beta^*, \gamma^*)$ such that $h(\alpha^*) = \beta^*$, then the view $\mathsf{View}_{\mathcal{V}^*}$ outputted by D^* is identically distributed to the view $\mathsf{View}_{\mathcal{V}^*}$ outputted by $(\mathcal{P}^{\mathsf{FS}}(w), \mathcal{V}^*(z))(1^\kappa, x)$.

Thus, the statistical distance between the distribution over views $\mathsf{View}_{\mathcal{V}^*}$ generated by D^* and the distribution over views $\mathsf{View}_{\mathcal{V}^*}$ generated by $(\mathcal{P}(w), \mathcal{V}(z))(1^\kappa, x)$ is upper bounded by the probability that D^* does not find an i, $1 \leq i \leq T^2$, such that $h(\alpha_i^*) = \beta_i^*$.

Now, note that for $(\mathsf{View}_1, \ldots, \mathsf{View}_{T^2}) = \big((h, \alpha_1^*, \beta_1^*, \gamma_1^*), \ldots, (h, \alpha_{T^2}^*, \beta_{T^2}^*, \gamma_{T^2}^*)\big)$ drawn from $\mathcal{REAL}(1^\kappa, x, w, z, \overline{\mathsf{val}})$ each β_i^* is independent of α_i^*, and is uniformly distributed. Therefore, for any fixed Fiat-Shamir hash h^{FS} and for each i, the event that $h^{\mathsf{FS}}(\alpha_i^*) = \beta_i^*$ is an independent event, and the probability of each such event is exactly $\frac{1}{2^{|\beta|}} \geq \frac{1}{T(\kappa)}$. Thus, the probability that D^* does not find an i, $1 \leq i \leq T^2$, such that $h^{\mathsf{FS}}(\alpha_i^*) = \beta_i^*$, when $(\mathsf{View}_1, \ldots, \mathsf{View}_{T^2})$ is drawn from $\mathcal{REAL}(1^\kappa, x, w, z, \overline{\mathsf{val}})$, is at most

$$\left(1 - \frac{1}{T(\kappa)}\right)^{T^2(\kappa)} \leq \exp(-T(\kappa)) \leq \frac{1}{2p(T(\kappa))}.$$

This concludes the proof of Lemma 4.6. □

5 Separating CS Proofs from Falsifiable Assumptions

In this section we show that for sufficiently hard NP languages, there exist PCPs such that Micali's CS proofs [Mic94] instantiated with such a PCP cannot be proven sound via a black-box reduction to any falsifiable assumption. This section is organized as follows. We first setup notation and recall some relevant definitions in Section 5.1. Then, in Section 5.6, we present our black-box impossibility result.

5.1 Preliminaries

5.2 Hash Functions

Definition 5.1. *Let X and Y be two sets, and let $\mathcal{H}^{\mathsf{univ}}$ be a family of functions from X to Y. We say that $\mathcal{H}^{\mathsf{univ}}$ is a universal family of hash functions if for any two elements $x_1, x_2 \in X$ s.t. $x_1 \neq x_2$, and for any two elements $y_1, y_2 \in Y$,*

$$\Pr_{h \in \mathcal{H}^{\mathsf{univ}}}[h(x_1) = y_1 \wedge h(x_2) = y_2] = \frac{1}{|Y|^2}$$

Definition 5.2. *Let $\{\mathcal{H}_\kappa^{\mathsf{crh}}\}_{\kappa \in \mathbb{N}}$ be a class of functions where every $h \in \mathcal{H}_\kappa^{\mathsf{crh}}$ maps $t(\kappa)$ bits to κ bits. We say that $\{\mathcal{H}_\kappa^{\mathsf{crh}}\}_{\kappa \in \mathbb{N}}$ is a collision-resistant hash (CRH) if for every $\kappa \in \mathbb{N}$, and every $\mathrm{poly}(\kappa)$-size adversary A,*

$$\Pr_{h \leftarrow \mathcal{H}_\kappa^{\mathsf{crh}}}\left[\begin{array}{l} x \neq y \\ h(x) = h(y) \end{array} \;\middle|\; (x, y) \leftarrow A(h)\right] \leq \mathrm{negl}(\kappa)$$

5.3 Statistically Hiding Commitments

Definition 5.3. *A 2-round statistically hiding bit commitment scheme $\Pi_{\mathrm{COM}} = (C, R)$ between a sender $C = (\mathrm{COM}, \mathrm{OPEN})$ and a receiver $R = (R_1, R_2)$ consists of two phases, namely, the* Commit *phase and the* Opening *phase, with syntax:*

COMMIT phase: *On input a security parameter 1^κ and randomness r_R, the receiver R computes the first message $h \leftarrow R_1(1^\kappa; r_R)$ and sends it to the committer C. To commit to a bit b, the committer C runs the commit algorithm* COM *on input (h, b) and randomness r_C to compute $c \leftarrow \text{COM}(1^\kappa, h, b; r_C)$. C sends c to R.*

OPENING phase: *In order to decommit to bit b, the committer C runs the opening algorithm $z = \text{OPEN}(1^\kappa, b, c, r_C)$ and sends its output z to R. The receiver R outputs $b' \leftarrow R_2(1^\kappa, c, z, r_R)$ as the opened bit.*

We require Π_{COM} to satisfy the following two properties:

Statistically Hiding: *The commitment scheme Π_{COM} is said to be $s(\kappa)$-hiding if for all $\kappa \in \mathbb{N}$, the distributions $\mathsf{View}_{\langle C(0),R\rangle}(1^\kappa)$ and $\mathsf{View}_{\langle C(1),R\rangle}(1^\kappa)$ are $s(\kappa)$-close, where $\mathsf{View}_{\langle C(b),R\rangle}(1^\kappa)$ denotes the view of R when interacting with $C(1^\kappa, b)$ (this view simply consists of R's random coins and the messages it receives from C). In particular, when $s(\kappa) = \text{negl}(\kappa)$, then Π_{COM} is said to be statistically hiding.*

Computational Binding: *For all poly-size (cheating senders) C^*,*

$$\Pr\left[\forall b \in \{0,1\},\ R_2(1^\kappa, c, z_b, r_R) = b \;\middle|\; \begin{array}{l} h \leftarrow R_1(1^\kappa; r_R) \\ (c, z_0, z_1) \leftarrow C^*(1^\kappa, h) \end{array}\right] \leq \text{negl}(\kappa),$$

where the probability is over the randomness r_R of the receiver R.

Statistically Hiding Commitments from CRH. We briefly recall the statistically hiding bit commitment scheme $\Pi_{\text{COM}} = (C, R)$ of Halevi and Micali [HM96].

Let $\{\mathcal{H}_\kappa^{\mathsf{crh}}\}_{\kappa\in\mathbb{N}}$ be a CRH where every $h \in \mathcal{H}_\kappa^{\mathsf{crh}}$ maps $t(\kappa)$ bits to κ bits. Let $\mathcal{H}^{\mathsf{univ}}$ be a universal family of hash functions where every $h^{\mathsf{univ}} \in \mathcal{H}^{\mathsf{univ}}$ maps $t(\kappa)$ bits to 1-bit. The commitment scheme Π_{COM} is described as follows. The receiver R's message is a uniformly chosen $h \leftarrow \mathcal{H}_\kappa^{\mathsf{crh}}$. To commit to a bit b, the commitment algorithm COM generates a uniformly chosen $h^{\mathsf{univ}} \leftarrow \mathcal{H}^{\mathsf{univ}}$ and a random string z of length $t(\kappa)$ s.t. $h^{\mathsf{univ}}(z) = b$. It then computes $y = h(z)$ and outputs $c = (y, h^{\mathsf{univ}})$. Finally, to decommit to b, the opening algorithm OPEN simply outputs z.

We now recall the following theorem from [HM96]:

Theorem 5.4 (Halevi-Micali [HM96])**.** *For every $\kappa \in \mathbb{N}$, the bit commitment scheme Π_{COM} is $2^{-(t(\kappa)-\kappa-6)/3}$ hiding.*

5.4 Merkle Trees

Merkle tree hashing [Mer89] is a technique that allows the computation of a succinct commitment to a long string π and a succinct decommitment to any bit of π. In what follows, we give a brief informal description of the Merkle Tree hashing algorithm.

Let $h \leftarrow \mathcal{H}_\kappa^{\mathsf{crh}}$ be a randomly chosen hash function that maps $t(\kappa)$ bits to κ bits. Then, the commitment and opening algorithms are described as follows:

$\text{COM}_{\text{MT}}(1^\kappa, \pi, h)$: Divide π into $p = |\pi|/t(\kappa)$ parts, each of length $t(\kappa)$, and evaluate the hash function h over each part. Repeat the operation on the resultant string, and so on, until a single κ-bit string c is obtained. Output (c, d) as the succinct commitment, where d is the depth of the resulting tree.

$\text{OPEN}_{\text{MT}}(1^\kappa, \pi, h, i)$: In order to decommit to the i^{th} bit π_i of the string π, output π_i along with all the hash values on the path, path_i, from the leaf π_i to the root c. In addition, output all the "necessary information" to verify these hash values. This information consists of $t(\kappa)$ bits for each level $j \in [d]$, such that when applying the hash h to this $t(\kappa)$-bit string we get the j'th hash value in path_i. Note that all-in-all, the output is of length $t \cdot d$. From now on we abuse notation, and let $\rho_i = (\pi_i, \mathsf{path}_i)$ denote the entire output. (We remark that we have explicitly added π_i to the opening for clarity of exposition.)

Informally speaking, we say that an opening $\rho_i = (\pi_i, \mathsf{path}_i)$ is *valid* if one can compute the (committed) root value c from path_i by repeated evaluations of the hash h.

5.5 Probabilistically Checkable Proofs

Definition 5.5 (ℓ-query PCP)**.** *An $\ell = \ell(\kappa)$-query efficient probabilistically checkable proof (PCP) for an* NP *language L with corresponding relation R is a tuple of efficient algorithms $\Pi_{\mathsf{pcp}} = (\mathcal{P}_{\mathsf{pcp}}, \mathcal{V}_{\mathsf{pcp}} = (\mathcal{V}^1_{\mathsf{pcp}}, \mathcal{V}^2_{\mathsf{pcp}}))$ with syntax:*

- $\pi \leftarrow \mathcal{P}_{\mathsf{pcp}}(1^\kappa, x, w)$*: On input a security parameter 1^κ, and an instance x, together with a corresponding witness w, the prover $\mathcal{P}_{\mathsf{pcp}}$ outputs a proof π. This algorithm may be randomized.*
- $0/1 \leftarrow \mathcal{V}^\pi_{\mathsf{pcp}}(1^\kappa, x; r)$*: The verifier first uses randomness r to generate an ℓ-size query set $q \leftarrow \mathcal{V}^1_{\mathsf{pcp}}(1^\kappa, x, r)$ denoted by $q = \{q_1, \ldots, q_\ell\}$, and then queries the proof π at the locations $q_1, \ldots, q_\ell$ to obtain ℓ-sized string $\pi_q = (\pi_{q_1}, \ldots, \pi_{q_\ell})$, where π_{q_i} represents the q_i^{th} bit of π. It then outputs $\mathcal{V}^2_{\mathsf{pcp}}(1^\kappa, x, q, \pi_q)$ as its decision.*

We require Π_{pcp} to satisfy the following properties:

Completeness: *For all $\kappa \in \mathbb{N}$ and $(x, w) \in R$,*

$$\Pr\left[\mathcal{V}^\pi_{\mathsf{pcp}}(1^\kappa, x) = 1 \middle| \pi \leftarrow \mathcal{P}_{\mathsf{pcp}}(1^\kappa, x, w)\right] \geq 1 - \text{negl}(\kappa),$$

where the probability is over the randomness of the prover $\mathcal{P}_{\mathsf{pcp}}$ and verifier $\mathcal{V}_{\mathsf{pcp}}$.

Soundness: *For all $\kappa \in \mathbb{N}$ and any (unbounded) cheating prover $\mathcal{P}^*_{\mathsf{pcp}}$,*

$$\Pr\left[\mathcal{V}^{\bar{\pi}}_{\mathsf{pcp}}(1^\kappa, \bar{x}) = 1 \ \wedge \ \bar{x} \notin L \,\middle|\, (\bar{x}, \bar{\pi}) \leftarrow \mathcal{P}^*_{\mathsf{pcp}}(1^\kappa)\right] \leq \text{negl}(\kappa),$$

where the probability is over the randomness of the verifier $\mathcal{V}_{\mathsf{pcp}}$.

Remark 5.6 (Verifier's random coins)**.** *We assume, without loss of generality, that the number of random bits used by the query generation algorithm $\mathcal{V}^1_{\mathsf{pcp}}$ of the PCP verifier, is at most the number of queries $\ell(\kappa)$.*

Definition 5.7 (T-Zero Knowledge PCP)**.** *A PCP $\Pi_{\mathsf{pcp}} = (\mathcal{P}_{\mathsf{pcp}}, \mathcal{V}_{\mathsf{pcp}} = (\mathcal{V}^1_{\mathsf{pcp}}, \mathcal{V}^2_{\mathsf{pcp}}))$ for an* NP *language L is said to be $T = T(\kappa)$ zero-knowledge if there exists a simulator $\mathcal{S}_{\mathsf{pcp}}$ of size at most $\text{poly}(T(\kappa))$ such that for every $\kappa \in \mathbb{N}$, every instance $x \in L$ of length at most $\text{poly}(\kappa)$, with a corresponding witness w, every randomness r of the verifier producing a query $q \leftarrow \mathcal{V}^1_{\mathsf{pcp}}(1^\kappa, x, r)$, we have that $(r, \pi^{\mathcal{S}}_q)$ and (r, π_q) are $T(\kappa)$-indistinguishable, where $\pi^{\mathcal{S}}_q \leftarrow \mathcal{S}_{\mathsf{pcp}}(1^\kappa, x, r)$ and π_q is the answer to query q determined from $\pi \leftarrow \mathcal{P}_{\mathsf{pcp}}(1^\kappa, x, w)$.*

We note that the work of Dwork et al. [DFK+92] gives the construction of perfect zero-knowledge PCPs (as per the above definition) for NP with negligible soundness error when $\ell = \omega(\log \kappa)$.

5.6 CS Proofs

In this section we show that for sufficiently hard NP languages, there exist PCPs such that Micali's CS proof, instantiated with such PCPs, cannot be proven sound via a black-box reduction to any falsifiable assumption.

Since Micali's CS proofs are obtained by applying the Fiat-Shamir heuristic on Kilian's 4-round succinct argument [Kil92], we start by recalling Kilian's protocol, denoted by $\Pi = (\mathcal{P}, \mathcal{V})$.

Kilian's protocol. We first define some notation. Let $\mathcal{H}_\kappa^{\mathsf{crh}}$ be a CRH where each $h \in \mathcal{H}_\kappa^{\mathsf{crh}}$ maps $t(\kappa)$ bits to κ bits. Let $(\textsc{com}_{\textsc{mt}}, \textsc{open}_{\textsc{mt}})$ denote the Merkle Tree hash commitment and opening algorithms. For any NP language L, let $\Pi_{\mathsf{pcp}} = (\mathcal{P}_{\mathsf{pcp}}, \mathcal{V}_{\mathsf{pcp}} = (\mathcal{V}^1_{\mathsf{pcp}}, \mathcal{V}^2_{\mathsf{pcp}}))$ be an $\ell(\kappa)$-query PCP. Kilian's 4-round protocol Π is given in Figure 5.1.

Parameters: Security parameter κ.

Prover's Input: Statement x and a witness w for $x \in L$.

Verifier's Input: Statement x.

$\mathcal{V} \to \mathcal{P}$: $\mathcal{V}$ chooses $h \leftarrow \mathcal{H}_\kappa^{\mathsf{crh}}$ at random and sends h to $\mathcal{P}$.

$\mathcal{P} \to \mathcal{V}$: $\mathcal{P}$ performs the following steps:

- Run the PCP prover algorithm $\mathcal{P}_{\mathsf{pcp}}$ on input (x, w) (and uniformly chosen randomness, if necessary) to compute the proof string $\pi \leftarrow \mathcal{P}_{\mathsf{pcp}}(1^\kappa, x, w)$.
- Run the Merkle Tree hash commitment algorithm $\textsc{com}_{\textsc{mt}}$ on input (π, h) to compute the succinct commitment $c \leftarrow \textsc{com}_{\textsc{mt}}(1^\kappa, \pi, h)$.

$\mathcal{P}$ sends c to $\mathcal{V}$.

$\mathcal{V} \to \mathcal{P}$: $\mathcal{V}$ chooses random coins r for the PCP verifier $\mathcal{V}_{\mathsf{pcp}}$ and sends r to $\mathcal{P}$.

$\mathcal{P} \to \mathcal{V}$: $\mathcal{P}$ performs the following steps:

- Compute the PCP verifier $\mathcal{V}_{\mathsf{pcp}}$'s query set $q = \{q_1 \ldots, q_\ell\} \leftarrow \mathcal{V}^1_{\mathsf{pcp}}(1^\kappa, x, r)$.
- Run the Merkle Tree hash opening algorithm $\textsc{open}_{\textsc{mt}}$ on inputs $\{q_i\}_{i=1}^\ell$ to compute $\{\rho_i = \textsc{open}_{\textsc{mt}}(1^\kappa, \pi, h, q_i)\}_{i=1}^\ell$. Let $\rho = (\rho_1, \ldots, \rho_\ell)$.

$\mathcal{P}$ sends ρ to $\mathcal{V}$.

Verification: Let $\rho_i = (\pi_{q_i}, \mathsf{path}_{q_i})$. $\mathcal{V}$ accepts iff:

- $\forall i \in [\ell]$, $(\pi_{q_i}, \mathsf{path}_{q_i})$ is *valid* with respect to $q = \mathcal{V}^1_{\mathsf{pcp}}(1^\kappa, r)$.
- $\mathcal{V}^2_{\mathsf{pcp}}(1^\kappa, x, q, \pi_q) = 1$, where $\pi_q = (\pi_{q_1}, \ldots, \pi_{q_\ell})$.

Figure 5.1: Kilian's 4-round succinct argument $\Pi = (\mathcal{P}, \mathcal{V})$

(In)security of CS Proofs. Let Π^{FS} denote Micali's 2-message CS proof system obtained by applying the Fiat-Shamir transformation to Kilian's protocol Π using $h^{\mathsf{FS}} \leftarrow \mathcal{H}^{\mathsf{FS}}_{\kappa}$. For any NP language L and any PCP, Π_{pcp}, for L, Micali proved that Π is sound in the so called *random oracle model*, where h^{FS} is modeled as a random oracle. We now prove that for every $2^{\ell(\kappa)}$-hard language L, there exists an ℓ-query PCP such that the CS proof Π^{FS} for language L cannot be proven sound via a black-box reduction to any falsifiable assumption. More formally,

Theorem 5.8. *For all $\ell = \ell(\kappa)$ and any $2^{\ell(\kappa)}$-hard language L, there exists an ℓ-query PCP Π_{pcp} such that the soundness of CS proof Π^{FS} instantiated with Π_{pcp} for language L cannot be proven via a black-box reduction to a $2^{\ell(\kappa)}$-hard falsifiable assumption (unless the assumption is false).*

The following corollary follows easily from Theorem 5.8, similarly to the way that Corollary 4.3 follows from Theorem 4.2.

Corollary 5.9. *For any sub-exponentially hard language L and for any $\ell = \mathrm{poly}(\kappa)$, there exists an ℓ-query PCP Π_{pcp} such that the soundness of CS proof Π^{FS} instantiated with Π_{pcp} for language L cannot be proven via a black-box reduction to a $2^{\ell(\kappa)}$-hard falsifiable assumption (unless the assumption is false).*

We give a high-level overview of the proof of Theorem 5.8. Let L be a $2^{\ell(\kappa)}$-hard language. Our main idea is to show that when Kilian's 4-round succinct argument Π is instantiated with a specific PCP (with some zero-knowledge properties), then it is a (special) honest verifier $2^{\ell(\kappa)}$-zero knowledge argument for L, where the verifier's second message is of length at most ℓ (see Remark 5.6). This, when combined with Theorem 4.2 immediately yields the proof of Theorem 5.8.

A natural first idea towards showing that Kilian's 4-round protocol Π is (special) honest verifier $2^{\ell(\kappa)}$-zero knowledge is to simply instantiate it with a zero-knowledge PCP (see Definition 5.7). Unfortunately, this is not sufficient because zero-knowledge PCPs only guarantee that for a given query set q, the proof substring π_q can be simulated. In Kilian's protocol, however, the verifier also obtains some information about the proof substring $\pi_{\bar{q}}$, where $\bar{q}$ is the set consisting of each $i \in [|\pi|] \setminus q$. Thus, we need to ensure that the "extra" information that is given about $\pi_{\bar{q}}$ in Kilian's protocol Π can also be simulated.

We achieve this by making use of statistically hiding commitments. Specifically, given any zero-knowledge PCP proof string $\tilde{\pi}$, we encode it into a new, larger PCP proof string, π, such that the hash values of π statistically hide the bits of $\tilde{\pi}$. More details are given in the formal proof of Theorem 5.8 presented in Section 5.7.

5.7 Proof of Theorem 5.8

Let L be any $2^{\ell(\kappa)}$-hard NP language. Let $\Pi^{\mathsf{zk}}_{\mathsf{pcp}} = \left(\tilde{\mathcal{P}}_{\mathsf{pcp}}, \tilde{\mathcal{V}}_{\mathsf{pcp}} = (\tilde{\mathcal{V}}^1_{\mathsf{pcp}}, \tilde{\mathcal{V}}^2_{\mathsf{pcp}})\right)$ be a $2^{\ell(\kappa)}$ zero-knowledge PCP for L. We first construct a PCP $\Pi_{\mathsf{pcp}} = \left(\mathcal{P}_{\mathsf{pcp}}, \mathcal{V}_{\mathsf{pcp}} = (\mathcal{V}^1_{\mathsf{pcp}}, \mathcal{V}^2_{\mathsf{pcp}})\right)$ as follows:

Prover $\mathcal{P}_{\mathsf{pcp}}(1^\kappa, x, w; r_p)$: Run the algorithm $\tilde{\mathcal{P}}_{\mathsf{pcp}}$ on input (x, w) (and randomness from r_p, if necessary) to compute $\tilde{\pi} \leftarrow \tilde{\mathcal{P}}_{\mathsf{pcp}}(1^\kappa, x, w)$. Randomly choose a universal hash function $h^{\mathsf{univ}} \leftarrow \mathcal{H}^{\mathsf{univ}}$ that maps $t(\kappa)$ bits to a single bit. For every $i \in [|\tilde{\pi}|]$, choose $z_i \leftarrow \{0,1\}^{t(\kappa)}$ at random s.t. $h^{\mathsf{univ}}(z_i) = \tilde{\pi}_i$. Output $\pi = (z_1, \ldots, z_{|\tilde{\pi}|}, h^{\mathsf{univ}})$.

Verifier $\mathcal{V}^{\pi}_{\mathsf{pcp}}(1^\kappa, x; r_v)$: First run the algorithm $\mathcal{V}^1_{\mathsf{pcp}}$ with randomness r_v, described as follows:

- $q \leftarrow \mathcal{V}^1_{\mathsf{pcp}}(1^\kappa, x, r_v)$: Run the algorithm $\tilde{\mathcal{V}}^1_{\mathsf{pcp}}$ with randomness r_v to compute the $\tilde{\ell}$-sized query set $\tilde{q} \leftarrow \tilde{\mathcal{V}}^1_{\mathsf{pcp}}(1^\kappa, x, r_v)$ denoted by $\tilde{q} = \{\tilde{q}_1, \ldots, \tilde{q}_{\tilde{\ell}}\}$. For every $i \in [\tilde{\ell}]$, let u_i be the $t(\kappa)$-size set of locations $\{t(\kappa)(\tilde{q}_i - 1) + 1, \ldots, t(\kappa)\tilde{q}_i\}$. Further, let v be the $|h^{\mathsf{univ}}|$-sized set of locations corresponding to the last $|h^{\mathsf{univ}}|$ bits of π. Output query set $q = \{u_1, \ldots, u_{\tilde{\ell}}, v\}$. (Note that by opening all the sets u_i and v, we have that $q = \{q_1, \ldots, q_\ell\}$, where $\ell = t(\kappa)\tilde{\ell} + |h^{\mathsf{univ}}|$.)

Now, query the proof π at all the locations in the query set q to obtain string π_q, and run the algorithm $\mathcal{V}^2_{\mathsf{pcp}}$ with input π_q, described as follows:

- $0/1 \leftarrow \mathcal{V}^2_{\mathsf{pcp}}(1^\kappa, x, \pi_q)$: Parse π_q as $(z_{u_1}, \ldots, z_{u_{\tilde{\ell}}}, h^{\mathsf{univ}})$, where for every $i \in [\tilde{\ell}]$, z_{u_i} is the $t(\kappa)$-sized substring of π corresponding to the locations u_i, and h^{univ} is the substring corresponding to the locations v. For every $i \in [\tilde{\ell}]$, compute $\tilde{\pi}_{\tilde{q}_i} = h^{\mathsf{univ}}(z_{u_i})$. Let $\tilde{\pi}_{\tilde{q}} = (\tilde{\pi}_{\tilde{q}_1}, \ldots, \tilde{\pi}_{\tilde{q}_{\tilde{\ell}}})$. Output $\tilde{\mathcal{V}}^2_{\mathsf{pcp}}(1^\kappa, x, \tilde{\pi}_{\tilde{q}})$.

This completes the description of the PCP Π_{pcp}. The completeness and soundness properties of Π_{pcp} are straightforward.

Consider Kilian's 4-round argument system $\Pi = (\mathcal{P}, \mathcal{V})$ for language L instantiated with the PCP Π_{pcp}, as described above. We will show that Π is $2^{\ell(\kappa)}$-HVZK, which when combined with Theorem 4.2 yields Theorem 5.8. We first construct a simulator $\mathcal{S}$ for Π as follows:

Simulator $\mathcal{S}$. On input a hash function $h \leftarrow \mathcal{H}^{\mathsf{crh}}_\kappa$ and randomness r, the simulator $\mathcal{S}$ works in the following manner:

- Run $\mathcal{V}^1_{\mathsf{pcp}}(1^\kappa, r)$ to compute the corresponding query set $\tilde{q} = \{\tilde{q}_1, \ldots, \tilde{q}_{\tilde{\ell}}\}$ of $\tilde{\mathcal{V}}^1_{\mathsf{pcp}}(1^\kappa, r)$, and the more inclusive query set $q = \{q_1, \ldots, q_\ell\}$ of $\mathcal{V}^1_{\mathsf{pcp}}(1^\kappa, r)$.
- Run the simulator $\mathcal{S}_{\mathsf{pcp}}$ for the PCP $\Pi^{\mathsf{zk}}_{\mathsf{pcp}}$ on input $(x, r, \tilde{q})$ to obtain $\tilde{\pi}_{\tilde{q}} = (\tilde{\pi}_{\tilde{q}_1}, \ldots, \tilde{\pi}_{\tilde{q}_{\tilde{\ell}}})$. Further, for every $j \in [|\tilde{\pi}|] \setminus \tilde{q}$ (where $|\tilde{\pi}|$ denotes the size of the honest proof in $\Pi^{\mathsf{zk}}_{\mathsf{pcp}}$), choose a random bit $\tilde{\pi}_j$. Let $\tilde{\pi} = (\tilde{\pi}_1, \ldots, \tilde{\pi}_{|\tilde{\pi}|})$.
- Randomly choose a universal hash function $h^{\mathsf{univ}} \leftarrow \mathcal{H}^{\mathsf{univ}}$ that maps $t(\kappa)$ bits to a single bit. Then, for every $i \in [|\tilde{\pi}|]$, choose $z_i \leftarrow \{0,1\}^{t(\kappa)}$ at random s.t. $h^{\mathsf{univ}}(z_i) = \tilde{\pi}_i$. Let $\pi = (z_1, \ldots, z_{|\tilde{\pi}|}, h^{\mathsf{univ}})$.
- Run the Merkle Tree hash commitment algorithm $\textsc{com}_{\textsc{mt}}$ on input (π, h) to compute the succinct commitment $c^{\mathcal{S}} \leftarrow \textsc{com}_{\textsc{mt}}(1^\kappa, \pi, h)$.
- Run the Merkle Tree hash opening algorithm $\textsc{open}_{\textsc{mt}}$ on inputs $\{q_i\}_{i=1}^{\ell}$ to compute openings $\{\rho_i = \textsc{open}_{\textsc{mt}}(1^\kappa, \pi, h, q_i)\}_{i=1}^{\ell}$. Let $\rho^{\mathcal{S}} = (\rho_1, \ldots, \rho_\ell)$.
- Output $(c^{\mathcal{S}}, \rho^{\mathcal{S}})$.

We prove that for all $\kappa \in \mathbb{N}$, every instance $x \in L$ of length at most $\text{poly}(\kappa)$ with a corresponding witness w, every $h \in \mathcal{H}^{\mathsf{crh}}_\kappa$, every randomness r of $\mathcal{V}_{\mathsf{pcp}}$,

$$(h, c^{\mathcal{S}}, r, \rho^{\mathcal{S}}) \overset{T}{\approx} (h, c, r, \rho) \tag{5.1}$$

where $(c^{\mathcal{S}}, \rho^{\mathcal{S}}) \leftarrow \mathcal{S}(1^\kappa, x, h, r)$ and (h, c, r, ρ) denotes a protocol transcript between $\mathcal{P}(1^\kappa, x, w)$ and $\mathcal{V}(1^\kappa, x)$ with (h, r) as $\mathcal{V}$'s messages. We prove Equation (5.1) via a simple hybrid argument.

Hybrid H_0: This experiment corresponds to the honest protocol execution for statement x, where the verifier's messages are h and r. The output of the experiment is the protocol transcript (h, c, r, ρ).

Hybrid H_1: This experiment is the same as H_0 except that the proof string $\tilde{\pi}$ of the underlying ZK PCP $\Pi^{\mathsf{zk}}_{\mathsf{pcp}}$ is computed in the following manner:

- First run the honest prover algorithm $\tilde{\mathcal{P}}_{\mathsf{pcp}}$ on input (x, w) to compute proof string $\tilde{\pi}$.
- Let $\tilde{q} = \{\tilde{q}_1, \ldots, \tilde{q}_{\tilde{\ell}}\} \leftarrow \tilde{\mathcal{V}}^1_{\mathsf{pcp}}(1^\kappa, r)$. Then, for every $j \in [|\tilde{\pi}|] \setminus \tilde{q}$, choose a random bit b_j and set $\tilde{\pi}_j = b_j$. Let the proof string be $\tilde{\pi} = (\tilde{\pi}_1, \ldots, \tilde{\pi}_{|\tilde{\pi}|})$.

Hybrid H_2: This experiment is the same as H_1 except that instead of computing the proof string $\tilde{\pi}$ of the underlying ZK PCP $\Pi^{\mathsf{zk}}_{\mathsf{pcp}}$ as before, we do the following: Run the simulator $\mathcal{S}_{\mathsf{pcp}}$ on input $(x, r, \tilde{q})$ (where $\tilde{q} \leftarrow \tilde{\mathcal{V}}^1_{\mathsf{pcp}}(1^\kappa, r)$) to compute $\tilde{\pi}^{\mathcal{S}}_{\tilde{q}} = (\tilde{\pi}_{\tilde{q}_1}, \ldots, \tilde{\pi}_{\tilde{q}_{\tilde{\ell}}})$. For every $j \in [|\tilde{\pi}|] \setminus \tilde{q}$, we let $\tilde{\pi}_j$ be a random bit as before. Let the proof string be $\tilde{\pi} = (\tilde{\pi}_1, \ldots, \tilde{\pi}_{|\tilde{\pi}|})$.

Note that this experiment corresponds to our zero knowledge simulator $\mathcal{S}$ for Π.

This completes the description of the hybrid experiments. In order to conclude our proof, we now argue that H_0 is $2^{\ell(\kappa)}$-indistinguishable from H_2.

Indistinguishability of H_0 and H_1. Note that the only difference between H_0 and H_1 is that in H_1, for every $j \in [|\tilde{\pi}|] \setminus \tilde{q}$, the bit $\tilde{\pi}_j$ is randomly chosen (instead of being honestly computed by $\tilde{\mathcal{P}}_{\mathsf{pcp}}$). We argue that despite this difference, H_0 and H_1 are $2^{\ell(\kappa)}$-indistinguishable.

Consider the Merkle Tree hash computed over the proof string π. Let the leaves in the tree correspond to depth 0, and consider the κ-bit hash values at depth 1 in the tree. We denote the set of these hash values by level_1. It follows from our construction of the PCP proof string π and from Theorem 5.4 that $\forall i \in [|\tilde{\pi}|]$, the i^{th} hash value in level_1 (along with h^{univ}) is a $2^{-(t(\kappa)-\kappa-6)/3}$ hiding commitment of the bit $\tilde{\pi}_i$.

Now, let us consider the modified experiments H'_0 and H'_1 that are the same as H_0 and H_1 respectively, except that their outputs are modified in the following manner. Instead of outputting (h, c, r, ρ), they output $(h, r, \pi_q, \mathsf{level}_1)$, where (as defined earlier) π_q is the PCP proof substring corresponding to the query set $q \leftarrow \mathcal{V}^1_{\mathsf{pcp}}(1^\kappa, x, r)$, and as before, level_1 is the set of all κ-bit hash values at level 1 of the Merkle Tree hash. Clearly, the outputs of H'_0 and H'_1 carry no less information than H_0 and H_1, and therefore it suffices to argue $2^{\ell(\kappa)}$-indistinguishability of H'_0 and H'_1 in order to argue that H_0 and H_1 are $2^{\ell(\kappa)}$-indistinguishable.

Note that the only difference between H'_0 and H'_1 is the manner in which the commitments in level_1 are computed. In particular, in H'_0, the i'th hash value is a commitment to the bit $\tilde{\pi}_i$, while in H'_1, each hash value is a commitment to a random bit. If the commitment scheme described in Section 5.3 is $2^{-\ell(\kappa)}$-hiding, then by a standard hybrid argument, H'_0 and H'_1 are $2^{\ell(\kappa)}$-indistinguishable (see Definition 2.1). Thus, by setting $t(\kappa) = 3\ell(\kappa) + \kappa + 6$ and invoking Theorem 5.4, we achieve the desired result.

Indistinguishability of H_1 and H_2. Note that the only difference between H_1 and H_2 is the manner in which the proof substring $\tilde{\pi}_{\tilde{q}_1}, \ldots, \tilde{\pi}_{\tilde{q}_{\tilde{\ell}}}$ is computed. Specifically, it is computed by the honest PCP prover algorithm $\tilde{\mathcal{P}}_{\mathsf{pcp}}$ in H_1, while in H_2, it is computed by the simulator $\mathcal{S}_{\mathsf{pcp}}$. Thus, the $2^{\ell(\kappa)}$-indistinguishability of H_1 and H_2 immediately follows from the $2^{\ell(\kappa)}$-zero knowledge property of $\Pi^{\mathsf{zk}}_{\mathsf{pcp}}$.

This completes the proof of Theorem 5.8.

References

[AJL+12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold fhe. In *EUROCRYPT*, pages 483–501, 2012.

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, pages 106–115, 2001.

[BDNP08] Assaf Ben-David, Noam Nisan, and Benny Pinkas. Fairplaymp: a system for secure multi-party computation. In *ACM Conference on Computer and Communications Security*, pages 257–266, 2008.

[BDSK+13] Nir Bitansky, Dana Dachman-Soled, Yael Tauman Kalai, Sanjam Garg, Abhishek Jain, Adriana López-Alt, and Daniel Wichs. Why "fiat-shamir for proofs" lacks a proof. In *TCC*, 2013.

[BGW12] Nir Bitansky, Sanjam Garg, and Daniel Wichs. Why "fiat-shamir for proofs" lacks a proof. Cryptology ePrint Archive, Report 2012, 2012. `http://eprint.iacr.org/`.

[Blu81] Manuel Blum. Coin flipping by telephone. In *CRYPTO*, pages 11–15, 1981.

[Blu87] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, pages 1444–1451, 1987.

[BLV03] Boaz Barak, Yehuda Lindell, and Salil P. Vadhan. Lower bounds for non-black-box zero knowledge. In *FOCS*, pages 384–393, 2003.

[BMG07] Boaz Barak and Mohammad Mahmoody-Ghidary. Lower bounds on signatures from symmetric primitives. In *FOCS*, pages 680–688, 2007.

[BMG09] Boaz Barak and Mohammad Mahmoody-Ghidary. Merkle puzzles are optimal - an $o(n^2)$-query attack on any key exchange from a random oracle. In *CRYPTO*, pages 374–390, 2009.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[CD09] Ronald Cramer and Ivan Damgård. On the amortized complexity of zero-knowledge protocols. In *CRYPTO*, pages 177–191, 2009.

[Cor02] Jean-Sébastien Coron. Optimal security proofs for pss and other signature schemes. In *EUROCRYPT*, pages 272–287, 2002.

[DFK+92] Cynthia Dwork, Uriel Feige, Joe Kilian, Moni Naor, and Shmuel Safra. Low communication 2-prover zero-knowledge proofs for np. In *CRYPTO*, pages 215–227, 1992.

[DHT12] Yevgeniy Dodis, Iftach Haitner, and Aris Tentes. On the instantiability of hash-and-sign rsa signatures. In *TCC*, pages 112–132, 2012.

[DNRS99] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *FOCS*, pages 523–534, 1999.

[DOPS04] Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im)possibility of cryptography with imperfect randomness. In *FOCS*, pages 196–205, 2004.

[DRV12] Yevgeniy Dodis, Thomas Ristenpart, and Salil P. Vadhan. Randomness condensers for efficiently samplable, seed-dependent sources. In *TCC*, pages 618–635, 2012.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the fiat-shamir paradigm. In *FOCS*, pages 102–113, 2003.

[GKM+00] Yael Gertner, Sampath Kannan, Tal Malkin, Omer Reingold, and Mahesh Viswanathan. The relationship between public key encryption and oblivious transfer. In *FOCS*, pages 325–335, 2000.

[GMR01] Yael Gertner, Tal Malkin, and Omer Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *FOCS*, pages 126–135, 2001.

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

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *STOC*, pages 99–108, 2011.

[HH09] Iftach Haitner and Thomas Holenstein. On the (im)possibility of key dependent encryption. In *TCC*, pages 202–219, 2009.

[HM96] Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In *CRYPTO*, pages 201–215, 1996.

[HT98] Satoshi Hada and Toshiaki Tanaka. On the existence of 3-round zero-knowledge protocols. In *CRYPTO*, pages 408–423, 1998.

[IR89] Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *STOC*, pages 44–61, 1989.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *STOC*, pages 723–732, 1992.

[Mer89] Ralph C. Merkle. A certified digital signature. In *CRYPTO*, pages 218–238, 1989.

[Mic94] Silvio Micali. CS proofs. In *FOCS*, pages 436–453, 1994.

[MNPS04] Dahlia Malkhi, Noam Nisan, Benny Pinkas, and Yaron Sella. Fairplay - secure two-party computation system. In *USENIX Security Symposium*, pages 287–302, 2004.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In *CRYPTO*, pages 96–109, 2003.

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In *EUROCRYPT*, pages 160–176, 2003.

[Pas11] Rafael Pass. Limits of provable security from standard assumptions. In *STOC*, pages 109–118, 2011.

[PS96] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In *EUROCRYPT*, pages 387–398, 1996.

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In *TCC*, pages 1–20, 2004.

[Sim98] Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *EUROCRYPT*, pages 334–345, 1998.