SNARGs for Bounded Depth Computations and PPAD Hardness from Sub-Exponential LWE

Ruta Jawale
UIUC
USA
jawale2@illinois.edu

Yael Tauman Kalai
Microsoft Research, MIT
USA
yael@microsoft.com

Dakshita Khurana
UIUC
USA
dakshita@illinois.edu

Rachel Zhang
MIT
USA
rachelyz@mit.edu

ABSTRACT

We construct a succinct non-interactive publicly-verifiable delegation scheme for any log-space uniform circuit under the sub-exponential Learning With Errors (LWE) assumption. For a circuit $C : \{0,1\}^N \rightarrow \{0,1\}$ of size S and depth D, the prover runs in time $\mathsf{poly}(S)$, the communication complexity is $D \cdot \mathsf{polylog}(S)$, and the verifier runs in time $(D + N) \cdot \mathsf{polylog}(S)$. To obtain this result, we introduce a new cryptographic primitive: *a lossy correlation-intractable hash function family*. We use this primitive to soundly instantiate the Fiat-Shamir transform for a large class of interactive proofs, including the interactive sum-check protocol and the GKR protocol, assuming the sub-exponential hardness of LWE.

Additionally, by relying on the result of Choudhuri et al. (STOC 2019), we establish (sub-exponential) average-case hardness of PPAD, assuming the sub-exponential hardness of LWE.

CCS CONCEPTS

• **Theory of computation** → **Cryptographic primitives**; **Cryptographic protocols**; **Interactive proof systems**.

KEYWORDS

Fiat-Shamir heuristic, cryptographic protocols, delegation of computation, PPAD hardness

ACM Reference Format:
Ruta Jawale, Yael Tauman Kalai, Dakshita Khurana, and Rachel Zhang. 2021. SNARGs for Bounded Depth Computations and PPAD Hardness from Sub-Exponential LWE. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC '21), June 21–25, 2021, Virtual, Italy*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3406325.3451055

STOC '21, June 21–25, 2021, Virtual, Italy

ACM ISBN 978-1-4503-8053-9/21/06…$15.00
https://doi.org/10.1145/3406325.3451055

1 INTRODUCTION

In the past decade, significant efforts have been directed towards constructing succinct and efficiently verifiable proof systems. The key question in this area is the following: can one cryptographically generate a short certificate for the correctness of a long computation? Efforts to build succinct and efficiently verifiable certificates have been motivated by the increasing popularity of cloud services and blockchain technologies.

This task of constructing succinct proofs for long computations is information theoretically impossible. But there has been a long line of work on succinct proofs that achieve computational soundness against cheating provers by relying on cryptographic hardness assumptions. Such computationally sound proofs are referred to as *arguments* [20].

In this work we focus on building *publicly verifiable, succinct non-interactive arguments* (SNARGs)[1] from standard cryptographic assumptions. A SNARG enables a prover to generate a succinct, efficiently verifiable certificate attesting to the validity of relatively inefficient computation. As in almost all prior work, we consider the CRS model, which assumes the existence of a common reference string (CRS) known to all parties. Indeed, without a CRS it is impossible to obtain SNARGs with soundness against non-uniform provers. Despite extensive work in this area (which we discuss in Section 1.2), there were previously no known constructions of SNARGs in the CRS model for any interesting class of languages, under standard well-studied cryptographic assumptions.

Our first result is a SNARG for bounded depth deterministic computations under the sub-exponential hardness of learning with errors (LWE), a standard cryptographic assumption. We achieve this result by provably instantiating (under LWE) the Fiat-Shamir paradigm [36] applied to an interactive succinct proof, in particular, to the GKR protocol [41] for delegating bounded depth computations. The Fiat-Shamir paradigm is a general transformation that converts any public-coin interactive proof system to a non-interactive argument in the CRS model. It is used extensively in practice for constructing signature schemes [36], SNARGs [7, 59], and non-interactive zero knowledge
(NIZK) proofs [71].

[1] The term SNARG often refers to NP computation, but the focus of this work is on bounded-depth deterministic computation.

Loosely speaking, the Fiat-Shamir transform converts an interactive proof (P, V) for a language L to a non-interactive argument (P', V') for L in the CRS model. The CRS consists of randomly chosen hash functions $h_1, \ldots, h_\ell$ from a hash family $\mathcal{H}$, where ℓ is the number of rounds in (P, V). To compute a non-interactive proof for $x \in L$, the non-interactive prover $P'(x)$ generates a transcript corresponding to $(P, V)(x)$, denoted by $(\alpha_1, \beta_1, \ldots, \alpha_\ell, \beta_\ell)$, by emulating $P(x)$ and replacing each verifier message β_i by $\beta_i = h_i(\alpha_1, \beta_1, \ldots, \alpha_{i-1}, \beta_{i-1}, \alpha_i)$.[2] The verifier $V'(x)$ accepts if and only if $V(x)$ accepts this transcript and $\beta_i = h_i(\alpha_1, \beta_1, \ldots, \alpha_{i-1}, \beta_{i-1}, \alpha_i)$ for every $i \in [\ell]$.

This paradigm is extremely elegant and simple, and despite its abundant use in practice, its soundness is poorly understood. The Fiat-Shamir paradigm applied to constant round public-coin interactive protocols is sound in the Random Oracle Model (ROM) [6, 68], yet at the same time there are counterexamples that demonstrate its insecurity when applied to interactive arguments [4, 5, 40]. There have been several works that prove its soundness when applied to interactive proofs [21, 22, 44, 53], albeit under extremely strong assumptions. Recently, the work of Canetti *et al.* [21] and the followup work of Peikert and Shiehian [65] prove the soundness of the Fiat-Shamir paradigm, assuming plain LWE, when applied to a *specific* class of protocols, including (a variant of) the three round zero-knowledge proof of graph Hamiltonicity. This result gave the first NIZK for NP from LWE. Subsequently [19] generalized this to obtain NIZK from LPN and DDH/QR/DCR. But instantiating this paradigm under standard assumptions for *succinct* interactive proof systems has remained an important open problem.[3]

This Work. We instantiate that the Fiat-Shamir paradigm under the sub-exponential LWE assumption for a large class of multi-round interactive proofs, that we call FS-*compatible*. This means that we construct explicit hash functions (assuming sub-exponential LWE), such that the Fiat-Shamir transform instantiated with these hash functions is provably sound for any FS-compatible proof.

The class of FS-*compatible* interactive proofs includes the GKR interactive proof [41] and the celebrated sum-check protocol [58, 70]. By instantiating Fiat-Shamir for the GKR protocol, we directly obtain SNARGs for bounded depth from sub-exponential LWE. Our SNARGs are also post-quantum secure assuming the quantum sub-exponential hardness of LWE. Finally, instantiating the Fiat-Shamir paradigm for sumcheck has important implications to PPAD hardness, which we discuss in further detail later in this introduction.

Recall that the Fiat-Shamir paradigm requires a prover to non-interactively emulate verifier messages by computing a hash of the transcript so far. We define a special type of hash function family, namely a *lossy correlation intractable hash family*, and construct it based on the hardness of the LWE assumption. We then show that the Fiat-Shamir transform applied to any FS-compatible interactive proof is sound when using any sub-exponentially secure lossy correlation intractable hash family. This establishes the soundness of the non-interactive GKR and sum-check protocols (obtained by applying the Fiat-Shamir transform), assuming the sub-exponential hardness of LWE.

Previously, the work of Canetti *et al.* [21] proved soundness of the Fiat-Shamir paradigm when applied to the (succinct) GKR protocol, albeit under a very strong assumption: the existence of a fully homomorphic encryption (FHE) scheme that has *perfect circular security*, that is, every poly-size adversary, given Enc(sk), outputs sk with probability at most the probability of guessing (i.e., probability at most $\mathrm{poly}(\kappa) \cdot 2^{-\kappa}$, where κ is the security parameter). By contrast, in our work we achieve soundness assuming the sub-exponential hardness of LWE.

PPAD Hardness. Our results mentioned above have an important implication to the hardness of the complexity class PPAD, as defined by Papadimitriou [63]. The importance of this class, as well as the motivation for studying its hardness, stems from the fact that the problem of finding a Nash equilibrium is known to be PPAD-complete [27, 33].

There has been significant interest in reducing the hardness of PPAD to that of various cryptographic assumptions [1, 14, 29, 30, 35, 37, 45, 50, 57], which we discuss in Section 1.2.2. The most relevant to us is the recent work of Choudhuri *et al.* [29], which shows average-case PPAD hardness assuming average-case #SAT hardness and assuming adaptive soundness of the Fiat-Shamir transform applied to the sum-check protocol. Combining our result on the soundness of the Fiat-Shamir transformation applied to the sum-check protocol, with the result of Choudhuri *et al.* [29], we conclude that the sub-exponential hardness of LWE implies the sub-exponential average-case hardness of PPAD.[4] Moreover, since LWE is believed to be sub-exponentially hard even for quantum computers, and our proof (i.e., our reduction) only interacts with the adversary via black-box, straight-line access, we obtain sub-exponential average-case *quantum-hardness* of the complexity class PPAD, assuming sub-exponential quantum hardness of LWE.

1.1 Our Results

We now describe our results in some detail. First, we define and construct (assuming LWE) a lossy correlation intractable hash family, which as discussed above, is a key tool that helps us obtain our SNARGs. A lossy correlation intractable hash family is a combination of a correlation intractable hash family and a lossy trapdoor function family, introduced by the influential works of Canetti, Goldreich and Halevi [23], and Peikert and Waters [66], respectively.

At a high level, a hash family H is *correlation intractable* (CI) for a function family F, if for every $f \in F$ it is computationally hard, given a random hash key k, to find any input x such that $H(k, x) = f(x)$. Recently, the work of Canetti *et. al* [21], and the followup work of Peikert and Shiehian [65], construct a CI hash family for the family of all functions computable by circuits of an a priori fixed polynomial size (and the run-time of their CI hash functions grow with this bound). As mentioned above, these works led to the first construction of NIZK for NP from LWE.

Lossy Correlation Intractable Hash Functions. We define and construct a *lossy* CI hash family H that, in addition to being a CI hash

[2] Throughout this work, we assume w.l.o.g. that the first message is always sent by the prover, and the last one is sent by the verifier (though this last verifier message is moot).

[3] As we elaborate below, Canetti *et al.* [21] also proved the soundness of the Fiat-Shamir paradigm when applied to the succinct GKR protocol, albeit under a strong assumption.

[4] To obtain this conclusion we rely on the fact that sub-exponential hardness of LWE implies sub-exponential average-case hardness of #SAT.

family, has the following property: The hash keys can be generated using an alternative mode, called the *lossy mode*, such that keys generated in the lossy mode are indistinguishable from random hash keys.[5] In addition, in the lossy mode, the output of a CI hash function lies within a (sub-exponentially) small space of outcomes. We construct a lossy CI hash family by combining a CI hash family with a lossy trapdoor function family [66], both of which can be constructed from the LWE assumption.

Theorem 1.1 (Informal). *Under the* LWE *assumption, there exists a lossy correlation intractable hash family for all functions computable by circuits of an a-priori fixed polynomial size.*

The formal definition and construction of a lossy CI hash family (from a CI hash family and a lossy trapdoor function family) can be found in Section 3, and the high level overview can be found in Section 2. We point out that in the formal definition, a lossy CI hash family is associated with parameters (T, T', ω), where T relates to the CI-security,[6], T' relates to the lossy security,[7] and ω relates to the amount of information lost in the lossy mode.

We use a lossy CI hash family (with appropriate parameters) to soundly instantiate the Fiat-Shamir heuristic for a class of interactive proofs, which we call FS-compatible, and describe next. As mentioned above, this class includes the sum-check protocol and the GKR protocol.

FS-*Compatible Proofs.* Loosely speaking, we say that a public-coin interactive proof (for a language L) is FS-compatible if it satisfies the following two properties: (1) It is *round-by-round sound* as defined in [21]. At a high level, this means that every possible transcript prefix defines a statement which is either true or false. If the statement is false, then a cheating prover cannot expand it to an accepting transcript (except with negligible probability). If it is true, then it can be expanded to an accepting transcript (with probability 1). (2) The second property, roughly, is that for any instance x and transcript prefix τ that defines a false statement, there is a non-uniform advice string (which depends only on the verifier's messages in τ) such that there is an efficient (non-uniform) function BAD that is given this advice and, on input the prover's next message α, efficiently samples a verifier's next message β such that the resulting transcript prefix $\tau|\alpha|\beta$ defines a *correct* statement.

We emphasize that it was already proved in [21] that the Fiat-Shamir heuristic is sound when applied to any 2-round FS-compatible interactive proof.[8] In this work we focus on the multi-round setting, specifically on interactive proofs with a polynomial number of rounds, such as the sum-check protocol and the GKR protocol.

The formal definition of FS-compatible proofs can be found in Section 4.1. We mention that this definition is associated with parameters (T', d, ρ), where T' is the time it takes to compute the function State, d is related to the probability of a random verifier message converting a transcript prefix from being rejecting to being accepting (specifically, this probability is required to be at most $d/2^\lambda$, assuming each verifier message is in $\{0,1\}^\lambda$), and ρ is the (polynomial) time it takes to compute the function BAD (given the non-uniform advice).

We prove that when we apply the Fiat-Shamir transform to any FS-compatible interactive proof Π w.r.t. a lossy CI hash family, we obtain a sound non-interactive argument, assuming the parameters (T', d, ρ) of the FS-compatible proof Π and the parameters (T, T', ω) of the lossy CI hash family satisfy a specific relationship.[9] We refer the reader to Section 2 for a more detailed overview, and to Section 4 for the formal treatment.

Theorem 1.2 (Informal). *Applying the Fiat-Shamir transform to any* FS-*compatible interactive proof with (arbitrary) parameters* (T', d, ρ)*, w.r.t. a lossy* CI *hash family with parameters* (T, T', ω)*, results in a sound non-interactive argument, as long as* T *is large enough (depending on* T', d, ω *and on the protocol* Π*).*

We also prove that the sum-check protocol and the GKR protocol are both FS-compatible with appropriate parameters. As a result we obtain a non-interactive version of the sumcheck protocol, as well as SNARGs for bounded depth computation in the common random string model (i.e. where the CRS is unstructured/uniformly random).

Corollary 1.3 (Informal). *Under the sub-exponential hardness of* LWE[10]*, there exists a hash family* H *and a polynomial* p *such that the Fiat-Shamir transform w.r.t.* H *is sound when applied to the sum-check protocol, assuming the sum-check instance*

$$\sum_{b_1,\ldots,b_\ell \in B} g(b_1, \ldots, b_\ell) = v$$

is over a field $\mathbb{F}$ *such that* $\log|\mathbb{F}| \geq p(d, \ell, \log|B|)$*, where* ℓ *is the number of variables,* d *is the univariate degree of* g*, and where* B^ℓ *is the set we are summing over.*

Corollary 1.4 (Informal). *Under the sub-exponential hardness of* LWE*, there exists a hash family* H *such that the Fiat-Shamir transform w.r.t.* H *is sound when applied to the* GKR *protocol. For a log-space uniform circuit* $C : \{0,1\}^N \to \{0,1\}$ *of size* S *and depth* D*, the resulting non-interactive argument has the following efficiency guarantees: the prover runs in time* $\mathsf{poly}(S)$*, the verifier runs in time* $(D + N) \cdot \mathsf{polylog}(S)$*, and the communication complexity is* $D \cdot \mathsf{polylog}(S)$*.*

We also show how to use Corollary 1.3 to obtain PPAD hardness (and more specifically CLS hardness, for the class CLS ⊆ PPAD), following the blueprint of Choudhuri *et al.* [29].

Corollary 1.5 (Informal). CLS *is sub-exponentially hard on average,*[11] *assuming the sub-exponential hardness of* LWE.

A high-level overview can be found in Section 2, and the full theorem and proof can be found in the full version of our paper [47].

[5] In general, hash keys may not be random, rather they may be generated according to some key generation algorithm. We neglect this in this high-level overview, but do consider this general case in the technical sections.

[6] T-CI security means that no $\mathsf{poly}(T)$ adversary can break the CI requirement.

[7] T'-lossy security means that no $\mathsf{poly}(T')$ adversary can distinguish a key generated by $\mathcal{H}$.Gen from a key generated by $\mathcal{H}$.LossyGen.

[8] In their work they focus on what they call *trapdoor* Σ*-protocols*, but their proofs extend readily to any 2-round FS-compatible interactive proof.

[9] In particular, the parameter T' of the FS-compatible proof is equal to the parameter T' of the lossy CI hash family, and therefore the same notation.

[10] By sub-exponential hardness of LWE, we mean that there exists a constant $\epsilon > 0$ such that for $T(\lambda) = 2^{\lambda^\epsilon}$, no $\mathsf{poly}(T(\lambda))$-size adversary can distinguish LWE samples of size λ from uniform, with advantage better than $\mathsf{negl}(T(\lambda))$.

[11] By sub-exponential average-case hardness of CLS we mean that there is a problem in CLS and an efficiently sampleable distribution over instances of this problem that is sub-exponentially hard on average.

1.2 Related Work

In this section, we elaborate on related work, starting with related work on delegating computation in Section 1.2.1, and continuing with related work on PPAD-hardness in Section 1.2.2.

1.2.1 Related Work on Delegating Computation. Many delegation schemes have been proposed in the literature. These schemes can be roughly divided into three categories.

SNARGs. Extensive work, starting from the seminal work of Micali [59] and continuing with [8–10, 32, 39, 42, 56], construct SNARGs for all non-deterministic computations. However, the soundness of these schemes is proved either in the Random Oracle Model [6] or based on non-standard hardness assumptions known as "knowledge assumptions."[12] Such assumptions have been criticized for being non-falsifiable (as in [61]) and for yielding non-explicit security reductions. We mention that some of these works form the basis of several efficient implementations which are used in practice. Other schemes (for deterministic computations) are known based on non-standard assumptions related to obfuscation [2, 11, 24, 25, 28, 55] or multilinear maps [62]. This latter line of works construct SNARGs where the communication complexity grows linearly with the input length.

Very recently, [49] constructed a SNARG (for deterministic computations) based on an efficiently falsifiable decisional assumption on groups with bilinear maps. While this assumption seems reasonable, it is new to their work and has not been studied before. Moreover, it is known to be broken with quantum attacks. Independently, [21] constructed a SNARG for all bounded depth computations, assuming the existence of an FHE scheme with optimal circular security – which appears to be an extremely strong assumption.

Designated verifier schemes. A line of works beginning with [51, 52] and continuing with [3, 17, 18, 48] designed delegation schemes for deterministic computations and a sub-class of non-deterministic computations, based on standard assumptions (such as the hardness of LWE). These schemes, however, are not publicly verifiable. Rather, the CRS is generated together with a secret key, which is needed in order to verify the proofs.

Interactive schemes. In the interactive setting, we can achieve publicly verifiable schemes, even for non-deterministic computations, under standard assumptions. Kilian [54] constructed a four message protocol for any NP language with polylog communication, assuming the existence of a hash family that is collision-resistant w.r.t. sub-exponential time adversaries. It has been shown that this scheme can be converted into a three message protocol assuming a multi-collision resistant hash function [13]. The work of [64] constructs a two-message delegation scheme in addition to a (hard to compute) CRS for low-depth circuits, assuming an attribute-based encryption scheme.

Finally, we mention that in the interactive setting, we can achieve publicly verifiable schemes even unconditionally. For example, [41] and [69] give interactive delegation schemes for bounded depth and bounded space computations with unconditional soundness. As mentioned above, in this work we build on the GKR protocol from [41] to obtain our SNARG.

1.2.2 Related Work on PPAD *Hardness.* Recently, there have been a proliferation of results proving the hardness of the class PPAD, which was defined by Papadimitriou [63]. In his original paper, Papadimitriou suggested proving the hardness of PPAD under cryptographic assumptions. After two decades of little progress on the question, a recent sequence of works [14, 29, 30, 35, 37, 45, 67] established the hardness of PPAD (and even that of a subclass known as $\text{CLS} \subseteq \text{PPAD}$) under strong cryptographic assumptions. The first line of works, starting with that of Bitansky, Paneth and Rosen [14], assumes sub-exponentially secure indistinguishability obfuscation, or functional encryption [37, 45].

The second line of works, which is more relevant to us, started with the work of Choudhuri *et al.* [29] and relies on unambiguous incrementally updateable proofs. The work of [29] assumes the adaptive soundness of the Fiat-Shamir transformation when applied to the sum-check protocol (and assuming that #SAT is hard on average). They then use the work of Canetti *et al.* [21], which proves adaptive soundness of the Fiat-Shamir transformation applied to the sum-check protocol, to obtain PPAD hardness assuming the existence of a perfectly secure FHE (and assuming that #SAT with polylog variables is hard on average). The work of [67] relies on the soundness of the Fiat-Shamir transformation when applied to Pietrzak's interactive proof for repeated squaring [67] (and assuming the hardness of repeated squaring modulo a composite).

Very recently, Lombardi and Vaikuntantanathan [57] proved soundness of the Fiat-Shamir transform applied to Pietrzak's interactive proof for repeated squaring, assuming LWE is $2^{\lambda^{1-\epsilon}}$-hard (w.r.t. a hash function that takes time $T(\lambda) = 2^{\lambda^{\epsilon}}$ time to compute), thus obtaining average-case CLS hardness under this assumption (and assuming that repeated squaring is $2^{\lambda^{\epsilon}}$-hard). Independently, Kalai, Paneth, and Yang [50] obtained average-case CLS hardness under a quasi-polynomial-time assumption on groups with bilinear maps (and assuming SAT with $\log(n)^{1+\epsilon}$ variables is hard on average). Independently, Bitansky and Gerichter [12] prove average case hardness of the class PLS (which is not known to be contained in PPAD) under the same assumption.

Despite these works, obtaining PPAD hardness under a standard cryptographic assumption has remained an important open problem until this work.

Subsequent Work. Subsequent to the initial posting of this manuscript, a sequence of beautiful works achieved indistinguishability obfuscation (iO) from simpler assumptions including: (1) variants of (sub-exponential) leakage-resilient circular-security of LWE-based encryption [16, 38, 72], or (2) (sub-exponential) SXDH, PRGs in NC0, LWE and a variant of LPN over large fields [46]. As noted above, iO is a powerful tool that also implies PPAD hardness and SNARGs (where the communication complexity grows linearly with the input length). However, besides requiring additional assumptions, all existing constructions (of iO) are extremely complex and prohibitively inefficient. On the other hand, we offer a completely different perspective towards obtaining these results from

[12] For example, the Knowledge-of-Exponent assumption [31] asserts that any efficient adversary that is given two random generators (g, h) and outputs (g^z, h^z) must also "know" the exponent z.

only (sub-exponential) LWE. We show that Fiat-Shamir is sound for a large class of interactive proofs (variants of which are often used in practice), when instantiated with a simple *lossy correlation-intractable hash function.*

2 TECHNICAL OVERVIEW

We now outline our technical approach. We build on ideas from [21, 65] to argue that the Fiat-Shamir paradigm is sound when applied to a rich class of public-coin interactive proofs (and in particular, when applied to the sum-check and the GKR protocols). In what follows, we first discuss the hash functions constructed by [21, 65] and explain the difficulties with directly applying these hash functions to the sum-check (and to the GKR) protocol. We then explain our key idea of using a lossy trapdoor function family to get around these difficulties.

Background: Correlation Intractable Hash Functions [23]. At a high level, a hash family H is correlation intractable (CI) for a relation $\mathcal{R}(x, y)$ if it is computationally hard, given a random hash key k, to find any input x such that $(x, H(k, x)) \in \mathcal{R}$. It was observed in [23] that the Fiat-Shamir transform is sound if the initial protocol is statistically sound and the hash family used to reduce interaction is a CI hash family for all sparse relations.

Very recently, the beautiful works of [21, 65] constructed a CI hash family for a restricted class of relations, assuming circular-secure LWE, and subsequently, plain LWE. Very roughly, the relations considered in these works are functions (i.e., for every x there is a single y such that $\mathcal{R}(x, y) = 1$) of a priori bounded size. Specifically, for any polynomial ρ they consider the class F of all functions computable by a ρ-size circuit. They construct a CI hash family H such that for any function $f \in F$ and any poly-size $\mathcal{A}$:

$$\Pr_{k \leftarrow \mathsf{Gen}(1^\lambda)}[\mathcal{A}(k) = x \ : \ H(k, x) = f(x)] = \mathsf{negl}(\lambda).$$

The works of [21, 65] used the Fiat-Shamir paradigm, together with this hash family, to obtain a NIZK proof for NP based on LWE. Their main conceptual observation, which is the starting point of our work, is the following: There are several interesting interactive proofs for which the "bad" verifier challenge, which allows a prover to cheat, can be computed by an *efficient* (non-uniform) function.[13] Therefore, replacing the verifier message by the output of a CI hash function results in a verifier message that does not allow a prover to cheat, except with negligible probability.

In this work, we focus on applying the Fiat-Shamir transformation to succinct protocols, such as the sum-check and GKR protocols. In what follows, we first focus on the sum-check protocol, and explain why the approach of [21, 65] fails when applied to the sum-check protocol. We then define a special CI hash family, which we call a *lossy* CI hash family. We show how this overcomes the failure point above, and argue that the resulting non-interactive sum-check protocol, obtained by applying the Fiat-Shamir transformation w.r.t. a lossy CI hash family, is sound. Then we show how to construct such a lossy CI hash famiy from a CI hash family and a lossy trapdoor family. Finally, we show that a lossy CI hash family can be used to securely instantiate the Fiat-Shamir paradigm for a broader class of interactive proofs, which we refer to as FS-*compatible* protocols (which includes the GKR protocol as well as the sum-check protocol). We end this overview with a brief explanation of how we obtain our PPAD hardness result.

We start with a brief description of the sum-check protocol.

The Interactive Sum-Check Protocol. In the sum-check protocol, the prover convinces the verifier that

$$\sum_{b_1,\ldots,b_\ell \in B} g(b_1, \ldots, b_\ell) = v,$$

for some known polynomial g of degree at most d in each variable over some large field $\mathbb{F}$ that contains the subset $B \subset \mathbb{F}$. The first message from the prover is the univariate polynomial

$$g_1(\cdot) = \sum_{b_2,\ldots,b_\ell \in B} g(\cdot, b_2, \ldots, b_\ell).$$

The verifier checks that g_1 is of degree $\leq d$ and that $\sum_{b \in B} g_1(b) = v$. If this is not the case, it rejects. Otherwise, it sends to the prover a random $t_1 \leftarrow \mathbb{F}$, and the task is reduced to proving that

$$\sum_{b_2,\ldots,b_\ell \in B} g(t_1, b_2, \ldots, b_\ell) = g_1(t_1).$$

In the next round, the prover sends

$$g_2(\cdot) = \sum_{b_3,\ldots,b_\ell \in B} g(t_1, \cdot, b_3, \ldots, b_\ell).$$

The verifier checks that g_2 is a univariate polynomial of degree $\leq d$ and that $\sum_{b \in B} g_2(b) = g_1(t_1)$. If this is not the case, it rejects. Otherwise, it sends a random $t_2 \leftarrow \mathbb{F}$, and the task is reduces to proving that

$$\sum_{b_3,\ldots,b_\ell \in B} g(t_1, t_2, b_3, \ldots, b_\ell) = g_2(t_2).$$

This continues for ℓ rounds, at the end of which the verifer checks on its own that $g_\ell(t_\ell) = g(t_1, \ldots, t_\ell)$ for a random $t_\ell \leftarrow \mathbb{F}$.

Non-Interactive Sum-Check via Fiat-Shamir: First Attempt. We consider applying the Fiat-Shamir transform [36] instantiated with a CI hash family (for all functions computable by bounded size circuits) to the sum-check protocol, in order to get a non-interactive argument in the CRS model. Specifically, the CRS contains ℓ hash keys, $k_1, \ldots, k_\ell$, one for each verifier message, and the prover non-interactively computes the i'th verifier message as $H(k_i, \tau_i)$, which is the outcome of the i'th hash function $H(k_i, \cdot)$ applied to the transcript so far.

Recall that in order to use the ideas from [21, 65], we need to argue that there is an efficient (non-uniform) function that computes the "bad" challenge, where a bad challenge is one that allows the prover to cheat. Let us first understand what a bad challenge is in the context of the sum-check protocol. Denote by $g_1, \ldots, g_\ell$ the messages of the honest prover in the sum-check protocol, and denote by $g_1^*, \ldots, g_\ell^*$ the messages of a malicious prover. Note that if a malicious prover (successfully) cheats, then it must be that $g_1^* \neq g_1$. The verifier sends a challenge $t_1 \leftarrow \mathbb{F}$, which reduces the task of proving the original sum-check to the task of proving that

$$\sum_{b_2,\ldots,b_\ell \in B} g(t_1, b_2, \ldots, b_\ell) = g_1^*(t_1).$$

[13] We note that this non-uniform advice is not efficiently computable, which is what makes these interactive proofs non-trivial.

We say that t_1 is a bad challenge if $g_1^*(t_1) = g_1(t_1)$, since it converts a false statement to a true statement, and thus allows the prover to cheat.

The first issue we encounter is the following: The polynomial $g_1^* - g_1$ is of degree d and therefore may have as many as d roots, which implies that there can be as many as d bad challenges. Recall that the CI hash family defined and constructed in [21, 65] avoids only a *single, efficiently computable* bad challenge, whereas here we have d possible bad challenges that we must all avoid.

Fortunately, *any* CI hash family, as described above, that avoids only a single bad challenge readily extends to avoid d possible bad challenges (as observed in [21]). Namely, if we let f denote an efficiently computable function that outputs one of the d bad challenges at random, then for any poly-size adversary $\mathcal{A}$,

$$\Pr_{k \leftarrow \mathsf{Gen}(1^\lambda)} [\mathcal{A}(k) = x \ : \ H(k, x) = f(x)] = \mathsf{negl}(\lambda).$$

This means that if $f_1(x), \ldots, f_d(x)$ are the d bad challenges, then for any poly-size $\mathcal{A}$,

$$\Pr_{k \leftarrow \mathsf{Gen}(1^\lambda)} [\mathcal{A}(k) = x \ : \ H(k, x) \in \{f_1(x), \ldots, f_d(x)\}] = d \cdot \mathsf{negl}(\lambda).$$

The next main obstacle is the following: in order to apply the CI hash family of [21, 65], we need the bad challenge function (that outputs a random bad challenge) to be efficiently computable. It is quite straightforward to argue that the bad challenge function corresponding to the *first* verifier challenge is efficiently computable: The non-uniform advice is the (true) function g_1 and the function on input g_1^* outputs a random root of $g_1^* - g_1$, which can be computed efficiently (for example) via the Cantor-Zassenhaus algorithm [26].

The problem kicks in after the first message. For a general round $i \in [\ell]$, the bad challenges are the roots of the polynomial $g_i^* - g_i$, for the same reason as for the first round: the prover can transition from a false statement in round i to a true statement in round $i + 1$ if and only if $g_i(t_i) = g_i^*(t_i)$. It is tempting to simply hardwire g_i as the non-uniform advice. However, recall that

$$g_i(\cdot) = \sum_{b_{i+1}, \ldots, b_\ell \in B} g(t_1, \ldots, t_{i-1}, \cdot, b_{i+1}, \ldots, b_\ell),$$

and thus, g_i depends on the first $i - 1$ challenges of the verifier $t_1, \ldots, t_{i-1}$. In the non-interactive setting, these challenges depend on the previous prover messages, and thus cannot be a priori fixed and hardwired as non-uniform advice. Moreover, computing this function g_i (as opposed to taking it as non-uniform advice) takes time $O(|B|^{\ell-i})$, which may be super-polynomial in general.

As a first attempt, we consider guessing the first $i - 1$ values $t_1, \ldots, t_{i-1}$, using these guesses to (non-uniformly) compute g_i, and hardwiring the resulting g_i into the bad challenge function. As before, on input g_i^*, the bad challenge function f efficiently computes a random root of the polynomial $g_i^* - g_i$ via the Cantor-Zassenhaus algorithm [26]. Now if the challenges $t_1, \ldots, t_{i-1}$ were guessed correctly, the function f indeed outputs a random bad challenge.

Note that for every $i \in [\ell]$, $t_1, \ldots, t_{i-1}$ are guessed correctly with probability $1/|\mathbb{F}|^{i-1}$. Unfortunately, we cannot afford to have such a small probability of guessing correctly. We cannot even afford our guess to be correct with probability $1/|\mathbb{F}|$, since our hash functions output hash values in $\{0, 1\}^{\log |\mathbb{F}|}$ (they output elements $t \in \mathbb{F}$). Thus we cannot hope to argue that a poly-size adversary outputs a bad challenge with probability smaller than $1/|\mathbb{F}|$ (since a random guess will be a bad challenge with probability $\geq 1/|\mathbb{F}|$). However, we *could* afford guessing correctly with probability $2^{-(\log |\mathbb{F}|)^\epsilon}$ (for a small enough constant $\epsilon > 0$), and then rely on sub-exponential security of their CI hash family (which in turn corresponds to relying on the sub-exponential hardness of the LWE assumption).

Lossy Correlation Intractable Hashing to the Rescue. In order to ensure that the correct polynomial g_i is guessed and hardwired to the bad challenge function with large enough probability, we will make it possible to *artificially* decrease the output space of the hash function family *in the proof of security*, so that $t_1, \ldots, t_{i-1}$ are guessed correctly with probability at least $2^{-(\log |\mathbb{F}|)^\epsilon}$. To this end, we define and construct a *lossy correlation-intractable hash family*, which in addition to satisfying the CI property discussed above, has an extra *lossy mode*. In this mode, the space of outcomes (or image) of the hash function is restricted to being of size at most $2^{(\log |\mathbb{F}|)^\epsilon}$ for a small enough constant $\epsilon > 0$.

To see why this could be helpful, consider the non-interactive sum-check protocol obtained by applying the Fiat-Shamir transform, where the CRS contains ℓ hash keys $k_1, \ldots, k_\ell$ for the lossy CI hash family. Now suppose there exists a cheating prover P^* that successfully cheats in the resulting non-interactive sum-check protocol with non-negligible probability. Recall that this means there must be a round $i \in [\ell]$ such that $g_i^* \neq g_i$ and yet $g_i^*(t_i) = g_i(t_i)$ (with non-negligible probability).

In the analysis, one can consider an alternative distribution of hash keys $k_1, \ldots, k_\ell$, where the first $i - 1$ hash keys $k_1, \ldots, k_{i-1}$ are generated using the lossy mode, and the rest of the keys are generated as before (using the standard mode). Assuming that keys generated via the lossy mode are (sufficiently) indistinguishable from keys generated using the standard mode, we conclude that it will still be the case that $g_i^* \neq g_i$ and yet $g_i^*(t_i) = g_i(t_i)$ (with non-negligible probability). More precisely, to be able to make this claim, we will assume that keys generated in the lossy mode are indistinguishable even for $\mathsf{poly}(|B|^\ell)$-size adversaries – because g_i can be computed in time $\mathsf{poly}(|B|^\ell)$. At this point, we can guess $t_1, \ldots, t_{i-1}$ with probability $(2^{-(\log |\mathbb{F}|)^\epsilon})^{i-1} \geq 2^{-(\log |\mathbb{F}|)^{2\epsilon}}$ assuming $\log |\mathbb{F}| \geq \ell^{1/\epsilon}$, and in turn contradict the sub-exponential correlation intractability property of the underlying CI hash family.

Constructing Lossy CI *Hash Functions.* Our construction of a lossy CI hash family for all bounded-size circuits combines a lossy trapdoor family [66] with a CI hash family for all bounded-size circuits [65], both of which can be constructed based on the LWE assumption. As mentioned above, we will need our lossy CI hash family to be sub-exponentially secure, and as a result we rely on the sub-exponential hardness of LWE.

A lossy trapdoor family, defined and constructed by Peikert and Waters [66], is a keyed family of functions where keys can be generated in two modes: an injective mode and a lossy mode. The injective mode has a corresponding trapdoor which can be used to efficiently invert the function. The lossy mode, in contrast, information theoretically loses information about its input. More specifically, it is associated with a lossy parameter ω and the guarantee is that the size of the function's output space is bounded by $2^{n-\omega}$, assuming the domain is $\{0,1\}^n$.

We construct lossy CI hash functions by concatenating CI hash functions with lossy trapdoor functions. Each lossy CI hash key consists of a pair of keys (k, k) where k is a key for a (underlying) CI hash function and k is a key for the lossy trapdoor function.

In the normal mode of operation, k is generated using the injective mode of the underlying lossy trapdoor family. In the lossy mode of operation of our lossy CI hash function, k is generated using the lossy mode of the underlying lossy trapdoor family. To evaluate a lossy CI hash function with keys (k, k) on input x, we first compute the lossy trapdoor function with key k on input x to obtain a value y, and then evaluate the CI hash function with key k on input y. Denoting the underlying CI hash family by H, the underlying lossy family by G, and our resulting lossy CI hash family by H', we have that

$$H'((k,\mathsf{k}),x) \triangleq H(k, G(\mathsf{k},x)).$$

We denote the standard key generation algorithm (which generates (k, k) where k is an injective key) by Gen and the lossy one (where k is a lossy key) by LossyGen.

The indistinguishability of keys generated by Gen and keys generated by LossyGen follows from the security of the trapdoor hash family, which asserts that injective keys are indistinguishable from lossy ones. Importantly, as we argue below, *the* Gen *mode continues to satisfy correlation intractability* (for all functions computable by bounded-size circuits, though the bound here is smaller than the bound for the underlying CI hash family H). Roughly speaking, this is due to the trapdoor inversion algorithm, together with the fact that H is a CI hash family (for all functions computable by bounded-size circuits).

FS-*Compatible Interactive Proofs.* The same argument extends to prove soundness of the Fiat-Shamir transformation for a larger class of protocols, which we call FS-*compatible.* As described in section 1.1, these are protocols that satisfy the properties of *round-by-round soundness*, and having an *efficiently computable* BAD *function.* We describe these properties formally below.

- **Round-by-round soundness [21].** There is a (not necessarily efficient) algorithm State that takes as input an instance x and a transcript prefix $\tau = (\alpha_1, \beta_1, \dots, \alpha_i, \beta_i)$, and outputs accept or reject, such that for every (x, τ), if $\mathsf{State}(x, \tau)$ = reject then for any next message of the prover α, with overwhelming probability over the next verifier message β, it holds that $\mathsf{State}(x, \tau|\alpha|\beta)$ = reject. Moreover, for every $x \notin L$, $\mathsf{State}(x, \emptyset)$ = reject, and for every $x \in L$ and honestly generated prefix τ, $\mathsf{State}(x, \tau)$ = accept
- **Efficient** BAD **function.** For every $x \notin L$ and every $i \in [\ell]$ (where ℓ denotes the number of rounds), and every fixing of the first $i-1$ verifier messages, denoted by $\beta_1, \dots, \beta_{i-1}$, there exists a (non uniform) efficient randomized function BAD that takes the first i messages of the prover, denoted by $\alpha_1, \dots, \alpha_i$, and outputs an element β_i, such that if $\mathsf{State}(x, \tau)$ = reject, for $\tau = (\alpha_1, \beta_1, \dots, \alpha_{i-1}, \beta_{i-1}, \alpha_i)$, then β_i is a random element in the (small) set
$$\{\beta \ : \ \mathsf{State}(x, \tau|\beta) = \mathsf{accept}\}. \quad (1)$$

We associate with each FS-compatible protocol parameters (T', d, ρ), where T' is the time it takes to compute the function State,[14] d is a bound on the number of bad challenges, i.e., a bound on the size of the set defined in Equation (1),[15] and ρ is a polynomial bound on the size of the circuit computing the function BAD.

We prove that both the sum-check protocol and the GKR protocol are FS-compatible (w.r.t. some parameters (T', d, ρ)). Moreover, we prove that applying the Fiat-Shamir transform w.r.t. a lossy CI hash family (for all functions computable by a bounded-size circuit) to any FS-compatible interactive proof results in a sound non-interactive argument. More specifically, if the interactive proof is (T', d, ρ) FS-comptible, then to prove soundness, the lossy CI hash family needs to be (T, T', ω) secure for all functions computable by ρ-size circuits (for a large enough T), which means that the following three conditions hold:

- **T'-Key Indistinguishability.** A poly(T')-size adversary cannot distinguish between lossy keys (generated by LossyGen) and standard ones (generated by Gen).
- **T-Correlation Intractability.** For any poly(T)-size adversary $\mathcal{A}$, and any function f computed by a ρ-size circuit,
$$\Pr[\mathcal{A}(k) = x \ : \ H(k, x) = f(x)] = \mathsf{negl}(T).$$
- **ω-Lossiness.** For every key k generated by LossyGen, the number of elements in the co-domain of this hash family is $2^{n-\omega}$, assuming the domain is $\{0,1\}^n$.

Moreover, T should be larger than the inverse probability of guessing all the verifier's messages in a protocol transcript, assuming the hash functions are generated in lossy mode.

Applying this to the GKR protocol implies that the resulting non-interactive GKR is sound. However, in the analysis we get that T must be exponential in the number of rounds, which in the GKR protocol corresponds to the depth of the computation being delegated, which implies that the communication complexity (as well as the verifier running time) becomes polynomial in the depth (as opposed to linear in the depth). To obtain verification time that is linear in the depth of computation, we consider a refined version of the above analysis, where we consider *history-independent* protocols, which are protocols where the prover and verifier can forget the initial transcript, and only compute their messages largely as a function of the last ν rounds (for some parameter ν). For such ν-history-independent protocols, we get better parameters, and in particular, need $T \geq 2^{\nu(n-\omega)}$, where in the GKR protocol ν does not depend on the depth, but rather depends only poly-logarithmically on the size of the circuit being delegated.

PPAD Hardness. To establish the sub-exponential hardness of Nash under sub-exponential LWE, we rely on the beautiful work of Choudhuri et. al. [29]. Specifically, [29] showed that *adaptive unambiguous soundness of Fiat-Shamir for sum-check* can be used to reduce #SAT instances to rSVL instances, where rSVL is a problem in the class CLS ⊆ PPAD. Roughly, unambiguity requires that it is (computationally) hard to find *two different accepting proofs*, even for a true statement. While our non-interactive sum-check protocol

[14] This function is usually inefficient for the verifier to compute on his own.

[15] The parameter d can be super-polynomial, though for the sum-check and GKR protocols it is polynomial.

does satisfy unambiguity due to the special structure of the sum-check protocol, we unfortunately fall short of proving full-fledged adaptive soundness. Loosely speaking, this is due to the fact that the multi-variate polynomial g needs to be known in advance in order to precompute the true claims g_i for each round $i \in [\ell]$.

However, we observe that the proof in [29] does not require full adaptivity over the choice of g. Specifically, they require a weaker form of adaptive unambiguous soundness, which we call *prefix adaptive* unambiguous soundness, where g is chosen non-adaptively but a prefix $\sigma_1, \ldots, \sigma_j$ can be chosen adaptively, and they require that the non-interactive sum-check proof of the (partially adaptive) statement

$$\sum_{b_{j+1},\ldots,b_\ell \in B} g(\sigma_1, \ldots, \sigma_j, b_{j+1}, \ldots, b_\ell)$$

satisfies soundness and unambiguity. We also observe that their proof only requires unambiguous soundness to hold for prefixes $\sigma_1, \ldots, \sigma_j$ for which each element σ_i in the prefix is either in the support of the hash function or an element in $[0, d]$.[16]

The proof techniques we develop allow us to prove such prefix adaptive unambiguous soundness for the non-interactive sum-check protocol. We mention that the guarantee that the prefix $\sigma_1, \ldots, \sigma_j$ is in the image of the hash function (or is in $[0, d]$) is crucial for us, since otherwise the adaptivity in σ would be too large for us to handle. Since we have the guarantee that these elements are in the image of the hash function, in the analysis we can modify the hash keys to be lossy and hence obtain the guarantee that $\sigma_1, \ldots, \sigma_j$ each belong to a small set of size $2^{n-\omega}$, which is small enough for us to guess these values in order to prove unambiguous soundness of the non-interactive sum-check protocol. We refer the reader to the full version of this paper [47] for details.

Notation. Throughout this paper, for any probabilistic algorithm $\mathcal{A}$ and any input x, we use the terminology "for every $y \in \mathcal{A}(x)$" as a shorthand to say "for every y in the support of $\mathcal{A}(x)$". Throughout this work, we consider interactive proofs $\Pi = (P, V)$, and make the following simplifying assumptions:

(1) Denoting by ℓ the number of rounds in Π, and we always assume for simplicity and w.l.o.g. that each round consists of two messages, the first sent by the prover and the second sent by the verifier. Thus a transcript of an ℓ-round protocol is of the form $\tau = (\alpha_1, \beta_1, \ldots, \alpha_\ell, \beta_\ell)$, where $(\alpha_1, \ldots, \alpha_\ell)$ are the prover messages and $(\beta_1, \ldots, \beta_\ell)$ are the verifier messages. For every $i \in [\ell]$, we denote by τ_i the first i rounds of τ. Namely, $\tau_i \triangleq (\alpha_1, \beta_1, \ldots, \alpha_i, \beta_i)$. For every $1 \le i < j \le \ell$, we denote by $\tau_{[i,j]} \triangleq (\alpha_i, \beta_i, \ldots, \alpha_j, \beta_j)$.

(2) We assume w.l.o.g. that all of the verifier messages $\beta_1, \ldots, \beta_\ell$ are of the same length, and we denote this length by $\lambda = \lambda(|x|)$, where x is the input. Intuitively, this length determines the soundness of the interactive proofs that we are interested in (FS-compatible ones).

We think of all the parameters (except $|x|$) as a function of λ, unless explicitly stated otherwise. For example, we think of ℓ as a function of λ, as opposed to a function of $|x|$. This is useful since we use a Fiat-Shamir hash function with security parameter $\lambda(|x|)$, and thus bounding all the parameters in terms of λ will be helpful when proving soundness of the Fiat-Shamir transformation.

3 LOSSY CORRELATION INTRACTABLE HASH FUNCTIONS

In this section we define and construct the lossy correlation intractable (lossy CI) hash family that we use for instantiating the Fiat-Shamir paradigm. This is obtained by combining any lossy trapdoor function family with any correlation intractable (CI) hash family for bounded size circuits. In Section 3.1, we recall the notion of a lossy trapdoor function family and the notion of a CI hash family. Then, in Section 3.2, we define our notion of a lossy CI hash family, and provide a construction.

3.1 Preliminaries

3.1.1 Lossy Trapdoor Functions. Lossy trapdoor functions were first defined and constructed (based on LWE) in an influential work of Peikert and Waters [66]. Loosely speaking, a lossy trapdor function family contains two types of functions: injective ones and lossy ones, such that one cannot distinguish between a random injective function in the family and a random lossy function in the family. An injective function can be generated together with a trapdoor, which allows one to efficiently invert the function, whereas a lossy function "loses" most information about its preimage, since its image is much smaller than its domain.

Definition 3.1 ((T, ω)-Lossy Trapdoor Family). A quadruple of PPT algorithms $(\mathsf{InjGen}, \mathsf{LossyGen}, \mathsf{Eval}, \mathsf{Inv})$ is said to be a (T, ω)-lossy trapdoor function family if there exist polynomials $n = n(\lambda)$, $n' = n'(\lambda)$, $s = s(\lambda)$ and $t = t(\lambda)$ for which the following syntax and properties are satisfied:

- **Syntax.**
  - $\mathsf{InjGen}(1^\lambda)$ takes as input a security parameter 1^λ and outputs a pair $(\mathsf{k}, \mathsf{td})$, where $\mathsf{k} \in \{0,1\}^s$ is a key corresponding to an injective function and $\mathsf{td} \in \{0,1\}^t$ is a corresponding trapdoor.
  - $\mathsf{LossyGen}(1^\lambda)$ takes as input a security parameter 1^λ and outputs a key $\mathsf{k} \in \{0,1\}^s$ corresponding to a lossy function.
  - $\mathsf{Eval}(\mathsf{k}, x)$ takes as input a key $\mathsf{k} \in \{0,1\}^s$ and an element $x \in \{0,1\}^n$ and outputs an element $y \in \{0,1\}^{n'}$.
  - $\mathsf{Inv}(\mathsf{k}, \mathsf{td}, y)$ takes as input a key $\mathsf{k} \in \{0,1\}^s$, a trapdoor $\mathsf{td} \in \{0,1\}^t$, and an element $y \in \{0,1\}^{n'}$, and outputs an element $x \in \{0,1\}^n \cup \{\perp\}$.
- **Properties.** The following properties hold:
  - **Injective Mode.** For every $\lambda \in \mathbb{N}$ and every $\mathsf{k} \in \mathsf{InjGen}(1^\lambda)$ the function $\mathsf{Eval}(\mathsf{k}, \cdot)$ is injective. Furthermore, for every $x \in \{0,1\}^{n(\lambda)}$, $\Pr[\mathsf{Inv}(\mathsf{k}, \mathsf{td}, \mathsf{Eval}(\mathsf{k}, x)) = x] = 1$.[17]
  - **ω-Lossiness.** For every $\lambda \in \mathbb{N}$ and every $\mathsf{k} \in \mathsf{LossyGen}(1^\lambda)$ the function $\mathsf{Eval}(\mathsf{k}, \cdot)$ has an image of size $2^{n(\lambda)-\omega(\lambda)}$.

[16] We identify the set $[0, d]$ with a set of $d + 1$ field elements.

[17] Typically, this requirement is only required to hold with overwhelming probability. We require it to hold with probability 1, only for the sake of simplifying the proof. This stronger requirement can be obtained assuming a leveled FHE with perfect correctness, following the construction in [15, 43].

  - T-**Security.** There exists a negligible function μ such that for every poly(T)-size adversary $\mathcal{A}$ and for every $\lambda \in \mathbb{N}$,

$$\left| \Pr_{\mathsf{k} \leftarrow \mathcal{G}.\mathsf{LossyGen}(1^\lambda)}[\mathcal{A}(\mathsf{k}) = 1] - \Pr_{\mathsf{k} \leftarrow \mathcal{G}.\mathsf{InjGen}(1^\lambda)}[\mathcal{A}(\mathsf{k}) = 1] \right| = \mu(T(\lambda))$$

Theorem 3.2. *[15, 34, 66]. Assuming the sub-exponential hardness of* LWE*, for every constant $0 < \delta < 1$ and every polynomial $n(\lambda)$, there exists a constant $0 < \epsilon < 1$ for which there exists a (T, ω)-lossy function family for $\omega(\lambda) = n(\lambda) - \lambda^\delta$ and $T = 2^{\lambda^\epsilon}$.*

3.1.2 CI *Hash Family.*

In this section, we recall the notion of a CI hash family. We start by recalling the notion of a hash function family.

Definition 3.3. A hash family $\mathcal{H}$ is associated with two algorithms ($\mathcal{H}$.Gen, $\mathcal{H}$.Hash), and a parameter $n = n(\lambda)$, such that:

- $\mathcal{H}$.Gen is a PPT algorithm that takes as input a security parameter 1^λ and outputs a key k.
- $\mathcal{H}$.Hash is a polynomial time computable (deterministic) algorithm that takes as input a key $k \in \mathcal{H}.\mathsf{Gen}(1^\lambda)$ and an element $x \in \{0,1\}^{n(\lambda)}$ and outputs an element y.

In this work we focus on hash families $\mathcal{H}$ such that for every $\lambda \in \mathbb{N}$, every key $k \in \mathcal{H}.\mathsf{Gen}(1^\lambda)$ and every $x \in \{0,1\}^{n(\lambda)}$, the output $y = \mathcal{H}.\mathsf{Hash}(k, x)$ is in $\{0,1\}^\lambda$. Throughout this work, we always assume that this is the case.

In what follows when we refer to a hash family, we usually do not mention the domain parameter n explicitly.

Definition 3.4 (T-Correlation Intractable). [23] A hash family $\mathcal{H} = (\mathcal{H}.\mathsf{Gen}, \mathcal{H}.\mathsf{Hash})$ is said to be *T-correlation intractable* (T-CI) for a function family $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ if the following two properties hold:

- For every $\lambda \in \mathbb{N}$, every $f \in \mathcal{F}_\lambda$, and every $k \in \mathcal{H}.\mathsf{Gen}(1^\lambda)$, the functions f and $\mathcal{H}.\mathsf{Hash}(k, \cdot)$ have the same domain and the same co-domain.
- For every poly(T)-size $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}}$ there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$ and every $f \in \mathcal{F}_\lambda$,

$$\Pr_{\substack{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda) \\ x \leftarrow \mathcal{A}(k)}}[\mathcal{H}.\mathsf{Hash}(k, x) = f(x)] = \mu(T(\lambda)).$$

The work of Canetti *et al.* [21] constructs a CI hash family for any function family $\mathcal{F}$ that consists of functions computable by bounded size circuits (where the runtime of the CI hash functions grows polynomially with this bound), assuming the existence of a sub-exponential circular secure FHE scheme. Following their work, Peikert and Shiehian [65] overcome the need to rely on circular security and obtain such a CI hash family from the sub-exponential LWE assumption.

Theorem 3.5. *[65] Assuming the sub-exponential hardness of* LWE*, there exists a constant $0 < \epsilon < 1$ such that for any polynomial S and any function family $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$, where $\mathcal{F}_\lambda$ consists of functions that are computable by circuits of size $S(\lambda)$, there exists a T-*CI *hash family $\mathcal{H} = (\mathcal{H}.\mathsf{Gen}, \mathcal{H}.\mathsf{Hash})$ for $\mathcal{F}$ (Definition 3.4), where $T = 2^{\lambda^\epsilon}$. Moreover, there exists a polynomial p such that for every $\lambda \in \mathbb{N}$ and every $k \in \mathcal{H}.\mathsf{Gen}(1^\lambda)$, the function $\mathcal{H}.\mathsf{Eval}(k, \cdot)$ is computable by a circuit of size $p(S(\lambda))$.*

3.2 Lossy CI Hash Functions

In this section we show how to take any CI hash family (Definition 3.4), together with any family of lossy trapdoor functions (Definition 3.1), and obtain what we call a *lossy* CI hash family.

Definition 3.6 ((T, T', ω)-Lossy CI*).* A hash family

$$\mathcal{H} = (\mathcal{H}.\mathsf{Gen}, \mathcal{H}.\mathsf{LossyGen}, \mathcal{H}.\mathsf{Hash})$$

is said to be (T, T', ω)-*lossy* CI for a function family $\mathcal{F}$ if the following holds:

- ($\mathcal{H}$.Gen, $\mathcal{H}$.Hash) is a T-CI hash family for $\mathcal{F}$ (Definition 3.4).
- The additional key generation algorithm $\mathcal{H}$.LossyGen takes as input a security parameter λ and outputs hash key k, such that the following two properties hold:
  - T'-**Key Indistinguishability.** For every poly(T')-size adversary $\mathcal{A}$, there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$

$$\left| \Pr_{k \leftarrow \mathcal{H}.\mathsf{LossyGen}(1^\lambda)}[\mathcal{A}(k) = 1] - \Pr_{k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)}[\mathcal{A}(k) = 1] \right| = \mu(T'(\lambda)).$$

  - ω-**Lossiness.** For every $\lambda \in \mathbb{N}$ and $k \in \mathcal{H}.\mathsf{LossyGen}(1^\lambda)$, denoting by $n = n(\lambda)$ the length of elements in the domain of $\mathcal{H}.\mathsf{Hash}(k, \cdot)$,

$$|\{\mathcal{H}.\mathsf{Hash}(k, x)\}_{x \in \{0,1\}^{n(\lambda)}}| \le 2^{n(\lambda) - \omega(\lambda)}.$$

Theorem 3.7. *There exists a (T, T', ω)-lossy* CI *hash family for $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ (Definition 3.4) assuming the existence of the following primitives:*

- *A (T', ω)-lossy trapdoor function family $\mathcal{G}$ (Definition 3.1), such that for every $\lambda \in \mathbb{N}$, $f \in \mathcal{F}_\lambda$, and $\mathsf{k} \in \mathcal{G}.\mathsf{Gen}(1^\lambda)$, the domain of $\mathcal{G}.\mathsf{Eval}(\mathsf{k}, \cdot)$ is equal to the domain of f.*
- *A T-*CI *hash family $\mathcal{H}$ (Definition 3.4) for the function family $\mathcal{F}'$, where the family $\mathcal{F}' = \{\mathcal{F}'_\lambda\}_{\lambda \in \mathbb{N}}$ is defined as follows: for each $\lambda \in \mathbb{N}$, $f' \in \mathcal{F}'_\lambda$ if and only if there exists $f \in \mathcal{F}_\lambda$, and $(\mathsf{k}, \mathsf{td}) \in \mathcal{G}.\mathsf{Gen}(1^\lambda)$ such that $f'_\lambda(\cdot) = f_\lambda(\mathcal{G}.\mathsf{Inv}(\mathsf{k}, \mathsf{td}, \cdot))$.*

Corollary 3.8. *Assuming the sub-exponential hardness of* LWE*, there exists a constant $0 < \epsilon_1 < 1$ and, for every constant $0 < \delta < 1$, a constant $0 < \epsilon_2 < 1$ such that for any polynomial S and any function family $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$ where $\mathcal{F}_\lambda$ consists of functions that are computable by circuits of size $S(\lambda)$, there exists a (T, T', ω)-lossy correlation intractable hash function family $\mathcal{H}$ for $\mathcal{F}$ (Definition 3.6). Here, $T = 2^{\lambda^{\epsilon_1}}$, $T' = 2^{\lambda^{\epsilon_2}}$, and $\omega = n - \lambda^\delta$, where n is the domain parameter associated with $\mathcal{H}$ (Definition 3.3).*

Proof. Assume the sub-exponential hardness of LWE. Then by Theorem 3.5 there exists a constant $0 < \epsilon_1 < 1$ such that for any polynomial S and any function family $\mathcal{F}' = \{\mathcal{F}'_\lambda\}_{\lambda \in \mathbb{N}}$, where $\mathcal{F}'_\lambda$ consists of functions that are computable by circuits of size $S(\lambda)$, there exists a T-CI hash family for $\mathcal{F}'$, where $T = 2^{\lambda^{\epsilon_1}}$. Moreover, by Theorem 3.2 for every constant $\delta > 0$, there exists $0 < \epsilon_2 < 1$ such that there exists a (T', ω)-lossy trapdoor family for $T'(\lambda) = 2^{\lambda^{\epsilon_2}}$ and $\omega = n - \lambda^\delta$. The corollary now follows from Theorem 3.7. □

Proof of Theorem 3.7 (Sketch). Let $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ be a function family, and let

$$\mathcal{G} = (\mathcal{G}.\mathsf{InjGen}, \mathcal{G}.\mathsf{LossyGen}, \mathcal{G}.\mathsf{Eval}, \mathcal{G}.\mathsf{Inv})$$

be a (T', ω)-lossy trapdoor family, such that for every $\lambda \in \mathbb{N}, f \in \mathcal{F}_\lambda$, and $\mathsf{k} \in \mathcal{G}.\mathsf{Gen}(1^\lambda)$, the domain of $\mathcal{G}.\mathsf{Eval}(\mathsf{k}, \cdot)$ is equal to the domain of f. Let

$$\mathcal{H} = (\mathcal{H}.\mathsf{Gen}, \mathcal{H}.\mathsf{Hash})$$

be a T-CI hash function family for function family $\mathcal{F}'$ as defined in the theorem statement.

Construction 3.9. *We construct our (T, T', ω)-secure lossy* CI *hash function family*

$$\mathcal{H}' = (\mathcal{H}'.\mathsf{Gen}, \mathcal{H}'.\mathsf{LossyGen}, \mathcal{H}'.\mathsf{Hash})$$

as follows:

- $\mathcal{H}'.\mathsf{Gen}(1^\lambda)$*:*
  - *Sample* $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$.
  - *Sample* $(\mathsf{k}, \mathsf{td}) \leftarrow \mathcal{G}.\mathsf{InjGen}(1^\lambda)$.
  - *Output* $k' = (k, \mathsf{k})$.
- $\mathcal{H}'.\mathsf{LossyGen}(1^\lambda)$*:*
  - *Sample* $k \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)$.
  - *Sample* $\mathsf{k} \leftarrow \mathcal{G}.\mathsf{LossyGen}(1^\lambda)$.
  - *Output* $k' = (k, \mathsf{k})$.
- $\mathcal{H}'.\mathsf{Hash}(k', x)$*:*
  - *Parse* $k' = (k, \mathsf{k})$.
  - *Output* $\mathcal{H}.\mathsf{Hash}(k, \mathcal{G}.\mathsf{Eval}(\mathsf{k}, x))$.

Using the properties of the lossy trapdoor family and the correlation intractable hash family, it can be shown that $\mathcal{H}'$ satisfies the properties of a $(T, T', \omega$-secure lossy CI hash family given in Definition 3.6.

□

4 SOUNDNESS OF THE FIAT SHAMIR PARADIGM

In this section we identify a class of interactive proofs for which the Fiat-Shamir transformation is sound when applied with any lossy CI hash family (as defined in Section 3). We call such interactive proofs FS-compatible, and define them formally in Section 4.1. In Section 4.3, we prove the soundness of the Fiat-Shamir transformation when applied to a FS compatible proof (w.r.t. a lossy CI hash family).

We mention that the work of [21] also identifies a class of interactive proofs for which the Fiat-Shamir paradigm can be soundly instantiated, called *instance-dependent trapdoor* Σ*-protocols*, and that their proof readily extends to any 2-round FS-compatible proof.

4.1 FS-Compatible Interactive Proofs

Loosely speaking, an interactive proof is said to be FS-compatible if it has two properties: The first is “round-by-round" soundness [21], and the second is that there exists a non-uniform advice that allows one to efficiently compute a “bad" verifier message that goes from a rejecting transcript prefix to an accepting one.

The first condition, round-by-round soundness, is defined below. We note that our notion of round-by-round soundness differs slightly from the one defined in [21].

Definition 4.1 (Round-by-Round Soundness). [21] Let $\Pi = (P, V)$ be a public-coin interactive proof system for a language L. We say that Π is (T', d)-round-by-round sound if there exists a deterministic function State that takes as input an instance x and a transcript prefix τ and outputs either accept or reject, such that the following properties hold:

(1) Let $\emptyset$ denote the empty transcript. Then for every $x \notin L$ it holds that $\mathsf{State}(x, \emptyset) = \mathsf{reject}$, and for every $x \in L$ it holds that $\mathsf{State}(x, \emptyset) = \mathsf{accept}$.
(2) For every x and transcript prefix $\tau = (\alpha_1, \beta_1, \ldots, \alpha_{i-1}, \beta_{i-1})$, if $\mathsf{State}(x, \tau) = \mathsf{reject}$, then for any (possibly unbounded) adversary $\mathcal{A}$, it holds that[18]

$$\Pr_{\substack{\alpha\leftarrow\mathcal{A}(x,\tau)\\ \beta\leftarrow\{0,1\}^\lambda}}[\mathsf{State}(x, \tau|\alpha|\beta) = \mathsf{accept}] \leq d(\lambda)\cdot 2^{-\lambda}. \qquad (2)$$

(3) For every complete transcript τ, if $\mathsf{State}(x, \tau) = \mathsf{reject}$ then $V(x, \tau) = 0$, and if $V(x, \tau) = 1$ then $\mathsf{State}(x, \tau) = \mathsf{accept}$.
(4) State is computable in time at most $T'(\lambda)$.[19]

The second condition for FS-compatibility is that the interactive proof admits an efficiently computable randomized function BAD for every round $i \in [\ell]$. This function depends, possibly inefficiently, on the instance x and all the verifier's random challenges $\beta_1, \ldots, \beta_{i-1}$ sent before round i. It obtains as input the prover messages $(\alpha_1, \ldots, \alpha_i)$ and it outputs β_i such that if $\mathsf{State}(x, \tau) = \mathsf{reject}$ for $\tau \triangleq (\alpha_1, \beta_1, \ldots, \alpha_{i-1}, \beta_{i-1})$, then β_i is a random element in the set

$$\{\beta \ : \ \mathsf{State}(x, \tau|\alpha_i|\beta) = \mathsf{accept}\}.$$

*Definition 4.2 (*FS*-Compatible Interactive Proofs).* Let T' and d be functions (not necessarily polynomial) and let ρ be a polynomial. A public-coin interactive proof (P, V) for a language L that has $\ell(\lambda)$ rounds of interaction is said to be (T', d, ρ)-FS-compatible if the following two properties hold:

(1) (P, V) is (T', d)-round-by-round sound w.r.t. a state function denoted by State (Definition 4.1).
(2) For every $x \notin L$, $i \in [\ell]$ and $\beta_1, \ldots, \beta_{i-1}$, there exists a (non-uniform[20]) randomized function BAD that takes as input $(\alpha_1, \ldots, \alpha_i)$ and randomness r, runs in time $\rho(\lambda)$, and its output satisfies the following guarantee:

   For every $(\alpha_1, \ldots, \alpha_i)$ such that $\mathsf{State}(x, \tau) = \mathsf{reject}$, for $\tau \triangleq (\alpha_1, \beta_1, \ldots, \alpha_{i-1}, \beta_{i-1})$, w.p. $1 - \mathsf{negl}(\lambda)$ (over r), $\mathsf{BAD}(\alpha_1, \ldots, \alpha_i)$ outputs a uniformly random element in the set $\mathcal{B}$, where

   $$\mathcal{B} = \{\beta \ : \ \mathsf{State}(x, \tau|\alpha_i|\beta) = \mathsf{accept}\}.$$

   If $\mathcal{B} = \emptyset$, $\mathsf{BAD}(\alpha_1, \ldots, \alpha_i)$ outputs $\perp$. Note that by Equation (2), $|\mathcal{B}| \leq d(\lambda)$.

The notion of a FS-compatible proof captures two important protocols: the sum-check and GKR prtocols. Jumping ahead, we note that generally speaking, the number of rounds $\ell(\lambda)$ in the GKR protocol can be super-polynomial, since typically $\lambda = \mathrm{polylog}(S)$, where S is the size of the computation being delegated, whereas

[18] We point out that we modify the definition in [21] to replace $\mathsf{negl}(\lambda)$ with $d(\lambda)\cdot 2^{-\lambda}$ on the right hand side of Equation (2).
[19] This requirement did not exist in the definition in [21].
[20] The non-uniformity depends on $x, \beta_1, \ldots, \beta_{i-1}$.

ℓ grows with the depth of the computation, which in general can be much larger than $\text{polylog}(S)$. Thus, it seems that there is no hope to argue that BAD is computable in polynomial time $\rho(\lambda)$ if its input length (which is of order $\ell(\lambda)$) is super-polynomial. Luckily, this is only a syntactic problem, since the function BAD corresponding to the GKR protocol does not depend on all the messages $(\alpha_1, \ldots, \alpha_i)$; rather it depends only on the last $\nu(\lambda) = \text{poly}(\lambda)$ of them. We say that the GKR protocol is *ν-history-independent*, which essentially means that each prover message is only a function of the last ν rounds of communication. We formally define history-independence below.

Definition 4.3. A protocol $\Pi = (P, V)$ is said to be *ν-history-independent* if for every $i > \nu$ and for every prefix transcript $\tau_{i-1} = (\alpha_1, \beta_1 \ldots, \alpha_{i-1}, \beta_{i-1})$, the i'th prover message α_i is computed by $\alpha_i = P(x, i, \tau_{[i-\nu, i-1]})$, where $\tau_{[i-\nu,i-1]} \triangleq (\alpha_{i-\nu}, \beta_{i-\nu}, \ldots, \alpha_{i-1}, \beta_{i-1})$.

In what follows, we define the notion of a ν-history-independent FS-compatible interactive proof. This is done by restricting the functions State and BAD to be ν-history-independent.

Definition 4.4. We say that a (T', d, ρ)-FS-compatible interactive proof is *ν-history-independent* if it is (T', d, ρ)-FS-compatible w.r.t functions State and BAD such that:

- There exists a function State$'$ computable in time $\leq T'(\lambda)$ such that for every x, every $i \in [\nu + 1, \ell]$, every $\tau_{i-1} = (\alpha_1, \beta_1, \ldots, \alpha_{i-1}, \beta_{i-1})$, it holds that
$$\text{State}(x, \tau_{i-1}) = \text{State}'(x, i, \tau_{[i-\nu,i-1]}).$$
- For every $(x, \beta_1, \ldots, \beta_{i-1})$ (the non-uniform advice of) the function BAD depends only on $(x, i, \beta_{i-\nu}, \ldots, \beta_{i-1})$, and it takes as input only $\alpha_{i-\nu}, \ldots, \alpha_i$.
Formally, for every x, $i \in [\nu+1, \ell]$ and every $(\beta_{i-\nu}, \ldots, \beta_{i-1})$ there exists a (non-uniform) randomized function BAD$'$ of size ρ, that takes as input only $\alpha_{i-\nu}, \ldots, \alpha_i$, such for every $(\beta_1, \ldots, \beta_{i-\nu-1})$, letting BAD denote the bad function that corresponds to $(x, \beta_1, \ldots, \beta_{i-1})$, for every $(\alpha_1, \ldots, \alpha_{i-\nu-1})$
$$\text{BAD}'(\alpha_{i-\nu}, \ldots, \alpha_i) = \text{BAD}(\alpha_1, \ldots, \alpha_i).$$

In the full version of our paper [47], we prove that the GKR protocol is ν-history-independent FS compatible, for $\nu(\lambda) \leq \text{poly}(\lambda)$.

Note that if we let $\nu = \ell$, any (T', d, ρ)-FS-compatible interactive proof is also ℓ-history-independent. Hence, any property of ν-history-independent FS-compatible interactive proofs holds also for plain FS-compatible interactive proofs, by setting $\nu = \ell$. In particular, in Section 4.3, we prove the soundness of the Fiat-Shamir paradigm applied to FS-compatible proofs that are ν-history independent. By setting $\nu = \ell$, soundness holds for general FS-compatible proofs as well.

4.2 Preliminaries: The Fiat-Shamir Paradigm

Let $\Pi = (P, V)$ be any public-coin interactive proof for a language L. Let $n = n(\lambda)$ denote the communication complexity of Π. Let $\mathcal{H} = (\mathcal{H}.\text{Gen}, \mathcal{H}.\text{Hash})$ be hash family such that, for every security parameter $\lambda \in \mathbb{N}$ and every $k \in \mathcal{H}.\text{Gen}(1^\lambda)$, $\mathcal{H}.\text{Hash}(k, \cdot)$ is a function with a domain $\{0,1\}^{n(\lambda)}$ and co-domain $\{0,1\}^\lambda$. We will also allow inputs to $\mathcal{H}.\text{Hash}(k, \cdot)$ that are shorter than n, by padding all inputs with 0's until the total length is n. We define the non-interactive protocol $\Pi_{\text{FS}}^{\mathcal{H}} = (P', V')$, obtained by applying the Fiat-Shamir transform to Π w.r.t. the hash family $\mathcal{H}$, in Figure 1.

The Non-Interactive Argument $\Pi_{\text{FS}}^{\mathcal{H}}$

Fix an input length $|x|$ and let $\lambda = \lambda(|x|)$.

- The common reference string CRS consists of ℓ keys $\{k_i \leftarrow \mathcal{H}.\text{Gen}(1^\lambda)\}_{i \in [\ell]}$.
- The prover P' takes as input (CRS, x) and does the following:
  1. Set $i = 1$ and $\tau_0 = \emptyset$.
  2. Compute $\alpha_i \leftarrow P(x, \tau_{i-1})$ and $\beta_i = \mathcal{H}.\text{Hash}(k_i, \tau_{i-1}|\alpha_i)$.
  3. Set $\tau_i = (\tau_{i-1}|\alpha_i|\beta_i)$.
  4. If $i = \ell$ then output τ_i. Otherwise, set $i = i + 1$ and go to Item 2.
- The verifier V' takes as input (CRS, x, τ) and does the following:
  1. Parse $\text{CRS} = (k_1, \ldots, k_\ell)$ and $\tau = (\alpha_1, \beta_1, \ldots, \alpha_\ell, \beta_\ell)$.
  2. Accept if and only if $V(x, \tau) = 1$ and for every $i \in [\ell]$ it holds that $\beta_i = \mathcal{H}.\text{Hash}(k_i, \tau_{i-1}|\alpha_i)$, where $\tau_{i-1} = (\alpha_1, \beta_1, \ldots, \alpha_{i-1}, \beta_{i-1})$.

Figure 1: The Non-Interactive Argument $\Pi_{\text{FS}}^{\mathcal{H}}$

4.2.1 *The Fiat-Shamir Paradigm for History-Independent Protocols.*

Suppose that Π is a ν-history-independent interactive proof (Definition 4.3).[21] We consider the following history-independent version of the Fiat-Shamir transformation, where we let n be the communication complexity in $\nu + 1$ rounds of the protocol Π. For every $i \in [\ell]$, rather than computing β_i as the hash of the entire transcript so far, it is computed by hashing the most recent ν rounds:
$$\beta_i = \mathcal{H}(k_i, \tau_{[i-\nu,i-1]}|\alpha_i),$$
The formal description is included in Figure 2.

4.3 The Soundness of the Fiat-Shamir Paradigm for FS-Compatible Interactive Proofs

In this section, we state our main theorem, that the (ν-history-independent) Fiat-Shamir paradigm is sound when applied to (ν-history-independent) FS-compatible interactive proofs.

Theorem 4.5. *Let $\Pi = (P, V)$ be a (T', d, ρ)-FS-compatible interactive proof (Definition 4.2) that is ν-history-independent (Definition 4.4) for some language L, where ρ is a polynomial function. Let ℓ be an arbitrary function such that Π has $\ell = \ell(\lambda)$ rounds. Denote the verifier's messages by $\beta_1, \ldots, \beta_\ell \in \{0,1\}^\lambda$, denote the overall prover runtime by $T_P(\lambda)$ and the verification time by $T_V(\lambda)$.*

Let n be an upper bound on the communication complexity of any $\nu + 1$ rounds of Π. Suppose that there exists a (T, T', ω)-lossy CI hash function family $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda \in \mathbb{N}}$ (Definition 3.6) for the family $\mathcal{F}$ consisting of all functions computable by circuits of size $\rho(\lambda)$. Moreover, suppose that the functions in $\mathcal{H}_\lambda$ map inputs in $\{0,1\}^{n(\lambda)}$ to outputs in $\{0,1\}^\lambda$, and that $T'(\lambda) \geq \ell(\lambda)$ and $T \geq \max\{d, 2^{\nu(n-\omega)}, T'\}$.

[21] Note that every Π is ν-history-independent for some $\nu \leq \ell - 1$.

ν-History-Independent Version of $\Pi_{\mathsf{FS}}^{\mathcal{H}}$

Fix an input length $|x|$ and let $\lambda = \lambda(|x|)$.

- The common reference string CRS consists of ℓ keys $\{k_i \leftarrow \mathcal{H}.\mathsf{Gen}(1^\lambda)\}_{i \in [\ell]}$.
- The prover P' takes as input (CRS, x) and does the following:
  1. Set $i = 1$ and $\tau_0 = \emptyset$.
  2. Compute $\alpha_i \leftarrow P(x, \tau_{[i-\nu, i-1]})$ and $\beta_i = \mathcal{H}.\mathsf{Hash}(k_i, \tau_{[i-\nu,i-1]}|\alpha_i)$.
  3. Set $\tau_i = (\tau_{i-1}|\alpha_i|\beta_i)$.
  4. If $i = \ell$ then output τ_i. Otherwise, set $i = i + 1$ and go to Item 2.
- The verifier V' takes as input (CRS, x, τ) and does the following:
  1. Parse $\mathsf{CRS} = (k_1, \ldots, k_\ell)$ and $\tau = (\alpha_1, \beta_1, \ldots, \alpha_\ell, \beta_\ell)$.
  2. Accept if and only if $V(x, \tau) = 1$ and for every $i \in [\ell]$ it holds that $\beta_i = \mathcal{H}.\mathsf{Hash}(k_i, \tau_{[i-\nu,i-1]}|\alpha_i)$.

Figure 2: The ν-History-Independent Version of $\Pi_{\mathsf{FS}}^{\mathcal{H}}$

Then the resulting non-interactive protocol $\Pi_{\mathsf{FS}}^{\mathcal{H}}$, obtained by applying the ν-history-independent Fiat-Shamir transform to Π w.r.t. the hash family $\mathcal{H}$ (Figure 2), has the following properties:

- **Completeness.** *If Π has completeness 1, then $\Pi_{\mathsf{FS}}^{\mathcal{H}}$ also has completeness 1.*
- **T'-Soundness.** *For any $\mathsf{poly}(T')$-size cheating prover P^*, there exists a negligible function μ such that for every $x^* \notin L$ and $\lambda = \lambda(|x^*|)$,*
$$\Pr_{\substack{\mathsf{CRS} \leftarrow \mathsf{Setup}(1^\lambda) \\ \tau^* \leftarrow P^*(\mathsf{CRS})}} [(V_{\mathsf{FS}}^{\mathcal{H}}(\mathsf{CRS}, x^*, \tau^*) = 1)] = \mu(T'(\lambda)).$$
- **Efficiency.** *There exists a polynomial p that depends on the lossy CI hash family $\mathcal{H}$ such that the total verifier runtime is $\ell(\lambda) \cdot p(\rho(\lambda)) + T_V(\lambda)$ and the total prover runtime is $\ell(\lambda) \cdot p(\rho(\lambda)) + T_P(\lambda)$.*

In the full version of this paper [47], we show that the sum-check and GKR protocols are FS-compatible and ν-FS-compatible, for some polynomial ν independent of the depth of the circuit being delegated. It follows as corollaries that:

Corollary 4.6. *(Informal) Assuming the sub-exponential hardness of* LWE*, there exists a hash family $\mathcal{H}$ such that the non-interactive argument obtained by applying the Fiat-Shamir transform to the sum-check protocol, with the hash family $\mathcal{H}$, is sound assuming the field size in the sum-check protocol instance is large enough.*

Corollary 4.7. *Assuming the sub-exponential hardness of* LWE*, for any log-space uniform $C : \{0,1\}^N \to \{0,1\}$ of size S and depth D, there is a non-interactive argument for the language $\{x : C(x) = 0\}$. This non-interactive argument has the following efficiency guarantees: the prover runs in time $\mathsf{poly}(S)$, the verifier runs in time $(D + N) \cdot \mathsf{polylog}(S)$, and the communication complexity is $D \cdot \mathsf{polylog}(S)$.*

ACKNOWLEDGEMENTS

This material is based on work supported in part by DARPA under Contract Nos. HR001120C0023 and HR001120C0024. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA. Rachel Zhang is supported by the Paul E. Gray (1954) UROP Fund.

We additionally thank Alex Lombardi, Pavel Hubâček and Giulio Malavolta for insightful discussions.

REFERENCES

[1] Tim Abbot, Daniel Kane, and Paul Valiant. 2004. On algorithms for Nash equilibria. *Unpublished manuscript* (2004). https://web.mit.edu/tabbott/Public/final.pdf.

[2] Prabhanjan Ananth, Yu-Chi Chen, Kai-Min Chung, Huijia Lin, and Wei-Kai Lin. 2016. Delegating RAM Computations with Adaptive Soundness and Privacy. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II.* 3–30. https://doi.org/10.1007/978-3-662-53644-5_1

[3] Saikrishna Badrinarayanan, Yael Tauman Kalai, Dakshita Khurana, Amit Sahai, and Daniel Wichs. 2018. Succinct delegation for low-space non-deterministic computation. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018.* 709–721. https://doi.org/10.1145/3188745.3188924

[4] Boaz Barak. 2001. How to Go Beyond the Black-Box Simulation Barrier. In *FOCS*. 106–115.

[5] James Bartusek, Liron Bronfman, Justin Holmgren, Fermi Ma, and Ron D. Rothblum. 2019. On the (In)security of Kilian-Based SNARGs. In *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 11892)*, Dennis Hofheinz and Alon Rosen (Eds.). Springer, 522–551. https://doi.org/10.1007/978-3-030-36033-7_20

[6] Mihir Bellare and Phillip Rogaway. 1993. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *ACM Conference on Computer and Communications Security*, Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby (Eds.). ACM, 62–73.

[7] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. 2016. Interactive Oracle Proofs. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 9986)*, Martin Hirt and Adam D. Smith (Eds.). 31–60. https://doi.org/10.1007/978-3-662-53644-5_2

[8] Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Aviad Rubinstein, and Eran Tromer. 2014. The Hunting of the SNARK. *IACR Cryptology ePrint Archive* 2014 (2014), 580. http://eprint.iacr.org/2014/580

[9] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. 2013. Recursive composition and bootstrapping for SNARKS and proof-carrying data. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013.* 111–120. https://doi.org/10.1145/2488608.2488623

[10] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. 2013. Succinct Non-interactive Arguments via Linear Interactive Proofs. In *TCC*. 315–333. https://doi.org/10.1007/978-3-642-36594-2_18

[11] Nir Bitansky, Sanjam Garg, Huijia Lin, Rafael Pass, and Sidharth Telang. 2015. Succinct Randomized Encodings and their Applications. *IACR Cryptology ePrint Archive* 2015 (2015), 356.

[12] Nir Bitansky and Idan Gerichter. 2020. On the Cryptographic Hardness of Local Search. In *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA (LIPIcs, Vol. 151)*, Thomas Vidick (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 6:1–6:29. https://doi.org/10.4230/LIPIcs.ITCS.2020.6

[13] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. 2018. Multi-collision resistance: a paradigm for keyless hash functions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, Ilias Diakonikolas, David Kempe, and Monika Henzinger (Eds.). ACM, 671–684. https://doi.org/10.1145/3188745.3188870

[14] Nir Bitansky, Omer Paneth, and Alon Rosen. 2015. On the Cryptographic Hardness of Finding a Nash Equilibrium. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, Venkatesan Guruswami (Ed.). IEEE Computer Society, 1480–1498. https://doi.org/10.1109/FOCS.2015.94

[15] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. 2019. Leveraging Linear Decryption: Rate-1 Fully-Homomorphic Encryption and Time-Lock Puzzles. In *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II.* 407–437. https://doi.org/10.1007/978-3-030-36033-7_16

[16] Zvika Brakerski, Nico Döttling, Sanjam Garg, and Giulio Malavolta. 2020. Factoring and Pairings are not Necessary for iO: Circular-Secure LWE Suffices. *IACR Cryptol. ePrint Arch.* (2020). https://eprint.iacr.org/2020/1024

[17] Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. 2017. Non-interactive delegation and batch NP verification from standard computational assumptions. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*. 474–482. https://doi.org/10.1145/3055399.3055497

[18] Zvika Brakerski and Yael Kalai. 2020. Witness Indistinguishability for Any Single-Round Argument with Applications to Access Control. In *Public-Key Cryptography - PKC 2020 - 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, May 4-7, 2020, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12111)*, Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas (Eds.). Springer, 97–123. https://doi.org/10.1007/978-3-030-45388-6_4

[19] Zvika Brakerski, Venkata Koppula, and Tamer Mour. 2020. NIZK from LPN and Trapdoor Hash via Correlation Intractability for Approximable Relations. In *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 12172)*, Daniele Micciancio and Thomas Ristenpart (Eds.). Springer, 738–767. https://doi.org/10.1007/978-3-030-56877-1_26

[20] Gilles Brassard, David Chaum, and Claude Crépeau. 1988. Minimum Disclosure Proofs of Knowledge. *J. Comput. Syst. Sci.* 37, 2 (1988), 156–189.

[21] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. 2019. Fiat-Shamir: from practice to theory. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, Moses Charikar and Edith Cohen (Eds.). ACM, 1082–1090. https://doi.org/10.1145/3313276.3316380

[22] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. 2018. Fiat-Shamir and Correlation Intractability from Strong KDM-Secure Encryption. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I*. 91–122. https://doi.org/10.1007/978-3-319-78381-9_4

[23] Ran Canetti, Oded Goldreich, and Shai Halevi. 2004. The random oracle methodology, revisited. *J. ACM* 51, 4 (2004), 557–594.

[24] Ran Canetti and Justin Holmgren. 2016. Fully Succinct Garbled RAM. In *ITCS*. ACM, 169–178.

[25] Ran Canetti, Justin Holmgren, Abhishek Jain, and Vinod Vaikuntanathan. 2015. Succinct Garbling and Indistinguishability Obfuscation for RAM Programs. In *STOC*. ACM, 429–437.

[26] David G. Cantor and Hans Zassenhaus. 1981. A new algorithm for factoring polynomials over finite fields. *Mathematics of Compuation* (1981), 587–592.

[27] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. 2009. Settling the complexity of computing two-player Nash equilibria. *J. ACM* 56, 3 (2009), 14:1–14:57. https://doi.org/10.1145/1516512.1516516

[28] Yu-Chi Chen, Sherman S. M. Chow, Kai-Min Chung, Russell W. F. Lai, Wei-Kai Lin, and Hong-Sheng Zhou. 2016. Cryptography for Parallel RAM from Indistinguishability Obfuscation. In *ITCS*. ACM, 179–190.

[29] Arka Rai Choudhuri, Pavel Hubácek, Chethan Kamath, Krzysztof Pietrzak, Alon Rosen, and Guy N. Rothblum. 2019. Finding a Nash equilibrium is no easier than breaking Fiat-Shamir. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, Moses Charikar and Edith Cohen (Eds.). ACM, 1103–1114. https://doi.org/10.1145/3313276.3316400

[30] Arka Rai Choudhuri, Pavel Hubácek, Chethan Kamath, Krzysztof Pietrzak, Alon Rosen, and Guy N. Rothblum. 2019. PPAD-Hardness via Iterated Squaring Modulo a Composite. *IACR Cryptol. ePrint Arch.* 2019 (2019), 667. https://eprint.iacr.org/2019/667

[31] Ivan Damgård. 1992. Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks. In *Proceedings of CRYPTOï¿œ91*. 445–456.

[32] Ivan Damgård, Sebastian Faust, and Carmit Hazay. 2012. Secure Two-Party Computation with Low Communication. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*. 54–74. https://doi.org/10.1007/978-3-642-28914-9_4

[33] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. 2009. The Complexity of Computing a Nash Equilibrium. *SIAM J. Comput.* 39, 1 (2009), 195–259. https://doi.org/10.1137/070699652

[34] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. 2019. Trapdoor Hash Functions and Their Applications. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 11694)*, Alexandra Boldyreva and Daniele Micciancio (Eds.). Springer, 3–32. https://doi.org/10.1007/978-3-030-26954-8_1

[35] Naomi Ephraim, Cody Freitag, Ilan Komargodski, and Rafael Pass. 2020. Continuous Verifiable Delay Functions. In *Advances in Cryptology - EUROCRYPT 2020 - 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10-14, 2020, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 12107)*, Anne Canteaut and Yuval Ishai (Eds.). Springer, 125–154. https://doi.org/10.1007/978-3-030-45727-3_5

[36] Amos Fiat and Adi Shamir. 1986. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO*. 186–194.

[37] Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. 2016. Revisiting the Cryptographic Hardness of Finding a Nash Equilibrium. In *Advances in Cryptology - CRYPTO 2016, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 9815)*, Matthew Robshaw and Jonathan Katz (Eds.). Springer, 579–604. https://doi.org/10.1007/978-3-662-53008-5_20

[38] Romain Gay and Rafael Pass. 2020. Indistinguishability Obfuscation from Circular Security. *IACR Cryptol. ePrint Arch.* (2020). https://eprint.iacr.org/2020/1010

[39] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. 2013. Quadratic Span Programs and Succinct NIZKs without PCPs. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*. 626–645. https://doi.org/10.1007/978-3-642-38348-9_37

[40] Shafi Goldwasser and Yael Tauman Kalai. 2003. On the (In)security of the Fiat-Shamir Paradigm. In *FOCS*. 102–.

[41] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. 2015. Delegating Computation: Interactive Proofs for Muggles. *J. ACM* 62, 4 (2015), 27.

[42] Jens Groth. 2010. Short Pairing-Based Non-interactive Zero-Knowledge Arguments. In *ASIACRYPT (Lecture Notes in Computer Science, Vol. 6477)*. Springer, 321–340.

[43] Brett Hemenway and Rafail Ostrovsky. 2012. Extended-DDH and Lossy Trapdoor Functions. In *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings (Lecture Notes in Computer Science, Vol. 7293)*, Marc Fischlin, Johannes A. Buchmann, and Mark Manulis (Eds.). Springer, 627–643. https://doi.org/10.1007/978-3-642-30057-8_37

[44] Justin Holmgren and Alex Lombardi. 2018. Cryptographic Hashing from Strong One-Way Functions (Or: One-Way Product Functions and Their Applications). In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, Mikkel Thorup (Ed.). IEEE Computer Society, 850–858. https://doi.org/10.1109/FOCS.2018.00085

[45] Pavel Hubácek and Eylon Yogev. 2017. Hardness of Continuous Local Search: Query Complexity and Cryptographic Lower Bounds. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, Philip N. Klein (Ed.). SIAM, 1352–1371. https://doi.org/10.1137/1.9781611974782.88

[46] Aayush Jain, Huijia Lin, and Amit Sahai. 2020. Indistinguishability Obfuscation from Well-Founded Assumptions. Cryptology ePrint Archive, Report 2020/1003. https://eprint.iacr.org/2020/1003.

[47] Ruta Jawale, Yael Tauman Kalai, Dakshita Khurana, and Rachel Zhang. 2020. SNARGs for Bounded Depth Computations and PPAD Hardness from Sub-Exponential LWE. *IACR Cryptol. ePrint Arch.* (2020). https://eprint.iacr.org/2020/980.pdf

[48] Yael Tauman Kalai and Omer Paneth. 2016. Delegating RAM Computations. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part II*. 91–118. https://doi.org/10.1007/978-3-662-53644-5_4

[49] Yael Tauman Kalai, Omer Paneth, and Lisa Yang. 2019. How to delegate computations publicly. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, Moses Charikar and Edith Cohen (Eds.). ACM, 1115–1124. https://doi.org/10.1145/3313276.3316411

[50] Yael Tauman Kalai, Omer Paneth, and Lisa Yang. 2020. PPAD-Hardness and Delegation with Unambiguous Proofs. CRYPTO.

[51] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. 2013. Delegation for bounded space. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*. 565–574. https://doi.org/10.1145/2488608.2488679

[52] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. 2014. How to delegate computations: the power of no-signaling proofs. In *STOC*. ACM, 485–494.

[53] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. 2017. From Obfuscation to the Security of Fiat-Shamir for Proofs. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 10402)*, Jonathan Katz and Hovav Shacham (Eds.). Springer, 224–251. https://doi.org/10.1007/978-3-319-63715-0_8

[54] Joe Kilian. 1992. A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing*. ACM, 723–732.

[55] Venkata Koppula, Allison Bishop Lewko, and Brent Waters. 2015. Indistinguishability Obfuscation for Turing Machines with Unbounded Memory. In *STOC*. ACM, 419–428.

[56] Helger Lipmaa. 2012. Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments. In *TCC*. 169–189.

[57] Alex Lombardi and Vinod Vaikuntanathan. 2020. Fiat-Shamir for Repeated Squaring with Applications to PPAD-Hardness and VDFs. Cryptology ePrint Archive, Report 2020/772. https://eprint.iacr.org/2020/772.

[58] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. 1992. Algebraic Methods for Interactive Proof Systems. *J. ACM* 39, 4 (1992), 859–868.

[59] Silvio Micali. 1994. CS Proofs (Extended Abstracts). In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*. 436–453. https://doi.org/10.1109/SFCS.1994.365746 Full version in [60].

[60] Silvio Micali. 2000. Computationally Sound Proofs. *SIAM J. Comput.* 30, 4 (2000), 1253–1298.

[61] Moni Naor. 2003. On Cryptographic Assumptions and Challenges. In *Proceedings of the 23rd Annual International Cryptology Conference*. 96–109.

[62] Omer Paneth and Guy N. Rothblum. 2017. On Zero-Testable Homomorphic Encryption and Publicly Verifiable Non-interactive Arguments. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II*. 283–315. https://doi.org/10.1007/978-3-319-70503-3_9

[63] Christos H. Papadimitriou. 1994. On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence. *J. Comput. Syst. Sci.* 48, 3 (1994), 498–532. https://doi.org/10.1016/S0022-0000(05)80063-7

[64] Bryan Parno, Mariana Raykova, and Vinod Vaikuntanathan. 2012. How to Delegate and Verify in Public: Verifiable Computation from Attribute-Based Encryption. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*. 422–439. https://doi.org/10.1007/978-3-642-28914-9_24

[65] Chris Peikert and Sina Shiehian. 2019. Noninteractive Zero Knowledge for NP from (Plain) Learning with Errors. In *Advances in Cryptology - CRYPTO 2019, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 11692)*, Alexandra Boldyreva and Daniele Micciancio (Eds.). Springer, 89–114. https://doi.org/10.1007/978-3-030-26948-7_4

[66] Chris Peikert and Brent Waters. 2008. Lossy trapdoor functions and their applications. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, Cynthia Dwork (Ed.). ACM, 187–196. https://doi.org/10.1145/1374376.1374406

[67] Krzysztof Pietrzak. 2019. Simple Verifiable Delay Functions. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA (LIPIcs, Vol. 124)*, Avrim Blum (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 60:1–60:15. https://doi.org/10.4230/LIPIcs.ITCS.2019.60

[68] David Pointcheval and Jacques Stern. 1996. Security Proofs for Signature Schemes. In *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding (Lecture Notes in Computer Science, Vol. 1070)*, Ueli M. Maurer (Ed.). Springer, 387–398. https://doi.org/10.1007/3-540-68339-9_33

[69] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. 2016. Constant-round interactive proofs for delegating computation. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*. 49–62. https://doi.org/10.1145/2897518.2897652

[70] Adi Shamir. 1992. IP = PSPACE. *J. ACM* 39, 4 (1992), 869–877.

[71] Riad S. Wahby, Ioanna Tzialla, Abhi Shelat, Justin Thaler, and Michael Walfish. 2018. Doubly-Efficient zkSNARKs Without Trusted Setup. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*. IEEE Computer Society, 926–943. https://doi.org/10.1109/SP.2018.00060

[72] Hoeteck Wee and Daniel Wichs. 2020. Candidate Obfuscation via Oblivious LWE Sampling. *IACR Cryptol. ePrint Arch.* (2020). https://eprint.iacr.org/2020/1042