Fiat-Shamir for Bounded-Depth Adversaries

Liyan Chen
MIT, Cambridge, MA, USA

Yilei Chen
Tsinghua University, Beijing, China
Shanghai Qi Zhi Institute, China

Zikuan Huang
Shanghai Qi Zhi Institute, China

Nuozhou Sun
Carnegie Mellon University, Pittsburgh, PA, USA

Tianqi Yang
Columbia University, New York, NY, USA

Yiding Zhang
Boston University, MA, USA

Abstract

We study how to construct hash functions that can securely instantiate the Fiat-Shamir transformation against *bounded-depth* adversaries. The motivation is twofold. First, given the recent fruitful line of research of constructing cryptographic primitives against bounded-depth adversaries under *worst-case complexity assumptions*, and the rich applications of Fiat-Shamir, instantiating Fiat-Shamir hash functions against bounded-depth adversaries under worst-case complexity assumptions might lead to further applications (such as SNARG for P, showing the cryptographic hardness of PPAD, etc.) against bounded-depth adversaries. Second, we wonder whether it is possible to overcome the impossibility results of constructing Fiat-Shamir for arguments [Goldwasser, Kalai, FOCS '03] in the setting where the depth of the adversary is bounded, given that the known impossibility results (against p.p.t. adversaries) are contrived.

Our main results give new insights for Fiat-Shamir against bounded-depth adversaries in both the positive and negative directions. On the positive side, for Fiat-Shamir for proofs with certain properties, we show that weak worst-case assumptions are enough for constructing explicit hash functions that give $\mathsf{AC}^0[2]$-soundness. In particular, we construct an $\mathsf{AC}^0[2]$-computable correlation-intractable hash family for constant-degree polynomials against $\mathsf{AC}^0[2]$ adversaries, assuming $\oplus\mathsf{L}/\mathsf{poly} \not\subseteq \widetilde{\mathsf{Sum}}_{n^{-c}} \circ \mathsf{AC}^0[2]$ for some $c > 0$. This is incomparable to all currently-known constructions, which are typically useful for larger classes and against stronger adversaries, but based on arguably stronger assumptions. Our construction is inspired by the Fiat-Shamir hash function by Peikert and Shiehian [CRYPTO '19] and the fully-homomorphic encryption scheme against bounded-depth adversaries by Wang and Pan [EUROCRYPT '22].

On the negative side, we show Fiat-Shamir for arguments is still impossible to achieve against bounded-depth adversaries. In particular,

- Assuming the existence of $\mathsf{AC}^0[2]$-computable CRHF against p.p.t. adversaries, for every poly-size hash function, there is a (p.p.t.-sound) interactive argument that is not $\mathsf{AC}^0[2]$-sound after applying Fiat-Shamir with this hash function.
- Assuming the existence of $\mathsf{AC}^0[2]$-computable CRHF against $\mathsf{AC}^0[2]$ adversaries, there is an $\mathsf{AC}^0[2]$-sound interactive argument such that for every hash function computable by $\mathsf{AC}^0[2]$ circuits, the argument does not preserve $\mathsf{AC}^0[2]$-soundness when applying Fiat-Shamir with this hash function. This is a low-depth variant of Goldwasser and Kalai.

2012 ACM Subject Classification Security and privacy → Hash functions and message authentication codes

Keywords and phrases Fiat-Shamir, Correlation Intractability

7th Conference on Information-Theoretic Cryptography (ITC 2026).
Editor: Yevgeniy Dodis; Article No. 4; pp. 4:1–4:22

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Digital Object Identifier 10.4230/LIPIcs.ITC.2026.4

Related Version *Full Version*: https://eprint.iacr.org/2024/256 [13]

Funding *Yilei Chen*: Research supported by Tsinghua University startup funding, and Shanghai Qi Zhi Institute Innovation Program.
Tianqi Yang: Research supported in part by NSF Grant CCF-2106429, CCF-2211238, CCF-2212136, CCF-2312224, an Amazon Research Award, and Google CyberNYC.

1 Introduction

The Fiat-Shamir transform [23] is a generic method for converting public-coin interactive protocols into non-interactive ones while preserving the original protocol's functionality. First proposed as a practical method to compile identification schemes into digital signatures, the Fiat-Shamir transform was then realized to be an extremely general technique for minimizing interaction in any public-coin protocol. The basic idea of Fiat-Shamir is to replace the verifier's message in each round (which consists of random coin tosses) with a deterministic hash of the protocol transcript so far. As a typical example, consider a three-round public-coin interactive proof/argument system (P, V) for a language L. When proving a statement $x \in L$, the prover P first sends a message α, the verifier V responds with random coins β, and then P sends the last message γ. To convert the interactive protocol (P, V) into a non-interactive protocol (P^{FS}, V^{FS}), the Fiat-Shamir transform uses a hash function h (randomly chosen from a hash family $\mathcal{H}$) and removes the verifier's coin tosses β by setting $\beta = h(x\|\alpha)$[1]. Then P^{FS} can directly compute β and send $(\alpha, \beta = h(x\|\alpha), \gamma)$ in one shot without any interaction (actually β can be omitted since it is clear from x, α and the public hash function h). See Figure 1 below for a visual explanation.

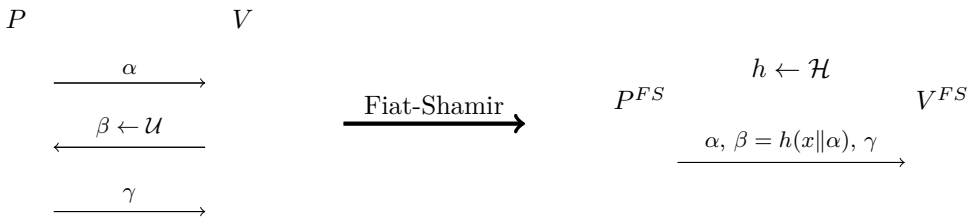

Figure 1 Instantiating Fiat-Shamir on (P, V) for proving $x \in L$.

1.1 The soundness of Fiat-Shamir

The central problem of Fiat-Shamir is its soundness – justifying whether and when the Fiat-Shamir transform results in a sound non-interactive protocol has long been a focus of research in cryptography. It was shown by Pointcheval and Stern [41] that the soundness of Fiat-Shamir holds in the *random oracle model* (ROM), i.e., the soundness of any constant-round public-coin protocol can be preserved if we model the Fiat-Shamir hash function as a random oracle. However, it is more desirable to prove the soundness of Fiat-Shamir

[1] "‖" means the concatenation of two strings. Here we use $h(x\|\alpha)$ but not $h(x, \alpha)$ to emphasize that h takes as input the entire transcript till now (consisting of x and α) instead of taking x and α separately as two inputs.

transformation in the *plain model*: the Fiat-Shamir hash should be instantiated by an explicit hash function in contrast to an idealized one, and the security in the random oracle model cannot provide any guarantee here.

In the plain model, the difference between *proofs* and *arguments* becomes crucial. By using proofs, we mean protocols that are sound against *any* cheating provers (can be computationally unbounded), while by using arguments, we mean protocols that are sound against only computationally bounded (probabilistic polynomial-time) cheating provers. We summarize the main positive results here.

- **Fiat-Shamir for proofs.** If a hash function is correlation intractable (CI) then it can securely instantiate Fiat-Shamir for proofs [11]. In the past few years there has been a long line of work on correlation intractable (CI) hash functions [10, 34, 11, 27, 9, 40, 7, 28, 31, 38, 15, 19] showing that we can securely instantiate Fiat-Shamir on interactive proofs of which the "bad relation" (which is roughly the messages (α, β) that may allow the prover to cheat) meets some special constraints. CI hash functions have already been used for applying Fiat-Shamir to some practical interactive proofs, e.g., the sumcheck protocol and the GKR protocol [32], and also for constructing non-interactive cryptographic primitives from standard cryptographic assumptions, e.g., non-interactive zero-knowledge (NIZK) [40, 7, 28, 31, 19] and succinct non-interactive arguments (SNARGs) [32, 17, 18, 29, 33, 15].
- **Fiat-Shamir for arguments.** In the context of arguments, however, the soundness of Fiat-Shamir is still poorly understood. In some applications of Fiat-Shamir, we do know how to instantiate Fiat-Shamir securely on some special arguments using just CI hash functions (see, e.g., the SNARG constructions from [35, 17, 18, 29, 15]). However, the security mainly comes from the statistical property in their soundness such that these arguments can somehow be regarded as proofs – actually we are still using the framework for proofs. Outside the scope of "proof-like" arguments, there are only few positive results about some specific protocols, e.g., Fiat-Shamir for Schnorr's or Lyubashevsky's identification schemes [14].

As we can see, positive results showing the soundness of Fiat-Shamir in the plain model concentrate almost all on proofs, while the soundness of Fiat-Shamir for arguments remains largely open (instantiating Fiat-Shamir soundly on proofs is intuitively easier since proofs have stronger security guarantees than arguments). Furthermore, there have been impossibility results or strong negative indications on the soundness of Fiat-Shamir for arguments in the plain model. Goldwasser and Kalai [26] constructed a set of three (contrived) arguments such that Fiat-Shamir using *any* explicit hash function must be insecure on at least one of them. In addition, Dwork et al. [21] showed that (constant-round, public-coin) zero-knowledge arguments can also serve as counter-examples for Fiat-Shamir[2], and constructions of such protocols are known based on various cryptographic assumptions (see, e.g., [3, 6, 37]).

1.2 Fiat-Shamir and cryptography against bounded-depth adversaries

In this work, we study the soundness of Fiat-Shamir in a new setting where the adversaries' *parallel time* is further limited. In contrast to the standard security notion in cryptography that the adversaries run in probabilistic polynomial time (p.p.t.) with no further limitation, we consider a relaxed version of security that the adversaries are from *bounded-depth* (and

[2] The intuition of [21] is: we can construct a cheating verifier that sends messages according to the Fiat-Shamir hash function, and the zero-knowledge property guarantees that a simulator can efficiently simulate the view of such a cheating verifier. If the protocol is sound after Fiat-Shamir, the simulator can then be used to decide the language. Therefore, Fiat-Shamir can only be sound when the zero-knowledge protocol proves a language in BPP.

thus bounded parallel time) circuit classes like $\mathsf{AC}^0[2]$ (poly-size constant-depth circuits with NOT gates and unbounded fan-in OR, AND, and MOD_2 gates) and TC^0 (poly-size constant-depth circuits with NOT gates and unbounded fan-in OR, AND, and threshold (THR) gates, where a threshold gate returns 1 if at least half of its inputs are 1, and 0 otherwise). If we can get positive results for Fiat-Shamir against bounded-depth adversaries, it will be very exciting: given the rich applications of (or inspired by) Fiat-Shamir in recent years (such as SNARGs for P, showing the cryptographic hardness of PPAD [16], etc.), instantiating Fiat-Shamir hash functions against bounded-depth adversaries might lead to further applications against bounded-depth adversaries. Furthermore, given the strong negative indications and the extreme lack of positive results on Fiat-Shamir for arguments against p.p.t adversaries, we would like to see if it is possible to bypass the known impossibility results in the bounded-depth setting.

Recently there has been a fruitful sequence of work on constructing cryptographic primitives with security against only bounded-depth adversaries[3], including one-way functions, pseudorandom generators, collision-resistant hash functions [20], one-way permutations [22], fully homomorphic encryption schemes [8, 42], and non-interactive zero-knowledge [42]. Compared to their analogs with the standard p.p.t. security, these primitives are indeed less secure but can rely only on worst-case complexity assumptions or even be unconditional (see, e.g., the construction of CRHF against AC^0 in [20], where AC^0 is the class of poly-size constant-depth circuits with NOT gates and unbounded fan-in OR and AND gates). Here we continue this line of work with a specific focus on Fiat-Shamir in the same setting: the soundness is rather weak (but still non-trivial) against circuit classes as low as $\mathsf{AC}^0[2]$, while the underlying assumption is likewise weak. The choice of bounded-depth circuit classes is also motivated by the developments in circuit complexity, where strong lower bounds for bounded-depth classes are known. Therefore, it is very likely that we can translate these lower bounds into good hash constructions that are only useful for Fiat-Shamir against the same bounded-depth classes.

Also note that the lowest circuit class we care about is $\mathsf{AC}^0[2]$, which stands for poly-size constant-depth circuits with NOT gates and OR, AND, MOD_2 gates with unbounded fan-in. We choose $\mathsf{AC}^0[2]$ because our results (as shown in the next section) can indeed work for classes as low as $\mathsf{AC}^0[2]$, and $\mathsf{AC}^0[2]$ is arguably the smallest circuit class that makes Fiat-Shamir reasonable. In cryptography, almost all hash functions (and other cryptographic primitives) use some kind of linearity or algebra over finite fields, and constructions without the use of parity or modulo gates are rare. To the best of our knowledge, there are only two exceptions about classes lower than $\mathsf{AC}^0[2]$. The first one is the cryptography in AC^0 by Degwekar et al.[20], but their construction is still inherently using parity, except they use some clever tricks to bound the number of bits in the parity by $\mathsf{polylog}$. The other one is the cryptography in NC^0 by Applebaum et al. [2] (where NC^0 is the same as AC^0 except that OR and AND gates have only bounded fan-in), but their construction is not robust enough, e.g., their PRG can only have additive stretch.

1.3 Our results

Our results include two independent parts that make progress on both the positive side and the negative side. On the positive side, we give new constructions of Fiat-Shamir hash functions for proofs in the bounded-depth setting. Inspired by the correlation intractable hash

[3] This research topic is also referred to as "fine-grained cryptography" like in [20]. We do not use the notion "fine-grained" because in complexity theory it usually refers to unrobust classes like $\mathsf{DTIME}[n^2]$. We use the name "cryptography against bounded-depth adversaries" to avoid any possible ambiguity.

by Peikert and Shiehian [40] and the FHE with bounded-depth security by Wang and Pan [42], we construct a new family of correlation intractable hash functions against bounded-depth adversaries. It allows us to generally instantiate Fiat-Shamir, at least on a class of interactive proofs, to obtain non-interactive cryptographic primitives in the bounded-depth setting. On the negative side, unfortunately, we observe that our positive results on proofs are almost the best we can do in the bounded-depth setting. In particular, we show that the known impossibility results for Fiat-Shamir for arguments like Goldwasser and Kalai [26] can be extended to the bounded-depth setting.

1.3.1 Positive results

We first present the positive results for Fiat-Shamir for *proofs* against bounded-depth cheating provers. In particular, we show new constructions of correlation-intractable (CI) hash functions for relations representable by constant-degree polynomials that are secure against bounded-depth adversaries.

Our construction is inspired by the CI hash function constructed by Peikert and Shiehian [40], which uses the FHE scheme of Gentry, Sahai, Waters [24] (henceforth GSW-FHE) as a building block. Our main observation is that the FHE scheme against low-depth adversaries constructed by Wang and Pan [42] bears a striking similarity with GSW-FHE. The similarity allows us to replace GSW-FHE with WP-FHE to obtain a CI hash function against low-depth adversaries assuming worst-case complexity assumptions. Our main positive result is:

▶ **Theorem 1** (Informally stated, see Theorem 28)**.** *Assuming* $\mathsf{NC}^1 \neq \oplus\mathsf{L}/\mathsf{poly}$*, there exists a CI hash function family for constant-degree polynomials against* NC^1 *that is computable in* $\mathsf{AC}^0[2]$.

The circuit class NC^1 (which stands for poly-size $O(\log n)$-depth circuits with bounded fan-in NOT, OR, and AND gates) is just a typical example and can actually be replaced with lower circuit classes. See Section 3.3 for more details. The worst-case complexity assumption is a standard assumption in the area of cryptography against bounded-depth adversaries: previous constructions of, e.g., one-way functions, pseudorandom generators, collision-resistant hash functions [20] are all based on the same assumption.

1.3.2 Negative results

On the negative side, we get two (incomparable) impossibility results in the bounded-depth setting. The first one shows the non-existence of a *universal* Fiat-Shamir hash against bounded-depth adversaries; while the second one extends the work by Goldwasser and Kalai [26] showing the existence of a protocol that makes every Fiat-Shamir hash fail even in the bounded-depth setting.

$\mathcal{C}$-soundness and $\mathcal{C}$-arguments. For simplicity, we say a protocol has $\mathcal{C}$-soundness if it is sound against adversaries computable in the complexity class $\mathcal{C}$. Also, we use the notation $\mathcal{C}$-arguments for protocols with $\mathcal{C}$-soundness. Typical choices of $\mathcal{C}$ include bounded-depth circuit classes $\mathsf{AC}^0[2]$, TC^0, or NC^1. When just saying an argument, we refer to a P/poly-argument.

We first (informally) define what a universal Fiat-Shamir hash is in the bounded-depth setting.

▶ **Definition 2** (Universal Fiat-Shamir hash, informal). *For circuit classes $\mathcal{D} \subseteq \mathcal{C}$, we say a hash function family $\mathcal{H}$ is a universal Fiat-Shamir hash from $\mathcal{C}$ to $\mathcal{D}$ if Fiat-Shamir using $\mathcal{H}$ always results in a sound non-interactive $\mathcal{D}$-argument starting from any (3-round public-coin) interactive $\mathcal{C}$-argument that matches the input and output lengths of $\mathcal{H}$ (i.e., the protocol's instance + first message length should be $\mathcal{H}$'s input length, and the protocol's second message length should be $\mathcal{H}$'s output length).*

Note that Definition 2 is already enough to capture a large class of hash functions, which is a fixed hash function family as long as the input and output lengths are determined. As an example, the CI hash function in [11] fits well into this format. However, there also exist hash functions outside the scope of Definition 2. For example, the CI hash from [40] does not fit into this definition[4] because its key length depends on some specific property of the interactive proofs we want to collapse, which cannot be fixed in advance given only the input and output lengths (more precisely, its key length depends on how fast we can compute the "bad function" of the protocol). Based on this definition, we give our first negative result:

▶ **Theorem 3** (Informally stated, see Theorem 5.2 of [13]). *Assuming the existence of a CRHF family that is computable in $\mathcal{D}$ and collision-resistant against $\mathcal{C}$ with any polynomial shrinkage, there does not exist a universal Fiat-Shamir hash from $\mathcal{C}$ to $\mathcal{D}$ with any polynomial input and output lengths.*

Here we further justify our assumption of a low-complexity CRHF. One can examine that the CRHF construction $f_A(x) = Ax \bmod q$ based on the hardness of short integer solution (SIS) [1] is actually computable in TC^0 (and has polynomial shrinkage). Also, the discrete-log (DLOG)-based CRHF construction $f_{g,h}(x_1, x_2) = g^{x_1} \cdot h^{x_2}$ can also be computed by a TC^0 circuit using some standard pre-processing techniques. Therefore, low-complexity CRHFs are known from these well-studied cryptographic assumptions, which leads to the following corollary:

▶ **Corollary 4.** *Assuming SIS/DLOG, there does not exist a universal Fiat-Shamir hash that can soundly transform any interactive argument into a non-interactive TC^0-argument.*

Actually, similar results hold also for circuit classes lower than TC^0. We present a candidate $\mathsf{AC}^0[2]$-computable poly-shrinkage CRHF based on a variant of SIS, indicating the impossibility of a universal Fiat-Shamir hash achieving even only $\mathsf{AC}^0[2]$-soundness. See Appendix B of the full version [13] for more discussions on the $\mathsf{AC}^0[2]$-computable CRHF.

Even more generally, we show that the result of Goldwasser and Kalai [26] – albeit looks complicated and requires super-constant depth at a first glance – can be generalized to the bounded-depth setting with adversaries from circuit classes as low as $\mathsf{AC}^0[2]$.

▶ **Theorem 5** (Informally stated, see Theorem 6.1 of [13]). *For any circuit class $\mathcal{C} \supseteq \mathsf{AC}^0[2]$, assuming the existence of a CRHF family with arbitrary polynomial shrinkage that is computable in $\mathcal{C}$ and collision-resistant against $\mathcal{C}$, there exists a 3-round public-coin $\mathcal{C}$-argument such that instantiating Fiat-Shamir using any hash function computable in $\mathcal{C}$ does not result in a sound non-interactive $\mathcal{C}$-argument.*

Compared to the previous result in Theorem 3 showing impossibility even when we require only some weaker soundness on the Fiat-Shamir-collapsed protocol, Theorem 5 shows only the impossibility of Fiat-Shamir hash functions preserving the same level of soundness

[4] Here we are only talking about the formats of these hash functions (e.g., how the key length is chosen) instead of their specific functionalities – of course, we do not have such a universal Fiat-Shamir hash with security against p.p.t. due to [26].

due to some technical issues. However, instead of saying every FS hash must fail on some tailored protocol, Theorem 5 is stronger in the sense that it shows the existence of some fixed protocol that makes every Fiat-Shamir hash fail. The Fiat-Shamir hash is allowed to depend arbitrarily on the fixed interactive protocols (e.g., the third message's length $|\gamma|$ or the prover/verifier running time, which are not allowed in Definition 2 and Theorem 3).

Also note that the CRHF in our assumption can be instantiated (at least for $\mathsf{AC}^0[2] \subseteq \mathcal{C} \subseteq \mathsf{NC}^1$) by the CRHF from [20] that is computable in $\mathsf{AC}^0[2]$ and secure against NC^1 based on the worst-case complexity assumption $\mathsf{NC}^1 \neq \oplus\mathsf{L}/\mathsf{poly}$.

1.4 Related works

Aside from the results like Goldwasser and Kalai [26] showing that there is an interactive argument such that no explicit hash function can collapse the protocol while preserving soundness, there have also been negative results for some restricted classes of protocols, in particular, for protocols related to SNARGs like Kilian's protocol [36]. Kilian's protocol is a 4-message succinct argument for any NP language. The idea of collapsing Kilian's protocol by Fiat-Shamir comes from Micali's CS proofs [39], which are now usually referred to as succinct non-interactive arguments (SNARGs). While Micali's construction applies Fiat-Shamir in the random oracle model, there have been negative results on Fiat-Shamir for Kilian's protocol in the plain model. Gentry and Wichs [25] showed a substantial barrier to constructing SNARGs in general (not limited to collapsing Kilian's protocol), and the problem of securely instantiating Fiat-Shamir on Kilian's protocol also suffers from this barrier. More recently, Bartusek et al. [5] showed that Fiat-Shamir with any explicit hash function can never be secure on some contrived instantiation of Kilian's protocol.

CI hash functions for constant degree polynomial relations against all ppt adversaries exist under various assumptions (for example, in [9, 7]), but none of them is computable in $\mathsf{AC}^0[2]$. After the initial publication of our work, Dao et al. [19] construct a CI hash function for degree-d polynomial relations against all ppt adversaries, assuming the hardness of solving random degree-d equations. When d is a constant, their hash function is also computable in $\mathsf{AC}^0[2]$. Let us remark that the assumptions used in their work and our work are technically incomparable, but we think our assumption is arguably weaker since it is a worst-case complexity assumption.

Organization. The rest of this paper is organized as follows. We first give an overview of the intuition and techniques underlying our results in Section 2. Then in Section 3, we present details of the main positive result, which is the construction of CI hash functions. Details of the negative results are deferred to the full version (see Sections 5 and 6 of [13]).

2 Technical Overview

2.1 Positive results

Our positive result is inspired by the CI hash function in [40], which crucially relies on the FHE scheme of Gentry, Sahai, Waters [24] (henceforth GSW-FHE). Our main observation is that the FHE scheme against low-depth adversaries constructed by Wang and Pan [42] (henceforth WP-FHE) bears a striking similarity with GSW-FHE, therefore it can be used as in [40] to obtain a CI hash function against low-depth adversaries.

Let us start by explaining the intuition of [40]. They consider sparse relations defined by efficient functions i.e. $\{(x, f(x))\}$ for some efficiently computable function f. Designing a hash function $h(k, x)$ that is correlation intractable for a single function-defined relation

is trivial: simply define $h(k, x) = f(x) + 1$. The construction [40] utilized the idea: for any f, a random hash key will look indistinguishable from a hash key that is specifically designed to be correlation intractable with respect to the relation defined by f, i.e. the construction achieves the notion of "somewhere correlation intractability". This idea can be implemented by using any fully homomorphic encryption scheme, i.e., the hash key is the homomorphic encryption of the circuit that computes a fixed function g_0 (hence for any f, it looks indistinguishable from the encryption of the circuit that computes f), and the hash function homomorphically computes $g_0(x) + 1$ using a universal circuit $\mathcal{U}(x, \cdot)$ s.t. $\mathcal{U}(x, f) = f(x)$. The result of homomorphic evaluation would be a ciphertext of $g_0(x) + 1$, so it remains as the last step to decrypt this ciphertext. One idea, as shown in [9], is to put the encryption of the FHE decryption function inside the public key.

$$h(k, x) = \mathsf{Eval}(\mathsf{pk}, \mathcal{U}(x, \cdot), \mathsf{ct}), \text{ where } k = (\mathsf{pk}, \mathsf{ct}), \mathsf{ct} = \mathsf{Enc}(\mathsf{pk}, g_0) \approx \mathsf{Enc}(\mathsf{pk}, \mathsf{Dec}(\mathsf{sk}, f(\cdot)) \oplus 1).$$

Then for any function f, if k is sampled according to $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{Dec}(\mathsf{sk}, f(\cdot)) \oplus 1))$, we have

$$f(x) = h(k, x) \implies f(x) = \mathsf{Enc}(\mathsf{pk}, \mathsf{Dec}(\mathsf{sk}, f(x)) \oplus 1) \implies \mathsf{Dec}(\mathsf{sk}, f(x)) = \mathsf{Dec}(\mathsf{sk}, f(x)) \oplus 1.$$

However, in this approach, in order to show the hash key k does not leak FHE secret key, we need to assume circular security. [40] bypasses this "decryption problem" based on an observation about the ciphertext of GSW-FHE. Namely, for any vector $(y_i)_{i \in [n]}$, if we have the ciphertext of each y_i, then we can compute $Ar + y$ where A is the FHE public key and r is derived from the randomness used to encrypt each y_i. Coarsely, if $y = f(x)$ as the result of homomorphic evaluation, then from $f(x) = Ar + f(x)$ we find a solution for $Ar = 0$. [40] manages to reduce breaking SIS assumption to breaking correlation intractability of their construction.

Our construction of CI against low-depth adversary can be explained in a single sentence: replace GSW-FHE in the construction [40] by WP-FHE. We claim that these two FHE schemes, though under different assumptions and are against different adversaries, are similar to each other in a precise sense: WP-FHE can be regarded as the mod-2, noise-free version of GSW-FHE. We provide the following table to sustain this intuition.

Table 1 Comparison between GSW-FHE and WP-FHE.

	GSW-FHE	WP-FHE
$(\mathsf{pk}, \mathsf{sk})$	$k^T A = e \approx 0$	$k^T A = 0$
Enc	$AR + mG$	$AR + mI$
Dec	$k^T \mathsf{ct} \cdot G^{-1}((0, \ldots, 0, q/2)^T)$	$k^T \mathsf{ct} \cdot (0, \ldots, 0, 1)^T$
$\mathsf{Eval}^{\mathrm{add}}$	$\mathsf{ct}_0 + \mathsf{ct}_1 - 2\mathsf{ct}_0 \cdot G^{-1}(\mathsf{ct}_1)$	$\mathsf{ct}_0 + \mathsf{ct}_1$
$\mathsf{Eval}^{\mathrm{mul}}$	$\mathsf{ct}_0 \cdot G^{-1}(\mathsf{ct}_1)$	$\mathsf{ct}_0 \cdot \mathsf{ct}_1$

The replacement allows the hash function to be computable in AC^0[2]. Also, we use a result in [22] to show that the security of our construction follows from the worst-case complexity assumption.

2.2 Negative results

Our main impossibility result comes from extending the impossibility result by Goldwasser and Kalai [26] to the bounded-depth settings. For readers familiar with [26], we first highlight the main technical challenges and techniques behind our extension.

- **Merkle tree hash**: The construction of [26] relies on a Merkle tree hash to succinctly commit to messages that may have an arbitrary polynomial size. In our bounded-depth setting, however, the depth of the Merkle tree is inherently limited to a constant. To handle limitation on depth, we base our construction on the stronger assumption of a low-depth CRHF with polynomial shrinkage – then a constant-depth Merkle tree is enough to handle any polynomial-size message.
- **PCP in $\mathsf{AC}^0[2]$**: [26] also uses probabilistic checkable proofs (PCPs) to guarantee the efficiency of the protocols. Since the prover in our setting is only allowed to do bounded-depth computations, we need a PCP satisfying all the required properties (including efficient oracle-construction, proof of knowledge, and reverse sampling as in [26, 4]) within a bounded-depth circuit class. To meet the efficiency requirements here, we construct such a PCP in $\mathsf{AC}^0[2]$, i.e., a PCP with $\mathsf{AC}^0[2]$-oracle-construction, $\mathsf{AC}^0[2]$-proof-of-knowledge, and $\mathsf{AC}^0[2]$-reverse-sampling (see Appendix C of the full version [13]).

Now we present the intuition of how to make Fiat-Shamir fail and our first impossibility result that directly follows the intuition (see Theorem 3). There is a simple intuition behind all existing contrived impossibility results: the prover in an interactive protocol is never able to predict the verifier's choice of the next message, but Fiat-Shamir makes the verifier's messages "predictable". Consider the standard 3-round public-coin setting where the protocol transcript is (α, β, γ). While sending the first message α, the prover can never predict the next message β in advance – no matter what prediction it makes, the verifier's random choice of β coincides with the prediction only with negligible probability. However, things become totally different when we collapse the protocol by Fiat-Shamir: with the help of a randomly chosen hash function $h \leftarrow \mathcal{H}_{FS}$, the prover now directly generates $(\alpha, \beta = h(x\|\alpha), \gamma)$ in one shot without the verifier's random choice. In other words, the next "verifier's message" $\beta = h(x\|\alpha)$ is just a fixed value after generating α, and of course, the malicious prover can predict β.

Based on the above observation, we can design an interactive protocol that simply asks the prover to predict the next message: in the first round, the prover tries to predict β by submitting the description of a function $\alpha = \hat{C}$ (we use C to denote the circuit of the function and $\hat{C}$ to denote the encoding of C), which means his prediction of β is $C(\alpha) = C(\hat{C})$. Then the verifier randomly chooses β and sends it to the prover, ignores the prover's third message, and finally accepts iff $\beta = C(\alpha)$ holds (which means the prediction is correct). This is an interactive proof for the empty language: the verifier always rejects except the randomly-chosen β happens to be the fixed value $C(\alpha)$, which holds with only negligible probability[5]. After applying Fiat-Shamir, unfortunately, the non-interactive protocol becomes totally broken. Given a Fiat-Shamir hash $h \leftarrow \mathcal{H}_{FS}$, the cheating prover can simply send the description of this hash function (i.e., send $\alpha = \widehat{h(x\|\cdot)}$ with the instance x hardcoded), and then the Fiat-Shamir transformation requires β to be $\beta = h(x\|\alpha)$, which makes the verifier accept.

This counter-example can work in the bounded-depth setting (since what the cheating prover does is simply copying the Fiat-Shamir hash function). However, there is an implicit limitation: when saying $\alpha = \widehat{h(x\|\cdot)}$, $\beta = h(x\|\alpha)$, we implicitly require the Fiat-Shamir

[5] Note that in an interactive protocol for the empty language, there is no *honest* prover since no instance can be proved (except with negligible probability). Or we can say the honest prover is just a machine that sends all-0 strings of some fixed lengths in each round. We clarify here that when describing the prover's behaviors, we actually mean what the prover is supposed to do (and the verifier will check it later). For example, by saying the prover sends $\alpha = \hat{C}$, we actually mean the cheating prover sends an arbitrarily α and the verifier will parse α into the description of a circuit C when deciding to accept/reject.

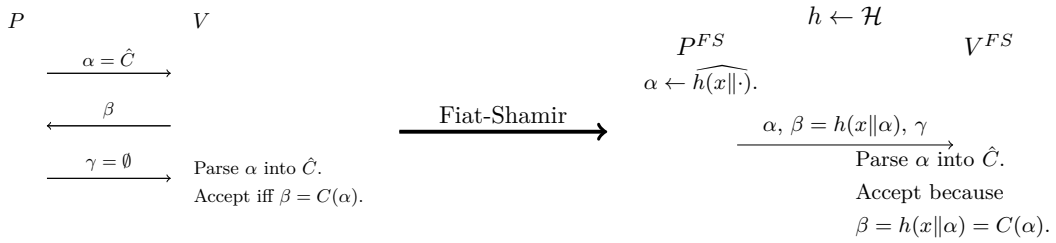

Figure 2 A protocol for the empty language that is not sound after Fiat-Shamir.

hash h to have a short enough description that can fit into its input, i.e. $|h| \leq |\alpha|$. As a result, this counter-example fails to capture a large class of Fiat-Shamir hash functions with the description length longer than the input length. This limitation somehow makes the counter-example worthless, since Fiat-Shamir hash functions usually have larger sizes: as shown in the work of Canetti et al. [12], a hash function cannot even be correlation intractable (i.e., preserving soundness for proofs) if its key length is shorter than the input length.

In order to bypass this limitation, we made some modifications based on the protocol above. Instead of sending the whole circuit description $\hat{C}$ in the first round, the prover only commits to such a description by sending a succinct commitment $\alpha = f(\hat{C})$ (f is the commitment function) – although the description may be long, its commitment can still be short enough to fit into the Fiat-Shamir hash's input. After that, the prover in the third round sends $\gamma = \hat{C}$ in clear to open the commitment, and the verifier decides to accept or reject by checking if $\alpha = f(\hat{C})$ (the opening is correct) and $\beta = C(\alpha)$ (the prediction is correct). See Figure 3 for an illustration of the modified protocol. In this case, a cheating prover after Fiat Shamir transformation can still set C to be the Fiat Shamir hash function $h(x||\cdot)$, and it is clear to see the verifier will accept the proof regardless the input x.

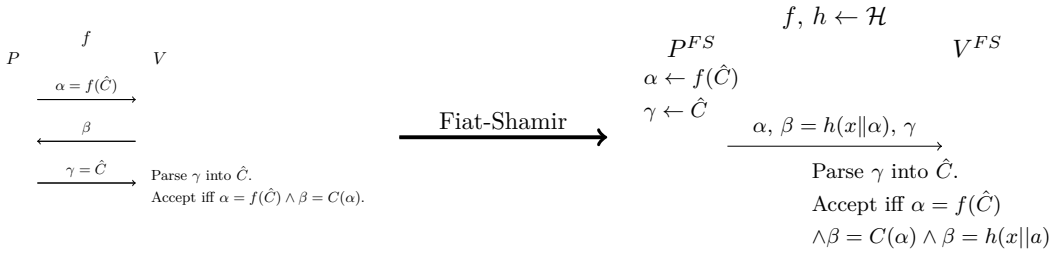

Figure 3 The modified protocol for the empty language.

In order to fit into the bounded-depth setting, we instantiate the succinct commitment f with a CRHF that can be computed in low depth and has enough shrinkage, which leads to our first negative result as shown in Theorem 3. Note that the exact shrinkage of CRHF matters here because we cannot boost the shrinkage arbitrarily in the bounded-depth setting: the circuit depth grows as we stack a CRHF several times to build a Merkle tree. The detailed security proof of this protocol is deferred to the full version (see Section 5 of [13]).

In the aforementioned counterexample, the hash function size remains bounded by a predetermined polynomial. This paper also explores adapting the findings of Goldwasser and Kalai [26] to achieve a more general negative result. Drawing inspiration from their construction, and by carefully addressing circuit depth and the challenges previously men-

tioned, we present three protocols. We demonstrate that at least one of these protocols must serve as a counterexample to the Fiat-Shamir transformation. Our strategy hinges on a win-win argument involving these three protocols: either an adversary can break the security of one protocol after the Fiat-Shamir transformation, thus invalidating it, or no adversary can compromise the security of another. In the latter case, we can leverage a protocol that is easily broken post-transformation but challenging to prove secure as our counterexample. The actual win-win argument is more intricate, involving all three protocols. For a detailed explanation, please refer to Section 6 of [13].

3 Correlation Intractability Against Bounded-Depth Adversaries

In this section we show positive results for Fiat-Shamir for proofs against bounded-depth adversaries. We first observe that the construction of correlation intractable (CI) hash functions of Peikert and Shiehian [40] is computable in TC^0, therefore it has already implied the existence of a CI hash function family computable in TC^0 against adversaries in circuit class $\mathcal{D} \supseteq \mathsf{TC}^0$, assuming SIS is hard against $\mathcal{D}$. We then show our main result: the construction of CI hash functions computable in $\mathsf{AC}^0[2]$ against adversaries in any circuit class $\mathcal{C} \supseteq \mathsf{AC}^0[2]$, assuming $\oplus\mathsf{L/poly} \not\subseteq \widetilde{\mathsf{Sum}}_{n^{-c}} \circ \mathcal{C}$ for some constant $c > 0$, which is a worst-case complexity assumption.

The rest of Section 3 is organized as follows. In Section 3.1 we provide the background needed for this section. In Section 3.2 we present the construction in [40]. The reasons we choose to present the construction in [40] are:

- We emphasize that the construction and hardness reduction can be computed in TC^0;
- The full construction makes the latter construction (which is our main result) more understandable.

Then in Section 3.3 we present our main construction of CI hash function computable in $\mathsf{AC}^0[2]$ against low-depth adversaries.

3.1 Selected Backgrounds

3.1.1 GSW Fully Homomorphic Encryption

▶ **Definition 6** (Lattice Gadgets over $\mathbb{Z}_q$). *For a positive integer modulus q, let $l = \lceil \lg q \rceil$. We define the "gadget" vector as*

$$g^T = (1, 2, \ldots, 2^{l-1}) \in \mathbb{Z}_q^l.$$

For every $u \in \mathbb{Z}_q$, there is an efficiently computable binary vector $g^{-1}(u) \in \{0,1\}^l$ such that $g \cdot g^{-1}(u) \equiv u \pmod q$, i.e., the binary representation of u. We treat $g^{-1} : \mathbb{Z}_q \to \{0,1\}^l$ as a function that computes binary decomposition.

The gadget matrix is defined as $G_n = I_n \otimes g^T$, and sometimes we drop the subscript n. Similarly, we define the function $G^{-1} = (I \otimes g^{-1}) : \mathbb{Z}_q^n \to \{0,1\}^{nl}$, which applies g^{-1} to each coordinate and concatenates the results. For any $u \in \mathbb{Z}_q^n$, we have

$$G \cdot G^{-1}(u) = u.$$

▶ **Construction 7** (GSW-FHE). *GSW-FHE is given by the function family* $\mathsf{GSW\text{-}FHE} = \{\mathsf{GSW\text{-}FHEGen}_n, \mathsf{GSW\text{-}Enc}_n, \mathsf{GSW\text{-}Dec}_n, \mathsf{GSW\text{-}Eval}_n\}_{n\in\mathbb{N}}$*, parameterized by the lattice dimension n, the error distribution χ, the modulus q, and the number of samples $m = n\lceil \log q \rceil$. We assume these parameters are chosen properly so that $LWE_{n-1,2m,q,\chi}$ is believed to be hard.*

- $\mathsf{GSW\text{-}FHEGen}_n$:
  1. *Take* $A' \leftarrow \mathbb{Z}_q^{(n-1)\times 2m}$, $s' \leftarrow \mathbb{Z}_q^{n-1}$, *and* $e \leftarrow \chi^{2m}$.
  2. *Compute* $b = A'^T s' + e \in \mathbb{Z}_q^{2m}$.
  3. *Return*

$$\mathsf{pk} = A = \begin{pmatrix} A' \\ b^T \end{pmatrix} \in \mathbb{Z}_q^{n\times 2m}, \mathsf{sk} = k = \begin{pmatrix} -s' \\ 1 \end{pmatrix} \in \mathbb{Z}_q^n.$$

  (Note that $k^T A = e^T \approx 0$*.)*
- $\mathsf{GSW\text{-}FHEGen}'_n$: *return* $\mathsf{pk} = A \leftarrow \mathbb{Z}_q^{n\times 2m}$ *sampled uniformly at random.*
- $\mathsf{GSW\text{-}Enc}_n(\mathsf{pk} = A, \mu \in \{0,1\})$:
  1. *Take* $R \leftarrow \{0,1\}^{2m\times m} \in \mathbb{Z}_q^{2m\times m}$.
  2. *Return* $\mathsf{ct} = AR + \mu G \in \mathbb{Z}_q^{n\times m}$.

  We allow $\mathsf{GSW\text{-}Enc}_n$ *to take a vector in* $\mathbb{Z}_2^m$ *as the input, instead of a single bit. In this case, the encryption of each dimension of that vector (using independent randomness) is concatenated together by each row. That is, for a length-*m *vector* x, $\mathsf{GSW\text{-}Enc}(\mathsf{pk} = A, x) = AR + x^T \otimes I_n \otimes g^T$, *where* $R \leftarrow \mathbb{Z}_2^{2m\times m^2}$.
- $\mathsf{GSW\text{-}Dec}_n(\mathsf{sk} = k, \mathsf{ct})$:
  1. *Take* $w = (0, 0, \ldots, 0, \lceil q/2 \rceil)^T \in \mathbb{Z}_q^n$.
  2. *Compute the value of* $V = k^T \mathsf{ct} \cdot G^{-1}(w)$, *output the decrypted message as* $\left\lceil \frac{V}{q/2} \right\rfloor$.
- $\mathsf{GSW\text{-}Eval}_n$. *We describe it by giving how to homomorphically compute addition and multiplication in* $\mathbb{Z}_q$.
  - $\mathsf{GSW\text{-}Eval}_n^{add}(\mathsf{pk} = A, (\mathsf{ct}_0, \mathsf{ct}_1))$: *return* $\mathsf{ct}_0 + \mathsf{ct}_1 - 2\mathsf{ct}_0 \cdot G^{-1}(\mathsf{ct}_1) \in \mathbb{Z}_q^{n\times m}$.
  - $\mathsf{GSW\text{-}Eval}_n^{mul}(\mathsf{pk} = A, (\mathsf{ct}_0, \mathsf{ct}_1))$: *return* $\mathsf{ct}_0 \cdot G^{-1}(\mathsf{ct}_1) \in \mathbb{Z}_q^{n\times m}$.

▶ Remark 8. For any constant-degree polynomial function family $\{f_n\}$, the corresponding circuit family computing $\mathsf{GSW\text{-}Eval}_n(\mathsf{pk}, f_n, (\mathsf{ct}_1, \ldots))$ is in TC^0. WP-FHE, which will be defined later, has a stronger property: the circuit family computing homomorphic evaluation of constant-degree polynomial function is in $\mathsf{AC}^0[2]$.

3.1.2 Inert Commitment

Let $A \in \mathbb{Z}_q^{n\times 2m}$ be the public key of GSW-FHE, and let $C_i = AR_i + x_i G$ be the homomorphic encryption of scalar $x_i \in \{0,1\}$, for $i \in [m]$. Then we concatenate all C_i's together, and view it as a commitment (relative to A) to $x^T = (x_1, \cdots, x_m)^T$ under randomness $R = \begin{pmatrix} R_1, & \cdots, & R_m \end{pmatrix}$, i.e.

$$C = \begin{pmatrix} C_1, & \cdots & , C_m \end{pmatrix} = AR + x^T \otimes G = AR + x^T \otimes I_n \otimes g^T = \mathsf{GSW\text{-}Enc}(A, x).$$

Since any matrix $M \in \mathbb{Z}_q^{n\times m}$ can be "vectorized" as an $v_M \in \mathbb{Z}_q^{nm}$, so that $(x^T \otimes I_n) \cdot v_M = Mx$, we have

$$c_M = C \cdot G^{-1}(v_M) = A \underbrace{(R \cdot G^{-1}(v_M))}_{r_M} + (x^T \otimes I_n \otimes g^T) \cdot (I_m \otimes I_n \otimes g^{-t}) v_M = Ar_M + Mx \in \mathbb{Z}_q^n.$$

We view c_M as an "inert commitment" to $Mx \in \mathbb{Z}_q^n$, and we write it as $\mathsf{InertEval}(M, C)$.

▶ **Definition 9** (Inert Commitment). *Given* $C \in \mathbb{Z}_q^{n\times m^2}$ *as the homomorphic encryption of some length-*m *vector, and* $M \in \mathbb{Z}_q^{n\times m}$ *as a matrix, we write*

$$\mathsf{InertEval}(M, C) := c_M = C \cdot G^{-1}(v_M),$$

where $v_M \in \mathbb{Z}_2^{nm}$ *is the vector satisfies* $(x^t \otimes I_n) \cdot v_M = Mx$ *for any length-*m *vector* x*, obtained by reordering the elements of* M*.*

3.1.3 Correlation intractability

▶ **Definition 10.** *We say a relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$ is searchable in a set of functions $\mathcal{F}$ if there exists a function $f : \mathcal{X} \to \mathcal{Y}$ in $\mathcal{F}$ such that if $(x, y) \in \mathcal{R}$ then $y = f(x)$.*

We say a relation class $\mathcal{S} = \{\mathcal{S}_n\}_{n\in\mathbb{N}}$ (i.e. a set of relations for each n) is searchable in a function family $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ if for each $n \in \mathbb{N}$, for all $\mathcal{R} \in \mathcal{S}_n$, $\mathcal{R}$ is searchable in $\mathcal{F}_n$.

▶ **Definition 11.** *Let $\mathcal{S} = \{\mathcal{S}_n\}$ be a relation class and $\mathcal{C}$ be a circuit class. A hash function family*

$$\mathcal{H} = \left\{ h_n : \{0,1\}^{l_k(n)} \times \{0,1\}^{l_{in}(n)} \to \{0,1\}^{l_{out}(n)} \right\}$$

with a key generation algorithm

$$\mathsf{Gen}_{\mathcal{H}} = \left\{ \mathsf{Gen}_{\mathcal{H},n} : \{0,1\}^* \to \{0,1\}^{l_k(n)} \right\}$$

is correlation intractable for $\mathcal{S}$ against $\mathcal{C}$ if for every circuit family $\mathcal{A} = \{\mathcal{A}_n\} \in \mathcal{C}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that for every $n \in \mathbb{N}$, for every $\mathcal{R} \in \mathcal{S}_n$,

$$\Pr\left[(x, h_n(k, x)) \in \mathcal{R} \;\middle|\; \begin{array}{l} k \leftarrow \mathsf{Gen}_{\mathcal{H},n} \\ x \leftarrow \mathcal{A}_n(k) \end{array} \right] < \mathsf{negl}(n).$$

We omit the polynomials $l_k(n), l_{in}(n), l_{out}(n)$ when they are clear from the explicit construction.

3.1.4 Cryptographic primitives against bounded-depth circuits

We consider the capability of adversaries as some circuit class that at least contains $\mathsf{AC}^0[2]$ (thus can compute matrix multiplication in $\mathbb{Z}_2$), with the assumption that the same circuit class augmented by an error-tolerant majority gate is not able to compute $\oplus\mathsf{L/poly}$.

▶ **Assumption 12.** *For a certain circuit class $\mathcal{C}$ s.t. $\mathsf{AC}^0[2] \subseteq \mathcal{C}$, there exists a constant $c > 0$ s.t. $\oplus\mathsf{L/poly} \not\subseteq \widetilde{\mathsf{Sum}}_{1/n^c} \circ \mathcal{C}$.*

For example, Assumption 12 is widely believed to be true for $\mathcal{C} \in \{\mathsf{AC}^0[2], \mathsf{TC}^0, \mathsf{NC}^1\}$.

▶ Remark 13. When $\mathcal{C} = \mathsf{NC}^1$, the assumption above is equivalent to $\oplus\mathsf{L/poly} \neq \mathsf{NC}^1$ as in [20].

In the following we introduce two distributions ZeroSamp and OneSamp. By using the result from randomized encoding [30, 2], any algorithm that distinguishes these two distributions can be used to compute functions in $\oplus\mathsf{L/poly}$. Therefore, from the hardness of $\oplus\mathsf{L/poly}$ (as in Assumption 12), we can show that it is hard to distinguish ZeroSamp from OneSamp. Please see [20] for more detailed discussion.

▶ **Definition 14** (ZeroSamp, OneSamp)**.** *Let n be the security parameter. For $r^{(1)} \in \mathbb{Z}_2^{n(n-1)/2}$, and $r^{(2)} \in \mathbb{Z}_2^{n-1}$, denote*

$$L(r^{(1)}) = \begin{pmatrix} 1 & r_1^{(1)} & r_2^{(1)} & \cdots & r_{n-1}^{(1)} \\ & 1 & r_n^{(1)} & \cdots & r_{2n-3}^{(1)} \\ & & \ddots & \ddots & \vdots \\ & & & 1 & r_{n(n-1)/2}^{(1)} \\ & & & & 1 \end{pmatrix}, R(r^{(2)}) = \begin{pmatrix} 1 & 0 & 0 & 0 & r_1^{(2)} \\ & 1 & 0 & 0 & r_2^{(2)} \\ & & \ddots & \vdots & \vdots \\ & & & 1 & r_{n-1}^{(2)} \\ & & & & 1 \end{pmatrix}.$$

Also, let M_0^n, M_1^n be the following $n \times n$ matrices:

$$M_0^n = \begin{pmatrix} \boldsymbol{0}^T & 0 \\ I_{n-1} & \boldsymbol{0} \end{pmatrix}, M_1^n = \begin{pmatrix} \boldsymbol{0}^T & 1 \\ I_{n-1} & \boldsymbol{0} \end{pmatrix} \quad (I_{n-1} \text{ is the identity matrix of rank } n-1).$$

Then we define two sampling procedures:

- $\mathsf{ZeroSamp}(n)$*: take* $r^{(1)} \leftarrow \{0,1\}^*, r^{(2)} \leftarrow \{0,1\}^*$ *and output* $L(r^{(1)})M_0^n R(r^{(2)})$.
- $\mathsf{OneSamp}(n)$*: take* $r^{(1)} \leftarrow \{0,1\}^*, r^{(2)} \leftarrow \{0,1\}^*$ *and output* $L(r^{(1)})M_1^n R(r^{(2)})$.

Note that the output of $\mathsf{ZeroSamp}(n)$ *always has rank* $n-1$*, and the output of* $\mathsf{OneSamp}(n)$ *always has rank* n*. What's more, the random bits used by* $A \leftarrow \mathsf{ZeroSamp}(n)$ *also gives the kernel of* A*, and therefore we sometimes use* $A, k \leftarrow \mathsf{ZeroSamp}(n)$ *where* k *s.t.* $Ak = 0$ *is additionally given by* $\mathsf{ZeroSamp}(n)$.

If the security parameter n *is clear from context, we may drop it and use the notation* ZeroSamp, OneSamp.

Previous literature discovered that NC^1 circuits cannot distinguish ZeroSamp and OneSamp if $\oplus\mathsf{L/poly} \neq \mathsf{NC}^1$. We extend this idea from NC^1 to other circuit classes that contain $\mathsf{AC}^0[2]$, as stated in the following lemma.

▶ **Lemma 15** (See Lemma 4.3 of [20] for $\mathcal{C} = \mathsf{NC}^1$)**.** *For any circuit class* $\mathcal{C} \supseteq \mathsf{AC}^0[2]$*, if Assumption 12 holds for* $\mathcal{C}$*, there is a negligible function* $\mathsf{negl}(\cdot)$ *such that for any family* $\mathcal{F} = \{f_n\}$ *in* $\mathcal{C}$*, for an infinite number of values of* n*,*

$$|\Pr[f_n(M) = 1 \mid M \leftarrow \mathsf{ZeroSamp}(n)] - \Pr[f_n(M) = 1 \mid M \leftarrow \mathsf{OneSamp}(n)]| < \mathsf{negl}(n).$$

▶ Remark 16. As Remark 3.1 of [20] points out, from worst-case complexity assumptions, we can only achieve infinitely-often primitives, as if we assume $\mathcal{D} \nsubseteq \mathcal{C}$ where $\mathcal{C}, \mathcal{D}$ are circuit classes, we only mean that there exists a function family $\{f_n\} \in \mathcal{D}$, such that for any function family $\{g_n\} \in \mathcal{C}$, $f_n \neq g_n$ for infinitely many n.

However, as long as we strengthen the assumption from the previous one to that there exists $\{f_n\} \in \mathcal{D}$, for all $\{g_n\} \in \mathcal{C}$, $\exists n_0$ such that $f_n \neq g_n$ for all $n > n_0$, we can achieve the conventional almost-everywhere security. We state this observation below. All our constructions assuming 12 automatically achieve almost-everywhere security using the strengthened assumption, and we only state theorems in the infinitely-often secure version for brevity.

▶ **Assumption 17** (Strengthened assumption)**.** *For a certain circuit class* $\mathcal{C}$ *s.t.* $\widetilde{\mathsf{AC}^0}[2] \subseteq \mathcal{C}$*, there exists* $c > 0$ *s.t.* $\exists \{f_n\} \in \oplus\mathsf{L/poly}$ *such that for any function family* $\{g_n\} \in \widetilde{\mathsf{Sum}}_{1/n^c} \circ \mathcal{C}$*,* $\exists n_0$ *such that* $f_n \neq g_n$ *for all* $n > n_0$*.*

▶ **Lemma 18.** *For any circuit class* $\mathcal{C} \supseteq \mathsf{AC}^0[2]$*, if Assumption 17 holds for* $\mathcal{C}$*, there is a negligible function* $\mathsf{negl}(\cdot)$ *such that for any family* $\mathcal{F} = \{f_n\}$ *in* $\mathcal{C}$*, for all but infinitely many* n*,*

$$|\Pr[f_n(M) = 1 \mid M \leftarrow \mathsf{ZeroSamp}(n)] - \Pr[f_n(M) = 1 \mid M \leftarrow \mathsf{OneSamp}(n)]| < \mathsf{negl}(n).$$

Since we extend previous work from NC^1 to general circuit classes $\mathcal{C}$, and our previous lemmas 15, 18 are slightly different from those in literature (see lemma 4.3 of [20], where $n = n(\lambda)$ is polynomial of security parameter), we include proofs of those lemmas in Appendix A of [13].

▶ **Construction 19** (WP-FHE [42]). *WP-FHE is given by the function family* $\mathsf{WP\text{-}FHE} = \{\mathsf{WP\text{-}FHEGen}_n, \mathsf{WP\text{-}Enc}_n, \mathsf{WP\text{-}Dec}_n, \mathsf{WP\text{-}Eval}_n\}_{n\in\mathbb{N}}$.

- $\mathsf{WP\text{-}FHEGen}_n$: *Take* $A^T, k \leftarrow \mathsf{ZeroSamp}(n)$, *and set* $\mathsf{pk} = A, \mathsf{sk} = k$.
- $\mathsf{WP\text{-}FHEGen}'_n$: *Take* $A^T \leftarrow \mathsf{OneSamp}(n)$, *and then output* $\mathsf{pk} = A$.
- $\mathsf{WP\text{-}Enc}_n(\mathsf{pk} = A, m \in \{0,1\})$: *Take* $R \leftarrow \mathbb{Z}_2^{n\times n}$, *then return* $\mathsf{ct} = AR + mI_n \in \mathbb{Z}_2^{n\times n}$. *We allow* $\mathsf{WP\text{-}Enc}_n$ *to take a vector in* $\mathbb{Z}_2^n$ *as input, instead of a single bit. In this case, the encryption of each dimension of that vector (using independent randomness) is concatenated together by each row. That is, for a length-*n *vector* x, $\mathsf{WP\text{-}Enc}(\mathsf{pk} = A, x) = AR + x^T \otimes I_n$, *where* $R \leftarrow \mathbb{Z}_2^{n\times n^2}$.
- $\mathsf{WP\text{-}Dec}_n(\mathsf{sk} = k, \mathsf{ct})$: *Let* c *be the* n*-th column vector of* ct, *and then return* $k \cdot c$.
- $\mathsf{WP\text{-}Eval}_n$. *We describe it by giving how to homomorphically compute addition and multiplication in* $\mathbb{Z}_2$.
  - $\mathsf{WP\text{-}Eval}_n^{add}(\mathsf{pk} = A, (\mathsf{ct}_0, \mathsf{ct}_1))$: *return* $\mathsf{ct}_0 + \mathsf{ct}_1 \in \mathbb{Z}_2^{n\times n}$.
  - $\mathsf{WP\text{-}Eval}_n^{mul}(\mathsf{pk} = A, (\mathsf{ct}_0, \mathsf{ct}_1))$: *return* $\mathsf{ct}_0\mathsf{ct}_1 \in \mathbb{Z}_2^{n\times n}$.

▶ Remark 20. We emphasize the striking similarity between GSW-FHE and WP-FHE. The (pk, sk) pairs of both schemes are (pk= A, sk= k) where A is a matrix, k is a vector with $k^TA \approx 0$ in GSW-FHE, and $k^TA = 0$ in WP-FHE. What's more, if we treat the identity matrix I as G's analog in $\mathbb{Z}_2$, then the encryption of both schemes is $\mathsf{Enc}(\mathsf{pk} = A, m) = AR + mG$. Similarly, homomorphic addition of both schemes is adding two ciphertexts together, and $\mathsf{Eval}^{\mathrm{mul}}(\mathsf{ct}_0, \mathsf{ct}_1) = \mathsf{ct}_0G^{-1}(\mathsf{ct}_1)$ for both schemes. Recall Table 1 as a summary of the above observations.

Our proof uses the following lemma:

▶ **Lemma 21** ([22]). *For any circuit class* $\mathcal{C} \supseteq \mathsf{AC}^0[2]$, *if Assumption 12 holds for* $\mathcal{C}$, *then* $\{p_A(x) = Ax : A \leftarrow \mathit{OneSamp}\}$ *is a collection of one-way permutations against* $\mathcal{C}$, *that is, for any adversary family* $\{\mathcal{A}_n\}_{n\in\mathbb{N}} \in \mathcal{C}$, *there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that for infinitely many* n,

$$\Pr\left[Ay = Ax \;\middle|\; \begin{array}{l} A \leftarrow \mathsf{OneSamp}(n) \\ x \leftarrow \mathbb{Z}_2^n \\ y \leftarrow \mathcal{A}_n(A, Ax) \end{array}\right] < \mathsf{negl}(n).$$

[22] only proves the lemma for $\mathcal{C} = \mathsf{NC}^1$, but actually their proof can be applied to any circuit class $\mathcal{C}$ that satisfies requirement 12. We give a sketch of their proof.

Proof sketch. The idea is to use the adversary $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ that breaks one-wayness with probability $1/p(n)$ to distinguish ZeroSamp and OneSamp. First, checking $Ay = Ax$ involves only matrix multiplication, so we can assume $\mathcal{A}_n$ returns y with $Ay = Ax$ or $\perp$. Since $\{A_n\}_{n\in\mathbb{N}} \in \mathcal{C}$ cannot distinguish ZeroSamp and OneSamp, we have

$$\Pr\left[Ay = Ax \;\middle|\; \begin{array}{l} A \leftarrow \mathsf{ZeroSamp}(n) \\ x \leftarrow \mathbb{Z}_2^n \\ y \leftarrow \mathcal{A}_n(A, Ax) \end{array}\right] > \frac{1}{p(n)} - \mathsf{negl}(n) > \frac{1}{p'(n)}.$$

For A sampled from ZeroSamp, whenever $\mathcal{A}_n$ finds y s.t. $Ay = Ax$, we have probability $\frac{1}{2}$ that $x \neq y$ due to the randomness of x (recall A is of rank $n-1$, so y has two solutions), in which case we have $x - y$ as a non-trivial kernel of A. Therefore by invoking $\mathcal{A}_n$ we have non-negligible probability to obtain the kernel of A, which can be used to distinguish ZeroSamp and OneSamp (recall that the output of OneSamp is always full-rank).

We emphasize that only matrix multiplication in $\mathbb{Z}_2$ is used in this reduction, and thus the reduction is computable in $\mathcal{C} \supseteq \mathsf{AC}^0[2]$. ◀

3.2 CI from SIS against bounded-depth adversaries

In this subsection, we follow the construction of [40] to show that it can be computed in TC^0, yielding a CI hash against bounded-depth adversary under the super-weak assumption of SIS against bounded-depth adversaries.

▶ **Theorem 22.** *For any circuit class $\mathcal{C}$ that contains TC^0, for any polynomial $S(n)$ and constant d, assuming the hardness of $\mathsf{SIS}_{n,m+1,q,\beta}$ for $q = \mathsf{poly}(n), l = \lceil \log q \rceil, m = nl$ and sufficiently large $\beta = m^{O(d)}$ against $\mathcal{C}$, there exists a hash family computable in TC^0 that is CI for any relation class searchable by size-$S(n)$[6], degree-d polynomials $\{f_n : \{0,1\}^n \to \{0,1\}^m\}_{n\in\mathbb{N}}$, against adversaries in $\mathcal{C}$.*

This theorem is proven by the construction exhibited in this subsection (see Construction 24, Lemma 26). By lemma B.3 of [13], when n is large enough, by setting $q \geq n^{\Omega(d)}$ s.t. $q \geq \beta \cdot n^{\Omega(1)}$, there is an polynomial time reduction from $n^{\Omega(d)}$-approximate SIVP to $\mathsf{SIS}_{n,m+1,q,\beta}$.

▶ Remark 23. The hash function is only CI for constant-degree polynomials because TC^0 circuits can only perform a constant number of homomorphic multiplications (which involve matrix multiplications). Therefore, achieving constant-degree polynomials is the best we can do with FHE-based constructions.

▶ **Construction 24.** *The hash family $\mathcal{H} = \{h_n\}_{n\in\mathbb{N}}$ with key generation algorithm $\mathsf{Gen}_{\mathcal{H}} = \{\mathsf{Gen}_{\mathcal{H},n}\}_{n\in\mathbb{N}}$ is parameterized by an arbitrary circuit size $S(n) = \mathsf{poly}(n)$ and depth d. Let $\mathcal{U}(C, x) = C(x)$ denote the universal circuit for size-S, degree-d polynomials (hence, it is also a constant degree polynomial itself). We write $q = q(n), l = \lceil \log q \rceil, m = nl$.*

- $\mathsf{Gen}_{\mathcal{H},n}$*: generate $A \leftarrow \mathbb{Z}_q^{n\times 2m}$ and $C \leftarrow \mathsf{GSW\text{-}Enc}(A, 0^{S(n)}) = AR + 0^{S(n)} \otimes G$, where $R \in \mathbb{Z}^{2m\times m}$. Choose a uniform random $a \leftarrow \mathbb{Z}_q^n$, output the hash key $k = (a, C)$.*
- $h_n(k = (a, C), x \in \{0,1\}^n)$*: let circuit $U_x(\cdot) = \mathcal{U}(x, \cdot)$, and output*

$$G_n^{-1}\left[a + \mathsf{InertEval}(G_n, \mathsf{GSW\text{-}Eval}(U_x, C))\right] \in \{0,1\}^m.$$

We first show this hash function family can be computed in TC^0, then show its correlation intractability. The latter property directly follows the proof of Theorem 3.4 in [40].

▶ **Lemma 25.** *Construction 24 can be computed by TC^0 circuits.*

Proof. Computing $\mathsf{Gen}_{\mathcal{H},n}$ and h_n only requires:

- Computing iterative sum mod $q = \mathsf{poly}(n)$; (In this construction, we only need to compute the product of a normal matrix (with each element in $\mathbb{Z}_q$) and a binary matrix (with each element in $\{0,1\}$), so it suffices to compute iterative sum in $\mathbb{Z}_q$.)
- Generating universal circuit U_x from x; (we only need to replace some input gates with binary values.)
- Computing G_n^{-1}, i.e. bit decomposition, which is trivial since we store a number in $\mathbb{Z}_q$ as a binary number.

All these tasks can be computed in TC^0. ◀

▶ **Lemma 26.** *Assuming the hardness of $\mathsf{SIS}_{n,m,q,\beta}$ for a sufficiently large $\beta = m^{O(d)}$ against $\mathcal{C}$ adversary, Construction 24 is correlation intractable (cf. Theorem 22).*

[6] Here "size" means the encoding length of the constant degree polynomial.

Proof. Let $\mathcal{A} = \{\mathcal{A}_n\}_{n\in\mathbb{N}} \in \mathcal{C}$ be any adversary that breaks correlation intractability against $\{f_n\}$, i.e.

$$\Pr\left[h_n(k=(a,C),x) = f_n(x) \;\middle|\; \begin{array}{l} A \leftarrow \mathbb{Z}_q^{n\times 2m} \\ C \leftarrow \mathsf{GSW\text{-}Enc}(A, 0^{S(n)}) \\ a \leftarrow \mathbb{Z}_q^n \\ x \leftarrow \mathcal{A}_n(k) \end{array}\right]$$

is non-negligible.

First, we change $C \leftarrow \mathsf{GSW\text{-}Enc}(A, 0^{S(n)})$ to $C \leftarrow \mathsf{GSW\text{-}Enc}(A, f)$ for $f = f_n$ by hybrid argument. It can be done because the distribution of C is statistically indistinguishable by the following lemma.

▶ **Lemma 27** (Item 1 of Proposition 2.10 in [40]). *For any $x \in \mathbb{Z}_q^n$, the statistical distance between the distribution of $(A \leftarrow \mathbb{Z}_q^{n\times 2m}, C = \mathsf{GSW\text{-}Enc}(A, x))$ and uniformly random is in $\mathsf{negl}(m)$ by the leftover hash lemma.*

Therefore,

$$\Pr\left[h_n(k=(a,C),x) = f_n(x) \;\middle|\; \begin{array}{l} A \leftarrow \mathbb{Z}_q^{n\times 2m} \\ C \leftarrow \mathsf{GSW\text{-}Enc}(A, f_n) \\ a \leftarrow \mathbb{Z}_q^n \\ x \leftarrow \mathcal{A}_n(k) \end{array}\right]$$

is also non-negligible. Then we can use $\mathcal{A}$ to break SIS. Since $h_n(k,x) = f_n(x)$ implies that

$$\begin{aligned} G_n \cdot f_n(x) =& G_n \cdot h_n(k,x) \\ =& a + \mathsf{InertEval}(G_n, \mathsf{GSW\text{-}Eval}(U_x, C)) \\ =& a + (Ar + G_n \cdot f_n(x)) \\ =& A'z + G_n \cdot f_n(x), \end{aligned}$$

where

$$A' = \begin{bmatrix} a & A \end{bmatrix} \in \mathbb{Z}_q^{n\times(m+1)}, z = (1,r) \in \mathbb{Z}^{m+1}.$$

Note that $||z|| = m^{O(d)}$ as U_x is done by universally computing depth-d circuits.

So our SIS attacker works as follows: given an SIS instance A', the attacker takes A as the last m columns of A' and generates $C \leftarrow \mathsf{GSW\text{-}Enc}(A, f)$ with randomness R retained. After taking $x \leftarrow \mathcal{A}_n(k)$, it simulates the computation of $h_n(k,x)$ to get r, then outputs $z = (1,r)$. The attacker is in $\mathcal{C}$ since the reduction is in TC^0. ◀

3.3 CI from worst-case complexity assumptions

In this subsection, we bypass the reliance on the lattice assumptions, which therefore also allows the hash function to be computable in $\mathsf{AC}^0[2]$, by applying techniques from [40] to WP-FHE (see Construction 19) instead of GSW-FHE (see Construction 7). This approach provides the construction of a hash function family that is computable in $\mathsf{AC}^0[2]$ and correlation-intractable for any relation searchable in constant-degree polynomials (with bounded circuit size) against any circuit class $\mathcal{C} \supseteq \mathsf{AC}^0[2]$, assuming $\oplus\mathsf{L/poly} \not\subseteq \widetilde{\mathsf{Sum}}_{n^{-c}} \circ \mathcal{C}$ for some $c > 0$. The result can be stated as follows:

▶ **Theorem 28.** *For any circuit class $\mathcal{C} \supseteq \mathsf{AC}^0[2]$, if Assumption 12 holds for $\mathcal{C}$, for any polynomial $S(n)$ and constant d, there exists a hash family computable in $\mathsf{AC}^0[2]$ that is CI for any relation class searchable by degree-d polynomials $\{f_n : \{0,1\}^n \to \{0,1\}^n\}_{n\in\mathbb{N}}$ computable by circuits of size $S(n)$, against adversaries in $\mathcal{C}$ (following the "infinitely many" version of security).*

3.3.1 Adaptation of inert commitment

In our construction, all computation are performed in $\mathbb{Z}_2$, hence we restate the inert commitment in $\mathbb{Z}_2$, i.e, set $q = 2$. Although it is just the special case for $q = 2$, we emphasize that in this case, G is simplified to the identify matrix (as what we observed in WP-FHE), hence the inert commitment is greatly simplified.

▶ **Definition 29** (Inert Commitment in $\mathbb{Z}_2$)**.** *Given $C = \mathsf{WP\text{-}Enc}(A, x) \in \mathbb{Z}_2^{n\times n^2}$ as the homomorphic encryption of some length-n vector, and $M \in \mathbb{Z}_2^{n\times n}$ as a matrix, we write*

$$\mathsf{InertEval}(M, C) := c_M = C \cdot v_M,$$

where $v_M \in \mathbb{Z}_2^{n^2}$ is the vector satisfies $(x^T \otimes I_n) \cdot v_M = Mx$ for any length-n vector x, obtained by reordering the elements of M.

3.3.2 Construction

Here we present the construction of the CI hash family to prove Theorem 28.

▶ **Construction 30.** *The hash family $\mathcal{H} = \{h_n\}_{n\in\mathbb{N}}$ with the key generation algorithm $\mathsf{Gen}_{\mathcal{H}} = \{\mathsf{Gen}_{\mathcal{H},n}\}_{n\in\mathbb{N}}$ is parameterized by an arbitrary circuit size $S(n) = \mathsf{poly}(n)$ and depth d. Let $U(C, x) = C(x)$ denote a universal circuit for size-S, degree-d polynomials.*

- *$\mathsf{Gen}_{\mathcal{H},n}$: generate $A \leftarrow \mathsf{OneSamp}$ and $C \leftarrow \mathsf{WP\text{-}Enc}(A, 0^{S(n)})$, choose a uniformly random $a \leftarrow \mathbb{Z}_2^n$ and output the hash key $k = (a, C)$.*
- *$h_n(k = (a, C), x \in \{0,1\}^n)$: define the circuit $U_x(\cdot)$ as $U_x(\cdot) = U(x, \cdot)$ and output*

$$a + \mathsf{InertEval}(I_n, \mathsf{WP\text{-}Eval}(U_x, C)) \in \{0,1\}^n.$$

Proof of Theorem 28. We prove that for any circuit class $\mathcal{C} \supseteq \mathsf{AC}^0[2]$, Construction 30 is a CI hash family against $\mathcal{C}$ if Assumption 12 holds for $\mathcal{C}$. Let $\mathcal{A} = \{\mathcal{A}_n\} \in \mathcal{C}$ be any adversary family that breaks CI for some sequence of polynomials $\{f_n\}$. Since given x, we can check whether $h_n((a, C), x) = f_n(x)$ in $\mathcal{C}$ (which only involves matrix multiplication and computing $f_n(x)$), w.l.o.g., we assume that $\mathcal{A}_n$ always outputs some $x \in \{0,1\}^n$ s.t. $h_n((a, C), x) = f_n(x)$ with non-negligible probability or outputs $\perp$ after taking $k = (a, C)$ sampled according to $\mathsf{Gen}_{\mathcal{H},n}$ as input.

We abuse the notation to let $f_n \in \{0,1\}^{S(n)}$ also denote the encoding of the circuit that computes this function. The idea is to apply the hybrid argument to change the input distribution of $\mathcal{A}_n$. We observe that the following two distribution are equivalent, since A (sampled from $\mathsf{OneSamp}$) is always full-rank, which makes AR below being uniformly random. Hence the two distributions are indistinguishable for $\mathcal{C}$ circuits.

- $\mathcal{D}_n^0 = \{(a, \mathsf{WP\text{-}Enc}(A, 0^{S(n)}) = AR + 0^{S(n)} \otimes I_n) : a \leftarrow \mathbb{Z}_2^n, A \leftarrow \mathsf{OneSamp}, R \leftarrow \mathbb{Z}_2^{n\times nS(n)}\}$.
- $\mathcal{D}_n^1 = \{(a, \mathsf{WP\text{-}Enc}(A, f_n) = AR + f_n \otimes I_n) : a \leftarrow \mathbb{Z}_2^n, A \leftarrow \mathsf{OneSamp}, R \leftarrow \mathbb{Z}_2^{n\times nS(n)}\}$.

▶ **Lemma 31.** *For any circuit class $\mathcal{C} \supseteq \mathsf{AC}^0[2]$, if Assumption 12 holds for $\mathcal{C}$, any circuit family $\{\mathcal{A}_n\} \in \mathcal{C}$ cannot distinguish $\mathcal{D}_n^0$ and $\mathcal{D}_n^1$, i.e,*

$$\left| \Pr\left[\mathcal{A}_n(k) = 1 \;\middle|\; k \leftarrow \mathcal{D}_n^0\right] - \Pr\left[\mathcal{A}_n(k) = 1 \;\middle|\; k \leftarrow \mathcal{D}_n^1\right] \right| < \mathsf{negl}(n).$$

So after switching the input to $\mathcal{A}_n$ from $k \leftarrow \mathcal{D}_n^0$ to $k' \leftarrow \mathcal{D}_n^1$, $\mathcal{A}_n$ should also output $x \in \{0,1\}^n$ s.t. $h_n(k' = (a, \mathsf{WP\text{-}Enc}(A, f_n)), x) = f_n(x)$ with non-negligible probability. Since

$$\begin{aligned} f_n(x) =& h_n(k', x) \\ =& a + \mathsf{InertEval}(I_n, \mathsf{WP\text{-}Eval}(U_x, \mathsf{WP\text{-}Enc}(A, f_n))) \\ =& a + \mathsf{InertEval}(I_n, AR + f_n(x)^t \otimes I_n) \\ =& a + Ar_{I_n} + f_n(x), \end{aligned}$$

we have $Ar_{I_n} = a$ (in $\mathbb{Z}_2$). So if we record the randomness used to compute $\mathsf{WP\text{-}Enc}(A, f_n)$, by calculating $h_n(k', x)$, we are able to compute $r_{I_n} = A^{-1}a$. However, this allows us to invert $p_A(x) = Ax$, which is impossible by Lemma 21. ◀

References

1. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996. doi:10.1145/237814.237838.
2. Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in nc^0. *SIAM J. Comput.*, 36(4):845–888, 2006. doi:10.1137/S0097539705446950.
3. Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS*, pages 106–115. IEEE Computer Society, 2001. doi:10.1109/SFCS.2001.959885.
4. Boaz Barak and Oded Goldreich. Universal arguments and their applications. *SIAM J. Comput.*, 38(5):1661–1694, 2008. doi:10.1137/070709244.
5. James Bartusek, Liron Bronfman, Justin Holmgren, Fermi Ma, and Ron D. Rothblum. On the (in)security of kilian-based snargs. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 522–551. Springer, 2019. doi:10.1007/978-3-030-36033-7_20.
6. Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: a paradigm for keyless hash functions. In *STOC*, pages 671–684. ACM, 2018. doi:10.1145/3188745.3188870.
7. Zvika Brakerski, Venkata Koppula, and Tamer Mour. NIZK from LPN and trapdoor hash via correlation intractability for approximable relations. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 738–767. Springer, 2020. doi:10.1007/978-3-030-56877-1_26.
8. Matteo Campanelli and Rosario Gennaro. Fine-grained secure computation. In Amos Beimel and Stefan Dziembowski, editors, *Theory of Cryptography - 16th International Conference, TCC 2018, Panaji, India, November 11-14, 2018, Proceedings, Part II*, volume 11240 of *Lecture Notes in Computer Science*, pages 66–97. Springer, 2018. doi:10.1007/978-3-030-03810-6_3.
9. Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-shamir: from practice to theory. In *STOC*, pages 1082–1090. ACM, 2019. doi:10.1145/3313276.3316380.
10. Ran Canetti, Yilei Chen, and Leonid Reyzin. On the correlation intractability of obfuscated pseudorandom functions. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 389–415. Springer, 2016. doi:10.1007/978-3-662-49096-9_17.

11 Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-shamir and correlation intractability from strong kdm-secure encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *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*, volume 10820 of *Lecture Notes in Computer Science*, pages 91–122. Springer, 2018. doi:10.1007/978-3-319-78381-9_4.

12 Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004. doi:10.1145/1008731.1008734.

13 Liyan Chen, Yilei Chen, Zikuan Huang, Nuozhou Sun, Tianqi Yang, and Yiding Zhang. Fiat-shamir for bounded-depth adversaries. Cryptology ePrint Archive, Paper 2024/256, 2024. URL: https://eprint.iacr.org/2024/256.

14 Yilei Chen, Alex Lombardi, Fermi Ma, and Willy Quach. Does fiat-shamir require a cryptographic hash function? In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part IV*, volume 12828 of *Lecture Notes in Computer Science*, pages 334–363. Springer, 2021. doi:10.1007/978-3-030-84259-8_12.

15 Arka Rai Choudhuri, Sanjam Garg, Abhishek Jain, Zhengzhong Jin, and Jiaheng Zhang. Correlation intractability and snargs from sub-exponential DDH. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part IV*, volume 14084 of *Lecture Notes in Computer Science*, pages 635–668. Springer, 2023. doi:10.1007/978-3-031-38551-3_20.

16 Arka Rai Choudhuri, Pavel Hubácek, Chethan Kamath, Krzysztof Pietrzak, Alon Rosen, and Guy N. Rothblum. Finding a nash equilibrium is no easier than breaking fiat-shamir. In *STOC*, pages 1103–1114. ACM, 2019. doi:10.1145/3313276.3316400.

17 Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. Non-interactive batch arguments for NP from standard assumptions. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part IV*, volume 12828 of *Lecture Notes in Computer Science*, pages 394–423. Springer, 2021. doi:10.1007/978-3-030-84259-8_14.

18 Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. Snargs for P from LWE. In *FOCS*, pages 68–79. IEEE, 2021.

19 Quang Dao, Aayush Jain, and Zhengzhong Jin. Non-interactive zero-knowledge from LPN and MQ. In *CRYPTO (9)*, volume 14928 of *Lecture Notes in Computer Science*, pages 321–360. Springer, 2024. doi:10.1007/978-3-031-68400-5_10.

20 Akshay Degwekar, Vinod Vaikuntanathan, and Prashant Nalini Vasudevan. Fine-grained cryptography. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part III*, volume 9816 of *Lecture Notes in Computer Science*, pages 533–562. Springer, 2016. doi:10.1007/978-3-662-53015-3_19.

21 Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *FOCS*, pages 523–534. IEEE Computer Society, 1999. doi:10.1109/SFFCS.1999.814626.

22 Shohei Egashira, Yuyu Wang, and Keisuke Tanaka. Fine-grained cryptography revisited. *J. Cryptol.*, 34(3):23, 2021. doi:10.1007/s00145-021-09390-3.

23 Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986. doi:10.1007/3-540-47721-7_12.

24 Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology*

Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, 2013. doi:10.1007/978-3-642-40041-4_5.

25 Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *STOC*, pages 99–108. ACM, 2011. doi:10.1145/1993636.1993651.

26 Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the fiat-shamir paradigm. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 102–113. IEEE Computer Society, 2003. doi:10.1109/SFCS.2003.1238185.

27 Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 850–858. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00085.

28 Justin Holmgren, Alex Lombardi, and Ron D. Rothblum. Fiat-shamir via list-recoverable codes (or: parallel repetition of GMW is not zero-knowledge). In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 750–760. ACM, 2021. doi:10.1145/3406325.3451116.

29 James Hulett, Ruta Jawale, Dakshita Khurana, and Akshayaram Srinivasan. Snargs for P from sub-exponential DDH and QR. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 520–549. Springer, 2022. doi:10.1007/978-3-031-07085-3_18.

30 Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *FOCS*, pages 294–304. IEEE Computer Society, 2000. doi:10.1109/SFCS.2000.892118.

31 Abhishek Jain and Zhengzhong Jin. Non-interactive zero knowledge from sub-exponential DDH. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2021. doi:10.1007/978-3-030-77870-5_1.

32 Ruta Jawale, Yael Tauman Kalai, Dakshita Khurana, and Rachel Yun Zhang. Snargs for bounded depth computations and PPAD hardness from sub-exponential LWE. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 708–721. ACM, 2021. doi:10.1145/3406325.3451055.

33 Yael Tauman Kalai, Alex Lombardi, and Vinod Vaikuntanathan. Snargs and PPAD hardness from the decisional diffie-hellman assumption. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part II*, volume 14005 of *Lecture Notes in Computer Science*, pages 470–498. Springer, 2023. doi:10.1007/978-3-031-30617-4_16.

34 Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of fiat-shamir for proofs. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 224–251. Springer, 2017. doi:10.1007/978-3-319-63715-0_8.

35 Yael Tauman Kalai, Vinod Vaikuntanathan, and Rachel Yun Zhang. Somewhere statistical soundness, post-quantum security, and snargs. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part I*, volume 13042 of *Lecture Notes in Computer Science*, pages 330–368. Springer, 2021. doi:10.1007/978-3-030-90459-3_12.

36 Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *STOC*, pages 723–732. ACM, 1992. doi:10.1145/129712.129782.

37 Susumu Kiyoshima. Public-coin 3-round zero-knowledge from learning with errors and keyless multi-collision-resistant hash. In *CRYPTO (1)*, volume 13507 of *Lecture Notes in Computer Science*, pages 444–473. Springer, 2022. doi:10.1007/978-3-031-15802-5_16.

38 Alex Lombardi and Vinod Vaikuntanathan. Correlation-intractable hash functions via shift-hiding. In *ITCS*, volume 215 of *LIPIcs*, pages 102:1–102:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.102.

39 Silvio Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4):1253–1298, October 2000. doi:10.1137/S0097539795284959.

40 Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 89–114. Springer, 2019. doi:10.1007/978-3-030-26948-7_4.

41 David Pointcheval and Jacques Stern. Security proofs for signature schemes. In Ueli M. Maurer, editor, *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, volume 1070 of *Lecture Notes in Computer Science*, pages 387–398. Springer, 1996. doi:10.1007/3-540-68339-9_33.

42 Yuyu Wang and Jiaxin Pan. Non-interactive zero-knowledge proofs with fine-grained security. In Orr Dunkelman and Stefan Dziembowski, editors, *Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part II*, volume 13276 of *Lecture Notes in Computer Science*, pages 305–335. Springer, 2022. doi:10.1007/978-3-031-07085-3_11.