# Does Fiat-Shamir Require a Cryptographic Hash Function?*

Yilei Chen**, Alex Lombardi***, Fermi Ma†, and Willy Quach‡

**Abstract.** The Fiat-Shamir transform is a general method for reducing interaction in public-coin protocols by replacing the random verifier messages with deterministic hashes of the protocol transcript. The soundness of this transformation is usually *heuristic* and lacks a formal security proof. Instead, to argue security, one can rely on the *random oracle methodology*, which informally states that whenever a random oracle soundly instantiates Fiat-Shamir, a hash function that is "sufficiently unstructured" (such as fixed-length SHA-2) should suffice. Finally, for some special interactive protocols, it is known how to (1) isolate a concrete security property of a hash function that suffices to instantiate Fiat-Shamir and (2) build a hash function satisfying this property under a cryptographic assumption such as Learning with Errors.

In this work, we abandon this methodology and ask whether Fiat-Shamir truly requires a cryptographic hash function. Perhaps surprisingly, we show that in two of its most common applications — building signature schemes as well as (general-purpose) non-interactive zero-knowledge arguments — there are sound Fiat-Shamir instantiations using extremely simple and non-cryptographic hash functions such as sum-mod-$p$ or bit decomposition. In some cases, we make idealized assumptions (i.e., we invoke the generic group model), while in others, we prove soundness in the plain model.

On the negative side, we also identify important cases in which a cryptographic hash function is provably necessary to instantiate Fiat-Shamir. We hope this work leads to an improved understanding of the precise role of the hash function in the Fiat-Shamir transformation.

## 1 Introduction

The Fiat-Shamir transform is a general-purpose method for converting public-coin interactive protocols into *non-interactive* protocols with the same functionality. As a prototypical example, let $\Pi$ denote a 3-message (public-coin) argument system with transcripts of the form $(\alpha, \beta, \gamma)$. Then, given any *hash function*

* The full version of this paper is available [16].

** Tsinghua University. Email: chenyilei@mail.tsinghua.edu.cn.

*** MIT. Email: alexjl@mit.edu. Research supported in part by an NDSEG fellowship. Research supported in part by NSF Grants CNS-1350619 and CNS-1414119, and by the Defense Advanced Research Projects Agency (DARPA) and the U.S. Army Research Office under contracts W911NF-15-C-0226 and W911NF-15-C-0236.

† Princeton University and NTT Research. Email: fermima@alum.mit.edu.

‡ Northeastern University. Email: quach.w@husky.neu.edu.

$h$, the Fiat-Shamir transform of $\Pi$ using $h$, denoted $\Pi_{\mathrm{FS},h}$, is a one-message argument system in which the prover sends an entire transcript $(\alpha, \beta = h(\alpha), \gamma)$ in one shot.

The Fiat-Shamir transform was introduced by [27] to remove interaction from a 3-message identification scheme, but it was later realized[1] that the transformation is extremely general: it can plausibly be applied to *any* constant-round public-coin interactive argument system (and more). Due to its generality and its *practical efficiency* (it removes interaction with very low computational overhead), the transformation has been a cornerstone of both theoretical and practical cryptography for over 30 years. Some of its applications include the construction of efficient signature schemes [27, 52, 50], non-interactive zero-knowledge arguments (NIZKs) [1, 12, 11, 49], and succinct non-interactive arguments (SNARGs) [36, 43, 7, 2, 4, 3, 56, 6, 5].

However, the vast majority of applications of the Fiat-Shamir transform are only *heuristically sound*. That is, the resulting non-interactive protocols do not have proofs of soundness based on the computational intractability of a well-studied mathematical problem [32]. Nonetheless, the protocols appear to be sound in practice, so it has been a long-standing goal of theoretical cryptography to *justify* the soundness of the transformation.

So far, there have been two main approaches for justifying soundness of Fiat-Shamir.

- **The Random Oracle Model** [1]: In this design methodology, a Fiat-Shamir hash function is first modeled as a random function $\mathcal{O}$ to which all parties (honest and dishonest) have public query access. Security is "argued" by showing that the protocol $\Pi_{\mathrm{FS},\mathcal{O}}$ is sound "in the random oracle model" (i.e., against query-bounded adversaries). In reality, the hash function $h$ is instantiated by an "unstructured" hash function (such as SHA-2 on bounded-length inputs), where the implicit expectation is that "Fiat-Shamir for $\Pi$" is not an application that can distinguish $h$ from a random oracle.
- **Correlation Intractability**: In a recent line of work [34, 12, 33, 11, 49, 9, 37], a different methodology was developed for provably instantiating Fiat-Shamir in the standard model:
  - Identify a special class $\mathcal{C}$ of protocols and a cryptographic security property $\mathcal{P}$ of a hash function family $\mathcal{H}$ such that if $\mathcal{H}$ satisfies $\mathcal{P}$, then $\mathcal{H}$ soundly instantiates Fiat-Shamir for every $\Pi \in \mathcal{C}$. In all cases so far, $\mathcal{P}$ has been a restricted form of correlation intractability [13].
  - Construct a hash function family satisfying $\mathcal{P}$ under reasonable (hopefully standard) cryptographic assumptions.

The first of these approaches attempts to justify the use of Fiat-Shamir in high generality, while the second provides full security proofs for carefully chosen protocols and hash functions.

[1] See discussion in [1]

*Why Cryptographic Hash Functions?* In both approaches above, it is essential that the hash function $h$ possesses a form of *cryptographic hardness*. In the random oracle methodology, it is heuristically assumed that $h$ is indistinguishable from a truly random function (at least in any meaningful way), while in the standard model, results so far have relied on correlation-intractable hash families [47, 13] whose security can be based on standard cryptographic assumptions [11, 49, 9].

All of these results support the intuition that the Fiat-Shamir hash family $\mathcal{H}$ provides a form of cryptographic hardness that ensures the soundness of $\Pi_{\mathrm{FS},\mathcal{H}}$. In this work, we ask whether this intuition is accurate.

*Is it possible to instantiate the Fiat-Shamir heuristic with a* non-cryptographic *hash function?*

We note that this question requires formalizing what it means to be a "non-cryptographic" (rather than cryptographic) hash function; we partially address this issue later, but this remains somewhat up to interpretation.

A related question concerns the *design* of Fiat-Shamir hash functions. What should they look like? Again, prior works give us some possible answers:

- As originally proposed in [27], a Fiat-Shamir hash function could be instantiated using a pseudorandom function family [31] (they give DES as an example instantiation).
- As proposed in the random oracle methodology [1], the following design advice is given. "When instantiating a random oracle by a concrete function $h$, care must be taken first to ensure that it is adequately conservative in its design so as not to succumb to cryptanalytic attack, and second to ensure that $h$ exposes no relevant 'structure' attributable to its being designed from some lower-level primitive." In other words, the hash function should be *unstructured* and *complex* enough to be indistinguishable from a random function.
- In the provably secure instantiations of [11, 49], the hash function families are based on flavors of *fully homomorphic encryption*, which can be instantiated from lattice assumptions [29, 10].
- In a recent work of [9], a (modified) *trapdoor hash function* [25] is used, which has instantiations based on the DDH/LWE/QR/DCR assumptions.

A common theme is that all of the candidate Fiat-Shamir hash functions above are *complex*. Indeed, they have to be complex enough to realize the described security properties. In contrast, we ask:

*Is it possible to instantiate Fiat-Shamir with a* simple *hash function?*

As an example, can we hope to have a *linear* Fiat-Shamir hash function $h(x) = Ax + b$?

We note that for various contrived protocols $\Pi$, the answer is "yes" for uninteresting reasons. For example, given any constant-round, public-coin interactive protocol $\Pi$, there is a protocol $\tilde{\Pi}$ that replaces all prover messages $\alpha_i$

with random-oracle commitments $\mathcal{O}(\alpha_i)$ and requires the prover to open these commitments in the last round. For this protocol $\tilde{\Pi}$, even the identity function can be used to instantiate Fiat-Shamir in the random oracle model, since we have in effect *already* applied a random-oracle Fiat-Shamir transformation when converting $\Pi$ to $\tilde{\Pi}$.

To avoid these trivialities, we phrase our goal more specifically: for various *naturally occurring* protocols (or classes of naturally occurring protocols), determine if simple/non-cryptographic hash functions may suffice for Fiat-Shamir, and give principled justification for this possibility or impossibility.

### 1.1 Our Contributions

We begin the systematic study of instantiating Fiat-Shamir with simple and non-cryptographic hash functions. In particular, we focus on two common and important use cases of Fiat-Shamir:

1. Round-compressing 3-message identification schemes [27, 52, 40], and
2. Round-compressing 3-message honest-verifier zero knowledge argument systems to obtain NIZK arguments for $\mathsf{NP}$ [1, 12, 11, 49, 21, 17, 9].

For these two use cases, we identify some common 3-message protocols to which Fiat-Shamir is applied:

- Schnorr's identification scheme [52].
- The Chaum-Pedersen interactive proof system for the Diffie-Hellman language [15].
- Lyubashevsky's lattice-based identification scheme [40].
- More generally, $\Sigma$-protocols [23], which are typically repeated in parallel to obtain negligible soundness error.

In this work, we consider whether existing protocols from above can be round-compressed using a simple/non-cryptographic hash function. We are able to show both negative results and (perhaps surprisingly) *positive* results on this front.

Before stating our results more formally, we discuss (1) the specific problems we want to solve and (2) what constitutes a solution to the problem.

**Our Methodology** There are two major issues to resolve in order to define our problem:

(i) What does it mean for a hash function to be *cryptographic*?
(ii) How do we give evidence for the soundness (or lack thereof) of our round-compressed protocols?

We first partially address question (i). One appealing intuitive definition of a cryptographic hash function is as follows:

**Definition 1 (Cryptographic Hash Function, definition attempt).** *A hash function $h$ (or hash function family $\mathcal{H}$) is* cryptographic *if there is a game $\mathcal{G}$ between a challenger and adversary (who is given $h$ or $h \leftarrow \mathcal{H}$) with a* statistical-computational gap*; that is, the maximum probability that a computationally bounded adversary can win $\mathcal{G}$ is noticeably smaller than the maximum probability that an unbounded adversary can win $\mathcal{G}$.*

Unfortunately, this definition has major issues. In particular, under a literal interpretation of the definition, if $\mathsf{NP} \not\subset \mathsf{BPP}$, then *every* hash function is "cryptographic": just define the game $\mathcal{G}$ that ignores the hash family $\mathcal{H}$ and gives the adversary an instance of a hard $\mathsf{NP}$ problem to solve.

More specific to our application, the soundness of $\Pi_{\mathrm{FS},\mathcal{H}}$ is precisely a game with a computational-statistical gap so long as an accepting proof exists but is computationally hard to find. Therefore, no matter how "simple" or "non-cryptographic" $\mathcal{H}$ appears to be, as long as it can compile Fiat-Shamir for some protocol, it is necessarily "cryptographic" under this definition.

Indeed, an important philosophical point in this work is that the "computational hardness" within the soundness property of $\Pi_{\mathrm{FS},\mathcal{H}}$ can derive from two different places: the **hash family** $\mathcal{H}$ and the **interactive protocol** $\Pi$.

For our purposes, we appeal to the following intuitive (non-technical) definition of a cryptographic hash function:

**Definition 2 (Cryptographic Hash Function, intuition-level).** *Informally, a hash function $h$ (or hash function family $\mathcal{H}$) is* cryptographic *if there is a game $\mathcal{G}$ between a challenger and adversary with a* statistical-computational gap that does not derive from some separate hard problem*.*

Given this partial answer to question (i), we now describe how we handle (ii):

*How We Give Positive Results.* In order to obtain a positive result, we accomplish (at least) one of three things:

- We show that any hash function $h$ (or hash family $\mathcal{H}$) satisfying an *information-theoretic property* (e.g., pairwise-independence) suffices to instantiate $\Pi_{\mathrm{FS},\mathcal{H}}$ soundly. We believe that in spirit, this says that Fiat-Shamir for $\Pi$ does not require a cryptographic hash function (Definition 2), as a purely information theoretic property should be insufficient to establish computational hardness.
- We show that a *single fixed hash function* $h$ (rather than a distribution on hash functions) is enough to soundly instantiate $\Pi_{\mathrm{FS},h}$. More specifically, we show "average-case soundness", i.e., soundness on a random NO-instance. This is at least enough to strongly distinguish our Fiat-Shamir instantiations from random-oracle hash functions as well as correlation-intractable hash functions, which crucially rely on the randomness of the hash function to derive computational hardness.
- We instantiate $\Pi_{\mathrm{FS},h}$ with an *extremely simple* hash function $h$, such as a linear function modulo a prime $p$ or the bit decomposition function $\mathbf{G}^{-1}$ :

$\mathbb{Z}_q^n \to \mathbb{Z}_2^{n \log q}$. This does not directly prove that $h$ is not cryptographic, but it again distinguishes our constructions from prior work, in which the Fiat-Shamir hash functions are comparatively complex (see above). Indeed, they are sufficiently complex to guarantee security properties such as correlation intractability.

While some of our positive results hold in the standard model, others are shown to hold in the (auxiliary-input) generic group model [45, 53, 55, 18, 19]. One might ask why such a result is meaningful — after all, we are replacing one random oracle (the hash function) with another (the generic group labeling). However, the idealized assumptions in our constructions are used quite differently from assuming that a Fiat-Shamir hash function behaves like a random oracle. Indeed, our hash functions are information-theoretic and do not make any calls to the group oracle. As a result, our constructions are examples of *naturally occurring* interactive protocols $\Pi$ (unlike the contrived example from the introduction) that possess enough hardness to guarantee that $\Pi_{\mathrm{FS},h}$ is sound for *simple* choices of $h$ satisfying only information-theoretic properties.

Additionally, our lower bounds in the GGM suggest candidate schemes over concrete groups ($\mathbb{Z}_p^\times$ and elliptic curve groups) that are plausibly secure. Although interpreting hardness results in the GGM in the standard model requires care [28, 54, 24], we believe that it would be very interesting to understand the real-world security of the resulting (extremely simple!) schemes. We do some preliminary analysis of the concrete schemes — finding non-generic attacks for one of our two GGM-based protocols but not the other — but largely leave these questions open.

*How We Give Negative Results.* In order to obtain a negative result, we would like to show that for a particular protocol $\Pi$, if $\Pi_{\mathrm{FS},\mathcal{H}}$ is sound, then $\mathcal{H}$ necessarily satisfies some concrete cryptographic security property $\mathcal{P}$. However, as already discussed, such a theorem is not meaningful — $\mathcal{P}$ can just be "the soundness of $\Pi_{\mathrm{FS},\mathcal{H}}$." In other words, this fails to distinguish between hardness in the hash function family $\mathcal{H}$ from hardness in the protocol $\Pi$.

Instead, we switch the order of quantifiers in the theorem statement: we show that there exists a *universal* security property $\mathcal{P}$ such that for any protocol $\Pi \in \mathcal{C}$ in a large class, if a hash function family $\mathcal{H}$ soundly instantiates Fiat-Shamir for $\Pi$ then $\mathcal{H}$ necessarily satisfies $\mathcal{P}$. Since $\mathcal{P}$ is independent of the protocol $\Pi$, this comes closer to distinguishing $\mathcal{H}$-hardness from hardness in $\Pi$.

However, there is still one issue with the above strategy: $\mathsf{NP}$-completeness also gives a (trivial) universal property $\mathcal{P}$. To avoid this problem, we prove a *relativizing* result: the same property $\mathcal{P}$ is satisfied by $\mathcal{H}$ even if it instantiates Fiat-Shamir for various protocols $\Pi^{\mathcal{O}(\cdot)}$ that exist relative to an oracle distribution $\mathcal{O}$. This establishes that the property $\mathcal{P}$ is not "cheating" using $\mathsf{NP}$-completeness. As an example, our negative results will capture the $\{0,1\}$-challenge variant of Schnorr's identification scheme in the generic group model as well as Blum's Hamiltonicity protocol [8] instantiated in the random-oracle model.

Finally, we show that hash functions satisfying our property $\mathcal{P}$ imply the existence of one-way functions, the quintessential cryptographic object. This results in a formalization of the statement "one-way functions are necessary to instantiate Fiat-Shamir hash functions for natural protocols."

As an added bonus, we are also sometimes able to give direct attacks on $\Pi_{\mathrm{FS},\mathcal{H}}$ relative to an oracle (i.e., in the generic group model or the random oracle model). That is, for the idealized protocols, we show unconditional polynomial-query attacks on the non-interactive protocol. This is further evidence that a sound Fiat-Shamir instantiation must sometimes rely on hardness from the hash function family $\mathcal{H}$, in direct contrast to our positive results.

**Our Results** With the above discussion in mind, we are now ready to formally state our results. First, we give several positive results for soundly instantiating Fiat-Shamir with *non-cryptographic* hash functions.

*Fiat-Shamir for Lattice-Based Identification Schemes.* We first describe our positive results in the standard model, which hold for lattice-based analogues of the Schnorr protocol. In particular, we consider common variants of Lyubashevsky's identification schemes [38, 39, 40], which were designed to obtain efficient signature schemes in the random oracle model via Fiat-Shamir.

We obtain a sound Fiat-Shamir instantiation for the main protocol $\Pi$ defined in [40]. Our Fiat-Shamir hash function in $\Pi_{\mathrm{FS},h}$ maps $\mathbb{Z}_q$ elements to their bit-decomposition (also known as the $\mathbf{G}^{-1}$ function).

**Theorem 1.** *Consider Lyubashevsky's identification scheme over $\mathbb{Z}_q$ in dimension $n$. Define the hash function $h : \mathbb{Z}_q^n \to \mathbb{Z}_2^{n \log q}$ as the bit decomposition function*

$$h(v) = \mathbf{G}^{-1}(v).$$

*Then, under the Short Integer Solution (SIS) assumption, Fiat-Shamir applied to Lyubashevsky's scheme using hash function $h$ is sound on random instances.*

We note the following interesting details about our result.

- We obtain a meaningful soundness guarantee using a **deterministic hash function**. This stands in contrast to typical Fiat-Shamir instantiations.
- More generally, we prove Theorem 1 for a class of Fiat-Shamir hash functions (including bit-decomposition) satisfying an **information-theoretic property**.
- Most importantly, and uniquely to the lattice setting, we emphasize that soundness is proved in the **standard model**! More specifically, the SIS assumption suffices to argue *average-case* soundness, where soundness requires that a cheating prover cannot convince a verifier to accept on a random instance. We stress that this is the typical soundness notion for the setting of identification/signature schemes and a necessary relaxation for the case of deterministic hash functions.

To contrast this with prior work on Fiat-Shamir in the standard model [34, 12, 11, 49, 9], we note that (1) it was not known how to do Fiat-Shamir for the [40] protocol in the correlation intractability framework, and (2) our Fiat-Shamir compiler uses the bit decomposition function and *not* any form of CI.

Finally, as an extension of Theorem 1, we prove that variants of our protocol $\Pi_{\mathrm{FS}}$ show a surprising connection to Micciancio-Peikert lattice trapdoors [44, 41]. Namely, the prover algorithm in $\Pi_{\mathrm{FS}}$ can be interpreted as a preimage sampling algorithm using a Micciancio-Peikert trapdoor.

**Theorem 2 (Informal).** *Lattice-based Lyubashevsky signatures using the bit-decomposition Fiat-Shamir hash function are equivalent to lattice-based Hash-and-Sign signatures.*

This highlights a strong connection between two seemingly orthogonal paths to build signatures from lattice-based assumptions: one using lattice trapdoors [30, 14, 44] and the other through the Fiat-Shamir heuristic [38, 39, 40]. To the best of our knowledge (see [48]), no such connection was known before. We discuss this connection in more detail in the technical overview.

*Schnorr Signatures with a Linear Fiat-Shamir Hash Function.* Our next result concerns the Schnorr signature scheme, obtained by applying Fiat-Shamir to Schnorr's three-message protocol for proving knowledge of a discrete logarithm. We show that for signing *short* messages (i.e. the message space is a sparse subset of $\mathbb{Z}_p$), this classic application of the Fiat-Shamir paradigm does not seem to require any cryptographic properties from the underlying Fiat-Shamir hash function.

Recall that the Schnorr protocol works over a cryptographic group $G$ of order $p$, and that the Fiat-Shamir hash function takes as input a group element $g \in G$ along with a message $m \in \mathcal{M}$ to be signed, and outputs an element in $\mathbb{Z}_p$.

**Theorem 3 (Schnorr Signatures with a $\mathbb{Z}_p$-Linear Hash Function).** *Consider the Schnorr signature scheme over a group $G$ of order $p$, where the message space $\mathcal{M}$ is a sparse subset of $\mathbb{Z}_p$, i.e. $\mathcal{M} \subset \mathbb{Z}_p$ and $|\mathcal{M}|/\mathbb{Z}_p \leq \mathrm{negl}(\lambda)$. Let $\ell$ be the maximum bit-length representation of any group element, so that any $g \in G$ can be viewed as $g \in \{0,1\}^\ell = [2^\ell]$. Define the hash family*

$$h_k(g, m) \coloneqq g + m + k \pmod{p},$$

*where on the right-hand side, $g$ is the integer with binary representation $g \in \{0,1\}^\ell$.*

*In the auxiliary-input generic group model [55], the Schnorr signature scheme instantiated using $h$ as the Fiat-Shamir hash function is existentially unforgeable against chosen message attacks (EUF-CMA).*

As in the lattice setting, we can actually prove that Fiat-Shamir for Schnorr is sound whenever $h$ (or the family $\mathcal{H}$) satisfies an information-theoretic property. However, our security proof relies on the GGM and does not seem to carry over

to the standard model. Nonetheless, we view Theorem 3 as another interesting example of a Fiat-Shamir instantiation whose soundness does not rely on any cryptographic property of the hash function. Instead, **strong cryptographic hardness from the group turns out to be sufficient!**

Another takeaway from Theorem 3 is that Schnorr-like signatures can plausibly be obtained by combining a collision-resistant hash function (to implement hash-and-sign) with an information-theoretic Fiat-Shamir hash function (for Schnorr signatures on short messages). While this does not appear significantly different from using a cryptographic Fiat-Shamir hash function *in implementation*, it highlights the fact that cryptographic hashing is required for signatures only to (computationally) avoid *collisions* between long messages, and *not* for ensuring soundness of the Fiat-Shamir compilation.

*Aside on Generic Groups.* The Generic Group Model [53] models a cryptographic group $G$ as a random injection $G \to [L]$ for a sufficiently large "label space" $L$, by providing an oracle $\mathcal{O}$ that computes group products and inverses on (pairs of) labels.[2] The auxiliary-input GGM [55, 18] gives the adversary the additional power to *record* an arbitrary ($S$-bounded) function of the group's truth table to use for solving computational problems later.

In the plain GGM, soundness of our variant of Schnorr signatures follows from analysis due to [46]; this work characterized a security property of $\mathcal{H}$ that suffices for (long-message) signatures schemes in the GGM. For our purposes, it turns out that an *information-theoretic* property of $h$ suffices; see Section 2 for details. In fact, using the even simpler (keyless) function $h(g, m) = g + m$ is secure in the GGM.

However, since soundness is proved in the GGM, it is reasonable to ask whether the hardness result plausibly translates to concrete groups such as $\mathbb{Z}_p^\times$ or elliptic curve groups. Indeed, it is known that GGM lower bounds sometimes fail to carry over to these groups in cases of interest (see, e.g., [28, 54]). In this work, we observe that this issue *also* comes up in the case of Schnorr signatures as analyzed by [46]. In more detail, [46] proves that as long as a hash family $\mathcal{H}$ satisfies two (possibly computational) properties, then Schnorr signatures using $\mathcal{H}$ are secure in the GGM. On the other hand, we find choices of $\mathcal{H}$ that satisfy the premises of [46], but attacks exist over *all concrete groups*. This highlights an important situation where GGM-based analysis spectacularly fails to capture real-world attacks on a scheme.

On the other hand, we further observe that these non-generic attacks can be captured by the auxiliary-input GGM; that is,

[2] There is an alternative formulation of a Generic Group Model due to Maurer [42], but the *honest parties* in Schnorr's signature scheme execute non-generic algorithms according to this definition (since Maurer's GGM does not provide concrete representations of group elements, which are necessary to evaluate the Fiat-Shamir hash function), so a [42]-generic analysis is not applicable.

– Given some (possibly hard-to-compute) short piece of information $w$ about $G$ (but independent of the Schnorr public parameters), Schnorr signatures using $\mathcal{H}$ are insecure, **and**
– Over important concrete groups such as $\mathbb{Z}_p^\times$ or elliptic curve groups, this information $w$ is actually efficiently computable.

For example, the short information could be a solution $z$ to the equation $a^z = \ell$, where $\ell \in [L]$ is a fixed label such that $\ell \equiv -1 \pmod p$. To remedy this problem, we prove a lower bound in the aux-input GGM, thus avoiding an important class of "non-generic" attacks for the hash function in Theorem 3 (and more). This proof is the new technical component of Theorem 3.

In fact, we know of no efficient attacks on the scheme from Theorem 3 over the group $\mathbb{Z}_p^\times$. We find the question of whether this scheme is secure to be interesting, as it would result in a signature scheme that is extremely simple to write down — in fact, key generation, signing, and verifying only require random sampling and arithmetic over $\mathbb{Z}_p$. We do some preliminary analysis of the scheme in the full version but leave the question largely out of the scope of this paper.

*The Chaum-Pedersen Protocol and NIZKs for* NP. Next, we consider a minor variant of the interactive proof system due to Chaum and Pedersen [15] for proving membership in the Diffie-Hellman language $\mathcal{L}_{\mathrm{DH}} := \{(g, g^u, g^v, g^{uv})\}_{g \in G, u, v \in \mathbb{Z}_p}$. The protocol was originally introduced to instantiate a (special-purpose) blind signature scheme, but it has since found other applications (e.g., to the Cramer-Shoup cryptosystem [22]). Notably, a recent line of work [20, 35, 51, 21] has shown that a non-interactive, adaptively sound, (single-theorem) zero-knowledge argument for $\mathcal{L}_{\mathrm{DH}}$ (along with CDH) suffices to instantiate non-interactive zero-knowledge (NIZK) arguments for all of NP.

We prove in the (auxiliary-input) GGM that a simple, fixed Fiat-Shamir hash function $h$ suffices to compile the modified[3] Chaum-Pedersen protocol into an argument for $\mathcal{L}_{\mathrm{DH}}$ satisfying an intermediate (i.e., in between selective and adaptive) notion of soundness we call *semi-adaptive* soundness. Here, the prover is given a random $g^u$, and wins if it convinces the verifier to accept a NO-instance of $\mathcal{L}_{\mathrm{DH}}$ of the form $(g, g^u, g^y, g^z)$.

**Theorem 4.** *Let $\Pi^{CP}$ denote the modified Chaum-Pedersen protocol over a group $G$ of order $p$. Let $\ell$ be the maximum bit-length representation of any group element, so that any $g \in G$ can be viewed as $g \in \{0,1\}^\ell = [2^\ell]$. Define the hash function*

$$h(g_1, g_2, g_3, g_4) = g_1 + g_2 + g_3 + g_4 \pmod p,$$

*where on the right-hand side, each $g_i$ is the integer with binary representation $g_i \in \{0,1\}^\ell$.*

*In the auxiliary-input generic group model, $(\Pi^{CP})_{FS,h}$ is a semi-adaptively sound argument system for $\mathcal{L}_{\mathrm{DH}}$.*

[3] Our modification simply requires the verifier to reject if the third message $z$ is equal to $0 \in \mathbb{Z}_p$.

In the full version, we prove a stronger result: as long as $h$ satisfies an (easily satisfied but complicated to state) information theoretic property, $(\Pi^{\mathrm{CP}})_{\mathrm{FS},h}$ is sound in the aux-input GGM.

By tweaking the hash function to be $h'(\cdot) := h(\cdot) + r$ where $r$ is a common random string, $(\Pi^{\mathrm{CP}})_{\mathrm{FS},h'}$ becomes a (single-theorem) NIZK argument for $\mathcal{L}_{\mathrm{DH}}$ with semi-adaptive soundness. It turns out that semi-adaptive soundness suffices to instantiate the hidden bits model of [26], and consequently NIZKs for $\mathsf{NP}$ in the standard model [20, 35, 51, 21].

However, we also cryptanalyze this protocol over concrete groups such as $\mathbb{Z}_p^\times$ and elliptic curve groups (see the full version), and unlike the case of Schnorr signatures above, we find non-generic attacks (that fall outside the aux-input GGM) on the scheme. Thus, Theorem 4 should be viewed as a theoretical result that does *not* have direct implications over commonly used groups. This disparity between the GGM and the standard model appears to be quite subtle and deserves further study, as further discussed in our conclusion (Section 1.2).

*Negative Results.* To complement our positive results, we also show that for some protocols, Fiat-Shamir necessarily requires a cryptographic hash function. Our negative results apply to a large class $\mathcal{C}$ of **three-message honest-verifier zero-knowledge (HVZK) arguments** (or proofs), in particular, those obtained by taking parallel repetitions of sigma protocols with polynomial-size challenge space. Two prototypical examples to have in mind are:

- Blum's Hamiltonicity protocol [8], repeated in parallel to obtain negligible soundness error.
- The one bit challenge variant $\Pi^{\mathrm{bit-Sch}}$ of Schnorr's identification scheme, again repeated in parallel.

We analyze Fiat-Shamir for these protocols in **both** the standard model and in idealized models (the random-oracle model and the preprocessing GGM, respectively). We give evidence that analogues to Theorem 3, Theorem 4, and Theorem 1 *do not exist* for these protocols. Our two results are as follows.

- **Polynomial-Query Attacks**: First, we show that in idealized models, there will (unconditionally) be a polynomial-query attack on $\Pi_{\mathrm{FS},\mathcal{H}}$, *as long as* $\mathcal{H}$ *does not depend on the oracle*. In other words, a (poly-query) sound Fiat-Shamir instantiation requires that $\mathcal{H}$ depends on the oracle, which is one way of arguing that $\mathcal{H}$ is cryptographic.

  **Theorem 5 (Informal).** *For $\Pi = \Pi^{\mathrm{bit-Sch}}$ instantiated in the generic group model, if $\mathcal{H}$ is a hash family that does not call the group oracle, then $\Pi^{\mathrm{t}}_{\mathrm{FS},\mathcal{H}}$ is unsound in the GGM.*
  *For* any *instantiation of the [8] protocol in the random oracle model, if $\mathcal{H}$ is a hash family that does not depend on the oracle $\mathcal{O}$, then $\Pi_{\mathrm{FS},\mathcal{H}}$ is unsound. More generally, for any $\Pi \in \mathcal{C}$ constructed relative to an oracle $\mathcal{O}$, if $\mathcal{H}$ does not depend on $\mathcal{O}$, then $\Pi_{\mathrm{FS},\mathcal{H}}$ is unsound.*

  This is in contrast to Schnorr/Chaum-Pedersen reuslts, in which an oracle-independent hash function suffices for a sound Fiat-Shamir instantiation.

*Generalization: What is the class $\mathcal{C}$?* In full generality (see the full verseion), the class $\mathcal{C}$ of protocols $\Pi$ for which we give a polynomial-query attack on $\Pi_{\mathrm{FS},\mathcal{H}}$ is informally characterized as follows.

- $\Pi := \Pi^t_{\mathrm{Base}}$ is the parallel repetition of a 3-message public-coin HVZK argument system $\Pi_{\mathrm{Base}} = \Pi^{\mathcal{O}(\cdot)}_{\mathrm{Base}}$ (with simulator $\mathsf{Sim}$) relative to an oracle $\mathcal{O}$.
- The Verifier's challenge space $\Sigma$ in $\Pi_{\mathrm{Base}}$ is polynomial-size.
- The underlying language $L \notin \mathsf{BPP}$.
- $(\Pi_{\mathrm{Base}}, \mathsf{Sim})$ is challenge hiding (see the full verseion).

The last requirement (challenge hiding) is a technical condition that slightly strengthens the standard notion of HVZK.

We emphasize that our result makes no assumptions about the way in which the oracle $\mathcal{O}$ is used in the construction of the interactive protocol $\Pi_{\mathrm{Base}}$. The most substantial requirement is that $\Pi$ is the result of *parallel repetition* applied to a protocol with a small (i.e., polynomial) challenge space. This property distinguishes the protocols that we can attack from the protocols for which we find sound Fiat-Shamir instantiations.

– **Conditional Polynomial-time Attacks and Mix-and-Match Resistance**: We describe a concrete security property (which we call "mix-and-match resistance" ) such that for any protocol $\Pi$ in a large class $\mathcal{C}'$ (again including the two example protocols above, *in the standard model*), any hash function (family) $\mathcal{H}$ that instantiates Fiat-Shamir for $\Pi$ must possess this security property. In other words, we show:

**Theorem 6 (Informal).** *If $\mathcal{H}$ is* not *mix-and-match resistant, then for any $\Pi \in \mathcal{C}$, there is a polynomial-time attack on the soundness of $\Pi_{\mathrm{FS},\mathcal{H}}$.*

At a high level, mix-and-match resistance is a security property asserting the hardness of finding a *combination* of many partial inputs that hashes to a corresponding *combination* of prescribed outputs. We also show that mix-and-match resistant hash functions imply the existence of OWFs. Therefore, Theorem 6 implies that (in the setting above) if $\Pi_{\mathrm{FS},\mathcal{H}}$ is sound, then $\mathcal{H}$ can be used to build a OWF (obliviously to the protocol $\Pi$).

This result also holds in the ROM and the GGM, in the sense that if $\mathcal{H}$ does not depend on the oracle $\mathcal{O}$ and is *not* mix-and-match resistant, then the polynomial-query attack from Theorem 5 can be upgraded to a polynomial-time attack. As discussed above, this further establishes that the "mix-and-match resistance" property of $\mathcal{H}$ is not "borrowing hardness" from the protocol $\Pi$, since our analysis applies to protocols whose security is unconditional.

Somewhat orthogonally, one might wonder whether mix-and-match resistant hash functions (as introduced in this work) are known to exist under standard cryptographic assumptions. The works of [11, 49] tell us that the answer is "yes," because they give a standard-model instantiation of Fiat-Shamir for a protocol $\Pi \in \mathcal{C}$ under standard assumptions. In the full verseion , we explore this connection further by showing that correlation-intractable hash functions (as constructed by [11, 49]) suffice to instantiate Fiat-Shamir for (a variant of) the *idealized* Blum protocol.

### 1.2 Conclusions

One of the main takeaways of this work is that our title question "Does Fiat-Shamir require a cryptographic hash function?" is surprisingly deep and difficult to resolve. We believe that our positive and negative results improve our understanding of the ground truth and point to fascinating new research directions.

Before now, the prevailing intuition was that for any natural protocol (Schnorr, Lyubashevsky, Blum, etc.), sound Fiat-Shamir compilation necessitates a carefully-constructed *cryptographic* hash function. In this methodology, the soundness of Fiat-Shamir has been argued by either (1) treating the hash function as a random oracle or (2) invoking some concrete security property of the function family. That is, the computational hardness of some problem derived from $H$ guarantees the soundness of the protocol.

In this work, we argue soundness of Fiat-Shamir (for certain protocols) by using an *information-theoretic* property of $H$ together with cryptographic hardness from the interactive protocol. Despite the caveats in our results, the conceptual point is clear: it is possible to prove meaningful notions of soundness for a Fiat-Shamir protocol by using security properties of the interactive protocol itself *instead* of security properties of the hash function.

Moreover, the instantiations of our positive results have noticeable qualitative differences from prior approaches to Fiat-Shamir, such as being able to use a *single* hash function $h$ (rather than a family), much simpler hash functions, and ones that contain no associated cryptographic hardness. This constrasts strongly with how we usually think of Fiat-Shamir; essentially all prior work required that the hash function be complex and/or cryptographic.

On the other hand, we also show (and formalize a way to show) that some protocols *do* require a cryptographic Fiat-Shamir hash function. This implies that the ground truth is complicated and hard to characterize, but in our view, worth understanding.

*What about Fiat-Shamir in Practice?* Since Schnorr signatures are heavily used in practice, one might ask how our positive results over groups relate to the use of Fiat-Shamir over concrete groups. The answer to this question crucially depends on how accurately the generic group model (with preprocessing) reflects the concrete security of these protocols.

While generic group analysis is often considered to be a meaningful reflection of real-world attacks, we discovered multiple non-generic attacks on Fiat-Shamir protocols over groups. Such attacks are therefore not covered by prior generic analyses such as [46].

- In the case of Schnorr signatures over $\mathbb{Z}_p^\times$, all of the new attacks we found were captured by the *preprocessing* generic group model, and so our new analysis in the preprocessing model rules out all such attacks on many variants of Schnorr signatures. Therefore, we view our positive results for Schnorr as a first step towards finding secure simple variants of Schnorr signatures, such as the candidate given in Construction 11.

– On the other hand, we have already discovered attacks (see the full version) on certain variants of our Chaum-Pedersen protocol over groups such as $\mathbb{F}_p^\times$, even in settings where we have a valid (preprocessing) generic group analysis.

This results in a bizarre state of affairs in which it is unclear how to interpret generic group analyses for Fiat-Shamir protocols over groups; this deserves future attention and cryptanalytic effort. Nonetheless, we consider the conceptual contributions of these aux-input GGM analyses to be valuable whether they turn out to reflect real-world attacks or not.

*Future Work.* We believe that our framework can serve as a potential complement to the correlation intractability framework for provable Fiat-Shamir soundness. Towards this end, we broadly ask,

*Which interactive protocols allow for "simple" Fiat-Shamir compilers?*

To start with, we consider differences between the protocols in our positive and negative results. Heuristically, we note that all protocols in our positive results achieve negligible soundness error using a *single non-separable large challenge*. In contrast, the separability of the challenge in the parallel repetition of a $\Sigma$-protocol appears to necessitate using a cryptographic hash function.

In this context, our contributions are a starting point for a more precise understanding of *when* hardness is required from a Fiat-Shamir hash function.

## 2 Technical Overview

We give an overview of our positive results for lattice-based identification protocols in Section 2.1 and our positive results for group-based protocols in Section 2.2. We then describe some of our negative results in Section 2.3.

### 2.1 A Non-Interactive Lattice-Based Identification Scheme

We describe how we obtain positive results in the lattice setting (Theorem 1). We consider Lyubashevky's three-message identification protocol [40], which can be seen as a lattice analogue to the Schnorr protocol.

To sample an instance for the protocol, we sample a uniformly random wide matrix $\mathbf{A}$ over $\mathbb{Z}_q$ along with a wide matrix $\mathbf{R}$ with random small entries. The shared instance is $(\mathbf{A}, \mathbf{Y} = \mathbf{AR} \mod q)$, and the prover's goal is to convince the verifier it knows a short $\mathbf{R}$ satisfying $\mathbf{AR} = \mathbf{Y} \mod q$.

The interactive protocol $\Pi$ then executes as follows:

– The prover samples a short vector $\mathbf{t}$ and sends $\boldsymbol{\alpha} \coloneqq \mathbf{At} \mod q$.
– The verifier responds by sending a random vector $\mathbf{c}$ with small entries.
– The prover responds with $\mathbf{z} \coloneqq \mathbf{t} + \mathbf{Rc}$.
– The verifier accepts if $\mathbf{A} \cdot \mathbf{z} = \boldsymbol{\alpha} + \mathbf{Y} \cdot \mathbf{c} \mod q$ and $\mathbf{z}$ is short.

As in [40], this interactive protocol is average-case sound under the SIS assumption. We now analyze the non-interactive protocol $\Pi_{\mathrm{FS},\mathbf{h}}$ for a (vector-valued) Fiat-Shamir hash function $\mathbf{h}$. A malicious prover attacking the average-case soundness of $\Pi_{\mathrm{FS},\mathbf{h}}$ must solve the following problem.

- **Input:** Random matrices $(\mathbf{A}, \mathbf{Y})$ and the description of a (vector-valued) hash function $\mathbf{h}$.[4]
- **Output:** Vectors $\boldsymbol{\alpha}, \mathbf{z}$ such that $\mathbf{A} \cdot \mathbf{z} = \boldsymbol{\alpha} + \mathbf{Y} \cdot \mathbf{h}(\boldsymbol{\alpha}) \mod q$ and $\mathbf{z}$ is short.

Our main insight is that this problem is provably hard for a fixed Fiat-Shamir hash function $\mathbf{h}$ if simple information-theoretic conditions are satisfied.

**Theorem 7.** *Suppose* $\mathbf{h}$ *satisfies the following properties:*

1. $\mathbf{h}$ *produces "short" output, i.e, the entries are small relative to the modulus*
2. $\boldsymbol{\alpha}$ *is a* linear *function of* $\mathbf{h}(\boldsymbol{\alpha})$*, i.e. there exists a matrix* $\mathbf{G}$ *such that for all* $\boldsymbol{\alpha}$*,* $\mathbf{G} \cdot \mathbf{h}(\boldsymbol{\alpha}) = \boldsymbol{\alpha} \mod q$*.*

*Then,* $\Pi_{\mathrm{FS},\mathbf{h}}$ *is one-time (average-case) sound.*

Theorem 7 can be proved as follows. If the condition in Theorem 7 are satisfied, then the relation $\mathbf{A} \cdot \mathbf{z} - \boldsymbol{\alpha} - \mathbf{Y} \cdot \mathbf{h}(\boldsymbol{\alpha}) = \mathbf{0} \mod q$ checked by the verifier can be rewritten as

$$\left[\mathbf{A} \| \mathbf{Y} + \mathbf{G}\right] \cdot \begin{bmatrix} \mathbf{z} \\ -\mathbf{h}(\boldsymbol{\alpha}) \end{bmatrix} = \mathbf{0} \mod q. \tag{1}$$

Since $\mathbf{A}, \mathbf{Y}$ are (statistically) uniformly random and $\mathbf{z}, \mathbf{h}(\boldsymbol{\alpha})$ are short, a malicious prover outputting $\boldsymbol{\alpha}, \mathbf{z}$ is solving SIS for the random matrix $[\mathbf{A} \| \mathbf{Y} + \mathbf{G}]$.

A simple concrete instantiation of $\mathbf{h}$ is the bit-decomposition function that maps (vectors of) $\mathbb{Z}_q$ elements to (the concatenation of) their bit decomposition in $\{0,1\}^{\lceil \log q \rceil}$ (also called $\mathbf{G}^{-1}(\cdot)$ in the lattice literature). The corresponding $\mathbf{G}$ is the "powers-of-two" gadget matrix of Micciancio-Peikert [44].

*Connections to Lattice Signatures from Lattice Trapdoors.* Interestingly, it turns out the honest prover algorithm of the rejection sampling-based protocol *exactly* matches the trapdoor preimage sampling algorithm of Lyubashevsky-Wichs [41] using a Micciancio-Peikert trapdoor [44]. This can be seen by considering Eq. (1), which implies that the transcript of the protocol gives a short preimage of $\mathbf{0}$ of a matrix with a Micciancio-Peikert trapdoor (here $\mathbf{R}$). Average-case soundness implies that this should be hard to do without knowledge of $\mathbf{R}$ (further using that $[\mathbf{A} \| \mathbf{AR} + \mathbf{G}]$ looks uniformly random over the randomness of $\mathbf{R}$), and witness-indistinguishability implies that the preimage sampling algorithm reveals no more information about the trapdoor $\mathbf{R}$.

In fact, our protocol shows the connection between seemingly orthogonal paths to obtain signatures from lattice-based assumptions: one relying on lattice trapdoors and trapdoor preimage sampling [30, 44, 41] and another through

[4] $\mathbf{Y}$ is technically sampled as $\mathbf{A} \cdot \mathbf{R}$ for some a "short" matrix $\mathbf{R}$, but parameters are set so that $\mathbf{Y}$ is statistically close to uniform.

Fiat-Shamir [38, 39, 40]. The lattice signature schemes constructed from lattice trapdoors [30, 44, 41] can actually be *derived* by applying the Fiat-Shamir heuristic (with aborts) using the bit-decomposition function (namely $\mathbf{G}^{-1}(\cdot)$) as the hash function to Lyubashevsky's three-message identification scheme [40]. Let us start by describing the signature scheme for signing a short random message $\mathbf{v} \in \mathbb{Z}_q^n$. The Fiat-Shamir hash function takes as input the first message $\boldsymbol{\alpha}$ from the protocol, and the message $\mathbf{v}$, and outputs

$$h(\boldsymbol{\alpha}, \mathbf{v}) = \mathbf{G}^{-1}(\boldsymbol{\alpha} - \mathbf{v}).$$

The signature consists of the challenge $\mathbf{c} = \mathbf{G}^{-1}(\boldsymbol{\alpha} - \mathbf{v})$ and $\mathbf{z}$ from the third message of the protocol. The verifier of the signature takes $\mathbf{v}$ and its signature, and accepts if $\mathbf{A} \cdot \mathbf{z} = \boldsymbol{\alpha} + \mathbf{Y} \cdot \mathbf{c} \mod q$ and $\mathbf{z}$ is short, that is:

$$[\mathbf{A} \,\|\, \mathbf{G} + \mathbf{Y}] \begin{bmatrix} \mathbf{z} \\ -\mathbf{c} \end{bmatrix} = \mathbf{v} \pmod{q}. \tag{2}$$

We now argue that this gives a signature scheme for random (short) messages, where the adversary can receive signature of random messages, and seeks to forge a signature for a random message given by the challenger. To handle signing queries, one can sample $(\mathbf{z}, \mathbf{c})$, and set the message as $\mathbf{v} = [\mathbf{A} \,\|\, \mathbf{G} + \mathbf{Y}] \begin{bmatrix} \mathbf{z} \\ -\mathbf{c} \end{bmatrix}$.

Then, the hardness of signing a random message $\mathbf{v}$ is then equivalent to breaking the SIS problem for a random target $\mathbf{v}$. To sign an arbitrary long message $\mu$, we replace $\mathbf{v}$ in the previous protocol by $H(\mu)$ where $H$ is a random oracle. This exaclty recovers the trapdoor-based lattice signatures [30, 44, 41] in the random oracle model. We stress that here, the only purpose of the random oracle is to compress the message (in a hash-and-sign manner), as opposed to collapse an interactive protocol. In particular the Fiat-Shamir hash function is still the non-cryptographic $\mathbf{G}^{-1}$ function.

### 2.2 Fiat-Shamir for Schnorr in the Generic Group Model.

The following section on the generic group model (GGM) contains a number of technical arguments, designed to motivate and provide intuition for our group-based results. We provide a roadmap for the discussion:

1. First we explain why Fiat-Shamir for Schnorr is secure in the (plain) GGM, even for simple, information-theoretic hash functions. We start with the case of "no-message" signatures (non-interactive identification) and then extend our reasoning to handle messages and signing queries.
   We remark that our security claims for Schnorr in the *plain* GGM could have been proven using prior analysis of [46]. However, we have two reasons for "re-doing" the analysis here: (1) our goal is to provide clear intuition tailored to *information-theoretic* Fiat-Shamir hash functions, and (2) our analysis will readily extend to the auxiliary-input setting, which we motivate next.

2. We will demonstrate that for Schnorr signatures, a (plain) GGM security proof does not capture a class of non-uniform attacks that work on *any concrete group*. In fact, we show that for common groups such as $\mathbb{Z}_p^*$, these attacks do not even require non-uniform advice.
3. We address these issues by extending our analysis to hold in the *auxiliary-input* GGM, albeit for a slightly more restricted class of Fiat-Shamir hash functions. We show this class still contains simple, information-theoretic hash functions, and we discuss potential implications of these results.

*Non-Interactive Identification in the Generic Group Model.* We begin by considering the classic Schnorr protocol for proving knowledge of a discrete logarithm. Recall that the protocol relies on a cryptographic group $G = \langle g \rangle$ of prime order $p$. The prover and verifier share an instance $g^u$ for a random $u$ known to the honest prover, and engage in the following interaction:

- The prover samples a random $r \leftarrow \mathbb{Z}_p$ and sends $g^r$.
- The verifier replies with a random $c \leftarrow \mathbb{Z}_p$.
- The prover sends $z = r + cu$.
- The verifier accepts if $g^z = (g^r)(g^u)^c$.

To build intuition, we will try to construct a (one-time secure) non-interactive identification scheme using a simple Fiat-Shamir hash function. In a moment, we will extend this (to handle messages and signing queries) to build full-fledged digital signatures.

For a Fiat-Shamir hash function $h$, a malicious prover for the non-interactive Schnorr protocol must solve the following problem.

- **Input:** A group description $G = (g, p)$, a hash function $h : G \to \mathbb{Z}_p$, and a random group element $g^u$.
- **Output:** $g^r, z$ satisfying $g^z = (g^r)(g^u)^{h(g^r)}$.

We want to identify simple choices of $h$ that make this problem hard in the GGM. However, it will be illuminating to instead identify which choices of $h$ will make this problem *easy*.

This problem is clearly easy if $h$ is a constant function, i.e. $h(g^x) = c$ for all $g^x$; the malicious prover could always win by outputting $z = 0$ and $g^r = ((g^u)^c)^{-1} = g^{-uc}$. Taking this a step further, we can argue that for any constant $c \in \mathbb{Z}_p$, the hash function $h$ should not output $c$ on a $1/\text{poly}(\lambda)$ fraction of its inputs. Otherwise, a malicious prover can pick a random $z$ and set $g^r = g^{-uc+z}$. Since $g^r$ is distributed randomly, $h(g^r) = c$ holds with $1/\text{poly}(\lambda)$ probability, in which case $z, g^{-uc+z}$ is a solution.

Put another way, as long as the min-entropy of $h$ on a random input is $O(\log(\lambda))$, the above is a completely generic method (i.e. one that works on any cyclic group) for breaking the resulting non-interactive protocol.

It turns out that this simple class of $h$ — those functions which, on random inputs, produce a low min-entropy output — are the *only* hash functions for which generic group algorithms (in the sense of Shoup [53]) exist to solve the above problem. That is, all hash functions $h$ with super-logarithmic min-entropy can be proven to soundly compile non-interactive Schnorr in the GGM:

**Theorem 8.** *In the generic group model (GGM), the non-interactive Schnorr protocol is one-time secure provided $h(\cdot)$ on a random input has entropy $\omega(\log \lambda)$.*

Recall that in the generic group model, group elements $g^x$ are replaced by labels $\sigma(x)$ where $\sigma$ is a random injection from $\mathbb{Z}_p$ to an exponentially-larger label space $[L]$ (say of size $\Omega(p^3)$, where $p$ itself is a $\lambda$-bit prime). The attacker interacts with an oracle (who knows the truth table of $\sigma$) to perform honest group operations such as raising a group element to a known exponent, performing the group operation on any two group elements, and taking the inverse.

In this model, the only way an attacker can output a valid group label $\sigma(r)$ is to obtain this label from oracle queries (with overwhelming probability, any other label it might choose to output will not have a preimage). Furthermore, if the attacker is initialized with $\sigma(1), \sigma(u)$ for random $u \leftarrow \mathbb{Z}_p$, then any label it obtains from the oracle is of the form $\sigma(\alpha \cdot u + \beta)$, where $\alpha, \beta$ can be determined from prior oracle queries. In other words, the attacker must "know" $\alpha$ and $\beta$.

The attacker is trying to find $z$ along with $\sigma(r)$ such that $z = r + u \cdot h(\sigma(r))$. But the attacker knows $\alpha$ and $\beta$ such that $r = \alpha \cdot u + \beta$, so this equation can be written as $z = \alpha \cdot u + \beta + u \cdot h(\sigma(\alpha \cdot u + \beta))$. If $\alpha + h(\sigma(\alpha \cdot u + \beta)) \neq 0$, then the attacker can solve for $u$. However, this means the attacker has found a discrete log, which it can only do with negligible probability [53].

Therefore, it must be the case that $\alpha + h(\sigma(\alpha \cdot u + \beta)) = 0$. However, the poly-query attacker only learns $\sigma(\alpha \cdot u + \beta)$ for poly-many choices of $(\alpha, \beta)$, and for each distinct choice of $(\alpha, \beta)$, the resulting label $\sigma(\alpha \cdot u + \beta)$ is random. $h$ evaluated on a random input has min-entropy $\omega(\log(\lambda))$, so the probability $\alpha + h(\sigma(\alpha \cdot u + \beta)) = 0$ holds is negligible; a union bound over the polynomially-many $(\alpha, \beta)$ oracle queries completes the argument.

*Schnorr Signatures in the Generic Group Model.* We now consider a slightly more difficult task: compiling Schnorr's identification protocol into a digital signature scheme with existential unforgeability against chosen-message attacks (EUF-CMA security).

Note that the semantics of the hash function itself are now different: the standard Fiat-Shamir compiler for signatures takes as input a message $m \in \mathcal{M}$ to be signed (in addition to the first message of the interactive protocol), i.e. $h : G \times \mathcal{M} \to \mathbb{Z}_p$. For the purposes of this technical overview, we will restrict to the case where $\mathcal{M}$ is a poly($\lambda$)-size set.[5] We stress that a restriction to only signing "short" messages will be crucial to the following discussion.

Furthermore, the EUF-CMA security experiment requires security in the presence of an unbounded number of signing queries. So the EUF-CMA attacker must solve following task:

- **Input:** A group description $G = (g, p)$, a hash function $h : G \times \mathcal{M} \to \mathbb{Z}_p$, and a random group element $g^u$.

[5] This restriction can in fact be relaxed somewhat, but our positive statements for information-theoretic Fiat-Shamir hash functions in the generic group model will crucially rely on $|\mathcal{M}|/p$ being negligible in $\lambda$.

– **Oracle Queries:** The attacker is free to make an unbounded number of queries to a signing oracle who knows $u$. It submits any $m \in \mathcal{M}$, the signing oracle samples a random $r \leftarrow \mathbb{Z}_p$, computes $z = r + h(g^r, m) \cdot u$, and returns the signature $(g^r, z)$.
– **Output:** Any $(m^*, (g^{r^*}, z^*))$ where $m^* \in \mathcal{M}$ satisfying $g^{z^*} = (g^{r^*})(g^u)^{h(g^{r^*}, m^*) \cdot u}$ that was not the result of a signing query.

We would like to identify a class of hash functions $h$ for which this problem is hard, and as in the previous section, we will start by identifying choices of $h$ that make this problem *easy*.

Suppose that $h$ has the following *undesirable* property: for some choice of $m \in \mathcal{M}$, the random variable obtained by sampling random $g^r \leftarrow G$ and outputting $h(g^r, m)$ has min-entropy $O(\log \lambda)$. In this case, breaking EUF-CMA security can be done efficiently without any signing queries. Let $c \in \mathbb{Z}_p$ be such that $h(g^r, m) = c$ holds with noticeable probability (guaranteed to exist by the low min-entropy property). The attack is to a uniformly random value $z \leftarrow \mathbb{Z}_p$, and then compute $g^r = g^{-uc+z}$. Since $g^r$ is randomly distributed, then $h(g^r, m) = c$ with noticeable probability, and the resulting $(g^r, z)$ constitutes a valid signature on $m$. To prevent this attack, we must require that for all $m \in \mathcal{M}$, the random variable $h(g^r, m)_{g^r \leftarrow G}$ has min-entropy $\omega(\log \lambda)$.

Another *undesirable* property of $h$ is the following: suppose for some choice of distinct $m, m' \in \mathcal{M}$, the random variable $(\chi_{h(g^r,m)=h(g^r,m')})_{g^r \leftarrow G}$ (where $\chi_{x=y}$ is the indicator function that equals 1 if $x = y$ and 0 otherwise) has noticeable expected value, i.e. $h(g^r, m) = h(g^r, m')$ occurs with noticeable probability. If $h$ satisfies this property, there is a straightforward attack using one signing query: the attacker queries on $m$, learns a random valid signature $(g^r, z)$, and then submits $(m', (g^r, z))$ as its forgery. Since the signing oracle provides a randomly generated valid signature (i.e. $g^r$ is random in $G$), the Fiat-Shamir challenge for the $m$ and $m'$ executions will be identical with noticeable probability, meaning the signature $(g^r, z)$ for $m$ is a valid signature for $m'$ with noticeable probability. To prevent this attack, we must require that for all distinct $m, m' \in \mathcal{M}$, the random variable $(\chi_{h(g^r,m)=h(g^r,m')})_{g^r \leftarrow G}$ has negligible expectation.

To recap, we have the following *minimum* requirements on $h$:[6].

1. For all $m \in \mathcal{M}$, we the min entropy of $h(g^r, m)_{g^r \leftarrow G}$ is $\omega(\log \lambda)$.
2. For all distinct $m, m' \in \mathcal{M}$, we have $E_{g^r \leftarrow G}[(\chi_{h(g^r,m)=h(g^r,m')})] \leq \text{negl}(\lambda)$.

It turns out that these minimum requirements on $h$ are sufficient to guarantee EUF-CMA security of Schnorr in the GGM:

[6] This is the characterization for the case $|\mathcal{M}| = \text{poly}(\lambda)$. For larger message spaces (that still satisfy $|\mathcal{M}|/p \leq \text{negl}(\lambda)$), the requirements are mildly strengthened: we require that (1) for all targets $c \in \mathbb{Z}_p$, the probability over a random choice of $r$ that $h(g^r, m) = c$ *for any* $m$ is negligible, and that for any $m \in \mathcal{M}$, the probability over a random choice of $r$ that $h(g^r, m') = h(g^r, m)$ *for any* $m'$ is negligible (i.e., we reversed an order of quantifiers in each requirement). These are exactly information-theroetic analogues of the RPP and RPSP properties defined in [46].

**Theorem 9.** *Suppose $\mathcal{M} \subset \mathbb{Z}_p$ and $\mathcal{M} = \text{poly}(\lambda)$. Let $h : G \times \mathcal{M} \to \mathbb{Z}_p$ be any function satisfying conditions (1) and (2) above. Then the resulting Schnorr signature scheme is EUF-CMA secure in the generic group model.*

We first note that our proof of Theorem 8 implies that an attacker cannot generate a valid forgery before it has received any signing queries. That is, given $\sigma(u)$, the attacker cannot output $(m^*, (\sigma(r^*), z^*))$ where $m^* \in \mathcal{M}$ and $z^* = r^* + h(\sigma(r^*), m^*) \cdot u$. To see this, note that for any fixed $m$, the hash function $h(\cdot, m)$ satisfies the same min-entropy property required for non-interactive identification (by condition (1) on $h$). A union bound over $\mathcal{M}$ implies the attacker cannot provide a forgery for any $m$.

Given this analysis, we prove Theorem 9 in two steps.

- **Step 1: Generate signing queries without knowledge of $u$.** In this step, we write down a hybrid experiment in which the adversary's view has *no explicit dependence* on the discrete logarithm $u$. We accomplish this by instead *programming* the group oracle.
  In more detail, when signing queries are answered honestly, the adversary receives $(\sigma(r), r + u \cdot h(\sigma(r), m))$. However, these signing queries can be *simulated* in the following way:
  - Sample a random label $\ell \leftarrow [L]$
  - Sample a random exponent $z \leftarrow \mathbb{Z}_p$.
  - Program the value $\sigma(z_i - x \cdot h(\ell, m)) = \ell$. If the oracle $\sigma$ was already programmed at $\ell$, abort.
  - Output the signature $(\ell, z, m)$

  Moreover, this gives us an *implicit representation* of the group element corresponding to label $\ell$ as a *publicly known* linear combination of $g^u$ and $g$, namely, $(g^z \cdot (g^u)^{-h(\ell,m)})$. These group elements will all be distinct with high probability over the choice of $u$.
  Essentially, this simulated experiment is indistinguishable from the real security game as long as the programmed values $\sigma(z_i - u \cdot h(\ell, m)))$ do not contradict any of the adversary's previous queries to the group oracle. One can show that the probability of this is negligible because of the randomness of $u$ according to the adversary's view. This is effectively an invocation of the generic group hardness of computing discrete logs.
- **Step 2: Invoke the statistical properties of $h$.** Now that we have simulated all of the signature queries, we consider a potential forgery $(\sigma(r^*), z^* = r^* + u \cdot h(\sigma(r^*), m^*), m^*)$ and break into two cases.
  - **Case 1: $\ell^* := \sigma(r^*)$ matches one of the signing queries**. In this case, we claim that a forgery allows us to compute the discrete logarithm $u$. Indeed, this is because we have a signing query equation of the form

    $$z = r^* + h(\ell^*, m)u$$

    and a forgery equation of the form

    $$z^* = r^* + h(\ell^*, m^*)u.$$

Moreover, the two hash values $(h(\ell^*, m), h(\ell^*, m^*))$ must be distinct because (1) the marginal distribution on $\ell^*$ is random, and (2) we assumed that for a random $\ell^*$, there will not exist an $h$-collision with prefix $\ell^*$.

- **Case 2: $\ell^*$ does not match any signing query**. In this case, we also claim that a forgery allows us to compute the discrete logarithm $u$. Indeed, the forgery equation

$$z^* = r^* + h(\ell^*, m^*)u$$

along with the adversary's implicit representation of the exponent

$$r^* = \alpha + \beta u$$

(which follows from the fact that the advesary's view can be computed generically given only $g^u$) implies that

$$z^* = \alpha + (\beta + h(\ell^*, m^*))u.$$

Then, either $\beta + h(\ell^*, m^*) \neq 0$, in which case the adversary can indeed compute $u$, or $\beta + h(\ell^*, m^*) = 0$. We claim that the high min-entropy of $h(\ell, m)$ for *random* $\ell$ implies that this event is unlikely. Indeed, $\ell^*$ must have been obtained by *some* group oracle query, so this follows by a union bound over all group oracle queries made by the adversary.

This completes our proof sketch of Theorem 9.

*Preprocessing Attacks.* We next show how the [46] characterization of Schnorr signature security in the GGM fails to capture security in concrete groups. Since the attacks that we discover fall into the framework of the auxiliary-input GGM [55, 18], we then analyze Schnorr signatures in this stronger adversary model.

We first describe an attack in the case of Schnorr signatures for short messages, using the hash function $h(g^r, m) = g^r + m \pmod p$ over the group[7] $G = \mathbb{Z}_p^\times$. We showed above that this signature scheme is secure in the generic group model, but we will nonetheless give an attack over $\mathbb{Z}_p^\times$.

In order to have a well-specified protocol, we need to fix a mapping $\mathrm{Int} : G \to \mathbb{Z}$ from group elements to integers. For simplicity, we choose our mapping so that $R \in \mathbb{Z}_p^\times$ maps to the unique integer $a \in [-\frac{p-1}{2}, \frac{p-1}{2}]$ such that $R \equiv a \pmod p$.

The attack proceeds as follows: we are given a random group element $g^u$ and want to output $m, g^r, z$ satisfying $g^z = (g^r)(g^u)^{\mathrm{Int}(g^r)+m}$. We do this by picking $r, m$ such that $\mathrm{Int}(g^r) + m = 0 \pmod{p-1}$ and then setting $r = z$. So, for example, if the message space $\mathcal{M}$ contains $m = p-2$, then we can pick $r = 0$, so that $g^r \equiv 1 \pmod p$ and $1 + p - 2 \equiv 0 \pmod{p-1}$. This choice is by no means special; if $1 \in \mathcal{M}$, then we can pick $r = \frac{p-1}{2}$ and obtain another forgery.

This strategy readily generalizes to groups beyond $\mathbb{Z}_p^\times$: for a cyclic group $G$ of order $p$, all that is required to produce a forgery is knowledge of an exponent

[7] This group does not have prime order, but this detail is not relevant to our analysis.

$r \in \mathbb{Z}_p$ and a message $\mu \in \mathcal{M} \subset \mathbb{Z}_p$ such that $\mathrm{Int}(g^r) = -\mu \pmod p$. It also generalizes to the case of full Schnorr signatures over $G$, using hash functions of the form $h(g^r, m) = \mathrm{Int}(g^r) + H(m)$ for a collision-resistant hash function $H$. One can check that the hash function (family) $h$ satisfies the hypotheses of [46], so Schnorr signatures using $h$ are secure in the GGM. However, if $G$ has a known equation of the form

$$\mathrm{Int}(g^r) = -\mu,$$

and $H$ additionally satisfies $H(0) = \mu$ (which can be arranged without sacrificing collision resistance by hard-coding this value into a hash function $H$ whose range excludes $\mu$), then again $(r, r)$ is a valid signature. Thus, we see that for every group $G$ with some hard-coded equation $\mathrm{Int}(g^r) = -\mu$, there exists a hash family $h$ satisfying the [46] hypotheses which leads to an *insecure* instantiation of Schnorr signatures.

We now observe that one can view this attack as an attack in the *auxiliary-input generic group model*. The Aux-Input GGM is the following adversary model for some problem $\mathcal{P}$ over a group $G$.

- The adversary is given the description of a group $G$ as a random injection from $G \to [L]$ (i.e., the adversary is given the full truth tables of the group operation).
- The adversary then stores $S$ bits of information about this group $G$ (and forgets everything else).
- The adversary then receives an instance of $\mathcal{P}$ (as characterized by a security game with a challenger). As in the GGM, the adversary can also query the group oracle.

In other words, an aux-input GGM adversary is a GGM adversary that is augmented with some $S$ bits of non-uniform advice about the group.

Given this definition, it is easy to see that the attacks described above fall into the aux-input GGM. Indeed, as long as the adversary "remembers" one equation of the form $\mathrm{Int}(g^r) = -\mu$ (of which many are guaranteed to exist), it will be able to execute an attack. Thus, one can view the attacks on $\mathbb{Z}_p^\times$ and other groups as the result of the following three-step process:

- There exist attacks on the schemes above in the auxiliary-input GGM. This means that for every concrete group $G$, there exists a *non-uniform* attack on the scheme.
- In the case of specific groups such as $\mathbb{Z}_p^\times$, the non-uniform advice necessary to carry out the attack can be computed efficiently given the group description.

*Security in the Aux-Input GGM.* Given the existence of preprocessing attacks as above, in order to have confidence in the *concrete* security of a Schnorr signature scheme using hash family $h$, it is necessary to prove security in the auxiliary-input GGM.

Just as in the case of our GGM lower bounds, we give a characterization of hash functions (and hash function families) $h$ that lead to secure Schnorr

signatures in the auxiliary-input GGM. We state a special case of our theorem for the purposes of this overview; we refer to the full version for a more general statement.

**Theorem 10.** *Let $\mathcal{M} \subset \mathbb{Z}_p$ and $|\mathcal{M}|/\mathbb{Z}_p \leq \mathrm{negl}(\lambda)$. Suppose the (keyed) Fiat-Shamir hash function $H_k : [L] \times \mathcal{M} \to \mathbb{Z}_p$ satisfies the following properties:*

- *For any $m \in \mathcal{M}$, $h(g^u, m)$ has min-entropy $\log(|\mathcal{M}|) \cdot \log \lambda$ on a random $g^u \leftarrow G$.*
- ***Zero-avoidance**: For any (stateful, potentially unbounded) adversary $\mathcal{A}$:*

$$\Pr\left[H_k(\ell, m) = 0 \,|\, \ell \leftarrow \mathcal{A}(1^\lambda), k \leftarrow \mathcal{K}, m \leftarrow \mathcal{A}(k)\right] \leq \mathrm{negl}(\lambda);$$

*Then Schnorr signatures with Fiat-Shamir hash function $H_k$ are EUF-CMA secure in the AI-GGM against adversaries $(\mathcal{A}_1, \mathcal{A}_2)$ with advice of size $S = \mathrm{poly}(\lambda)$, $T = \mathrm{poly}(\lambda)$ oracle queries, $Q = \mathrm{poly}(\lambda)$ signing queries.*

The first of the two hypotheses is the same as in Theorem 9; the second rules out the preprocessing attacks described above. Similarly to before, Theorem 10 says that once these attacks are avoided, no further attacks in the Aux-Input GGM exist.

We prove Theorem 10 using the framework of [18], who show a rough equivalence between the auxiliary-input GGM and an a priori weaker adversary model called the *bit-fixing GGM (BF-GGM)*. Informally, in the BF-GGM, instead of learning an arbitrary $S$ bits of information about a random group $G$, the adversary can only remember the *labels* of $P$ group elements (and their corresponding exponents with respect to the canonical generator). In [18], it is shown that for any (efficient and generic) challenger-adversary game, security in the AI-GGM follows from security in the (ostensibly weaker) BF-GGM with a slight loss in parameters. We can apply this result directly to the soundness of Schnorr signatures, reducing our problem to proving a lower bound in the BF-GGM.

Now, we can conveniently extend all of our GGM analysis (i.e., the proof of Theorem 9 to apply in the BF-GGM (and therefore to the AI-GGM via [18]). The BF-GGM lower bound will look very similar to before:

- **Step 1: Generate signing queries without knowledge of $u$.** We simulate signing queries in exactly the same way as before. Some care is required to argue that indistinguishability still holds, because the adversary additionally has access to a short list of hard-coded group labels.
- **Step 2: Invoke the statistical properties of $h$.** We again consider a potential forgery $(\sigma(r^*), z^* = r^* + h(\sigma(r^*), m^*)u, m^*)$. This time, we break into *three* cases:
  - **Case 0: $\ell^*$ appears in the adversary's auxiliary information.** This case is unique to the BF-GGM setting; however, the forgery equation

$$z^* = r^* + h(\ell^*, m)u$$

    allows us to solve for $u$ unless $h(\ell^*, m) = 0$, which cannot happen (except with negligible probability) because we assumed that $h$ was 0-avoiding.

- **Case 1: $\ell^* := \sigma(r^*)$ matches one of the signing queries**. This case matches our GGM analysis above.
- **Case 2: $\ell^*$ does not match any signing query**. This case also matches our GGM analysis above.

This completes our proof sketch of Theorem 10.

*Application: (Candidate) Simple Schnorr Signatures.* One takeaway of our analysis is that it *might* be possible that simple compilations of Schnorr signatures (for small message space) are secure. The appeal of such a signature scheme is that all of the operations are extremely simple, and can be implemented with random sampling and modular arithmetic. We stress that the only evidence we have for security is that this scheme resists *generic preprocessing attacks*, and that so far, we have been unable to leverage non-generic properties of $\mathbb{Z}_p^\times$ to break this scheme. Further analysis of this simple scheme is beyond the immediate scope of this work, and we *strongly recommend* against considering this scheme "secure" unless it withstands significant cryptanalytic effort.

**Construction 11.** *Consider the Schnorr signature scheme for group $\mathbb{Z}_p^\times$, where the Fiat-Shamir hash function has random $k \leftarrow \mathbb{Z}_q$, and outputs $g^r + m + k (\bmod q)$ on input $(g^r, m)$:*

- *Group: $\mathbb{Z}_p^\times$ with a generator $g$ of a cyclic subgroup of order $q$, where $p = 2q + 1$.*
- *Message space: Any subset $M \subset \mathbb{Z}_q$ of $\mathrm{poly}(\lambda)$ size.*
- *Signing key: $sk \leftarrow \mathbb{Z}_q$.*
- *Verification key: $(k, g^{sk})$ where $k \leftarrow \mathbb{Z}_q$.*
- $\mathsf{Sign}(\mathsf{sk}, m)$*: Sample $r \leftarrow \mathbb{Z}_q$. Let $z = r + (g^r + m + k) \cdot sk (\bmod q)$. Output $(g^r, z)$.*
- $\mathsf{Ver}(\mathsf{vk}, m, (g^r, z))$*: Accept if $g^z = g^r \cdot (g^{sk})^{g^r + m + k} (\bmod p)$.*

*Extensions to Chaum-Pedersen and NIZKs for* $\mathsf{NP}$. Our analysis for Schnorr signatures in the AI-GGM easily extends to prove *semi-adaptive* soundness of the Chaum-Pedersen protocol for proving validity of a Diffie-Hellman tuple. As the security analysis is extremely similar to our analysis for Schnorr, we defer this result (and its implications for NIZKs for $\mathsf{NP}$) to the full version.

### 2.3 Negative Results

In this section, we give a simple example of a negative result that we can prove using our methods. In particular, we consider an idealized variant of Blum's Hamiltonicity protocol [8] in which the commitment scheme is instantiated with a random oracle.

The Blum protocol $\Pi = \Pi^{\mathrm{Blum}}$ is described in Fig. 1. For this example, we instantiate $\mathsf{Com}(b; r) = \mathcal{O}(b, r)$ as an idealized bitwise commitment scheme in the random oracle model. $\Pi$ then is repeated $t$ times in parallel to obtain soundness error $2^{-t}$.

| | | |
|---|---|---|
| $P(G, \sigma)$ | | $V(G)$ |
| $\pi \leftarrow S_n$, $G' = \pi(G)$<br>$\alpha \leftarrow \mathsf{Com}(G')$ | $\xrightarrow{\alpha}$ | |
| | $\xleftarrow{\beta}$ | $\beta \leftarrow \{0, 1\}$ |
| If $\beta = 0$:<br>decommit to $G'$ and reveal $\pi$.<br>If $\beta = 1$:<br>reveal $\pi \circ \sigma$ and decommit to the edges in $G'$ corresponding to the cycle $\pi \circ \sigma$. | $\xrightarrow{\gamma}$ | Accept if<br>all decommitments are valid and:<br>either $\beta = 0$ and $G' = \pi(G)$<br>or $\beta = 1$ and all decommitments are 1. |

**Fig. 1.** The Zero Knowledge Proof System $\Pi^{\text{Blum}}$ for Graph Hamiltonicity.

At first glance, especially given our positive results for Schnorr and Chaum-Pedersen, one might hypothesize that since we have made the commitment scheme "super-secure", Fiat-Shamir for $\Pi^t$ might be instantiable with a simple hash function $h$. In fact, we show that even for this idealized variant of the Blum protocol, a (successful) Fiat-Shamir hash function $h$ for this protocol necessarily satisfies a cryptographic security property.

As discussed earlier, there are two variants of this result. First, we give a polynomial-query attack on $\Pi^t_{\text{FS},h}$ for any hash function $h$ that does not invoke the random oracle $\mathcal{O}$. Then, we extend this polynomial-query attack to a polynomial-time attack assuming the *easiness* of some computational problem depending on $h$.

To understand our attack, we first consider an "obviously broken" choice of hash function $h$: define $h(\alpha_1, \ldots, \alpha_t) = (f(\alpha_1), \ldots, f(\alpha_t))$ to be a fixed function applied to each commitment separately. This corresponds to a parallel repetition of $\Pi_{\text{FS},f}$, which is the application of Fiat-Shamir to a protocol with constant soundness error. We know that such a non-interactive protocol is unsound via a *reset attack*: given an instance $G$, it is possible to prepare a commitment $\alpha_1$ that can successfully answer either a "0" challenge or a "1" challenge. Therefore, if $\alpha_1$ is prepared to answer the challenge $b$ (for a uniformly random bit $b$), we have that $f(\alpha_1) = b$ with probability 1/2 (since $\alpha_1$ hides $b$) and so after an expected constant number of string commitment queries, we obtain an accepting transcript $(\alpha_1, b_1, \gamma_1)$ for the first repetition. This can be done for each "slot", giving a polynomial-query break of soundness for the overall protocol.

To rephrase the attack, for our example choice of $h$, if one prepares enough "fake commitments" $\{\alpha_1^{(i)}\}, \{\alpha_2^{(i)}\}, \ldots, \{\alpha_t^{(i)}\}$ for each of the $t$ repetitions, then with high probability, there exists a *combination* of the individual commitments that hashes to the "bad challenge" whose answer was generated along with the commitments. We show that the above argument generalizes to *all* hash functions $h$. The poly-query attack is as follows.

1. For $1 \leq i \leq t, 1 \leq \ell \leq q$, sample a random bit $y_\ell^{(i)} \leftarrow \{0,1\}$ and sample message $\alpha_\ell^{(i)}$: if $y_\ell^{(i)} = 0$, sample $\alpha_\ell^{(i)}$ as in the honest protocol, while if $y_\ell^{(i)} = 1$, and sample $\alpha_i^{(\ell)}$ as a commitment to a cycle graph.
2. Find $v \in [q]^t$ such that $h(\alpha[v]) = y[v]$. Abort if no such $v$ exists.
3. Output $\alpha[v]$ as well as the necessary decommitments to $\alpha[v]$ (either the entire graph or just the edges in the cycle).

This constitutes a poly-query attack on the protocol $\Pi^t_{\mathrm{FS},H}$ in the random oracle model as long as Step (2) has a solution with high probability over $(\alpha, y)$. In the case $h = (f, \ldots, f)$ as above, this condition follows immediately. We show in the full version that for *any* $h$, as long as $q = \omega(t)$, Step (2) has a solution with high probability over $(\alpha, y)$.

To obtain a (conditional) polynomial-*time* attack on the protocol, we note that if the solution to the problem in Step (2) can be found *efficiently*, then the above attack can be implemented in polynomial time.

Crucially, the above analysis generalizes well because the computational problem in Step (2) does not depend on the protocol. We accomplish this by reducing breaking the soundness of $\Pi^t_{\mathrm{FS},h}$ to solving a "mix-and-match" problem of the following form: given many strings $\{\alpha_\ell^{(i)}\}$ ($q$ strings for each slot) which are each associated with a random bit $b_\ell^{(i)}$, find a concatenation $\alpha[v]$ of $t$ different $\alpha_\ell^{(i)}$ (one for each slot) such that $h(\alpha[v]) = b[v]$ (the corresponding combination of bits). This motivates our definition of "mix-and-match resistance" (see the full version), a security property which captures the analogous problems for a wide class of protocols $\Pi$.

While the analysis above is tailored to (parallel repeated) $\Pi^{\mathrm{Blum}}$, it turns out that the argument only relies on a couple of (basic) properties of the protocol, namely:

- Given a challenge $\beta$, it is possible to sample a (pseudorandom) first message $\alpha$ along with an accepting response $\gamma$ for $\alpha$, even when the statement $x$ is false. This property is used to construct a mix-and-match problem in our attack, and essentially follows from an *honest-verifier zero knowledge* property of the protocol.
- The protocol is obtained by applying parallel repetition to a protocol with *polynomial-size* challenge space. This independence property is enough to guarantee that the "mix-and-match" problem information-theoretically has a solution.

We refer the reader to the full version for more details on the extent to which the result generalizes.

## Acknowledgments

We thank Brynmor Chapman, Justin Holmgren, Akshayaram Srinivasan, and Daniel Wichs for many helpful discussions. Part of this work was done while

the authors were visiting the Simons Institute for the Theory of Computing in Spring 2020.

## References


1. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO'93. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (Aug 1994)
2. Ben-Sasson, E., Bentov, I., Chiesa, A., Gabizon, A., Genkin, D., Hamilis, M., Pergament, E., Riabzev, M., Silberstein, M., Tromer, E., Virza, M.: Computational integrity with a public random string from quasi-linear PCPs. In: Coron, J.S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part III. LNCS, vol. 10212, pp. 551–579. Springer, Heidelberg (Apr / May 2017)
3. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Fast reed-solomon interactive oracle proofs of proximity. In: Chatzigiannakis, I., Kaklamanis, C., Marx, D., Sannella, D. (eds.) ICALP 2018. LIPIcs, vol. 107, pp. 14:1–14:17. Schloss Dagstuhl (Jul 2018)
4. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Report 2018/046 (2018), https://eprint.iacr.org/2018/046
5. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Scalable zero knowledge with no trusted setup. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 701–732. Springer, Heidelberg (Aug 2019)
6. Ben-Sasson, E., Chiesa, A., Riabzev, M., Spooner, N., Virza, M., Ward, N.P.: Aurora: Transparent succinct arguments for R1CS. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 103–128. Springer, Heidelberg (May 2019)
7. Ben-Sasson, E., Chiesa, A., Spooner, N.: Interactive oracle proofs. In: Hirt, M., Smith, A.D. (eds.) TCC 2016-B, Part II. LNCS, vol. 9986, pp. 31–60. Springer, Heidelberg (Oct / Nov 2016)
8. Blum, M.: How to prove a theorem so no one else can claim it. In: Proceedings of the International Congress of Mathematicians. vol. 1, p. 2. Citeseer (1986)
9. Brakerski, Z., Koppula, V., Mour, T.: NIZK from LPN and trapdoor hash via correlation intractability for approximable relations. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 738–767. Springer, Heidelberg (Aug 2020)
10. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Ostrovsky, R. (ed.) 52nd FOCS. pp. 97–106. IEEE Computer Society Press (Oct 2011)
11. Canetti, R., Chen, Y., Holmgren, J., Lombardi, A., Rothblum, G.N., Rothblum, R.D., Wichs, D.: Fiat-Shamir: from practice to theory. In: Charikar, M., Cohen, E. (eds.) 51st ACM STOC. pp. 1082–1090. ACM Press (Jun 2019)
12. Canetti, R., Chen, Y., Reyzin, L., Rothblum, R.D.: Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part I. LNCS, vol. 10820, pp. 91–122. Springer, Heidelberg (Apr / May 2018)
13. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited (preliminary version). In: 30th ACM STOC. pp. 209–218. ACM Press (May 1998)

14. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 523–552. Springer, Heidelberg (May / Jun 2010)
15. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell, E.F. (ed.) CRYPTO'92. LNCS, vol. 740, pp. 89–105. Springer, Heidelberg (Aug 1993)
16. Chen, Y., Lombardi, A., Ma, F., Quach, W.: Does fiat-shamir require a cryptographic hash function? Cryptology ePrint Archive, Report 2020/915 (2020), https://eprint.iacr.org/2020/915
17. Ciampi, M., Parisella, R., Venturi, D.: On adaptive security of delayed-input sigma protocols and fiat-shamir NIZKs. In: Galdi, C., Kolesnikov, V. (eds.) SCN 20. LNCS, vol. 12238, pp. 670–690. Springer, Heidelberg (Sep 2020)
18. Coretti, S., Dodis, Y., Guo, S.: Non-uniform bounds in the random-permutation, ideal-cipher, and generic-group models. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 693–721. Springer, Heidelberg (Aug 2018)
19. Corrigan-Gibbs, H., Kogan, D.: The discrete-logarithm problem with preprocessing. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 415–447. Springer, Heidelberg (Apr / May 2018)
20. Couteau, G., Hofheinz, D.: Designated-verifier pseudorandom generators, and their applications. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part II. LNCS, vol. 11477, pp. 562–592. Springer, Heidelberg (May 2019)
21. Couteau, G., Katsumata, S., Ursu, B.: Non-interactive zero-knowledge in pairing-free groups from weaker assumptions. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 442–471. Springer, Heidelberg (May 2020)
22. Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO'98. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (Aug 1998)
23. Damgard, I.: On sigma-protocols, lecture notes, faculty of science aarhus university, department of computer science (2010)
24. Dent, A.W.: Adapting the weaknesses of the random oracle model to the generic group model. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 100–109. Springer, Heidelberg (Dec 2002)
25. Döttling, N., Garg, S., Ishai, Y., Malavolta, G., Mour, T., Ostrovsky, R.: Trapdoor hash functions and their applications. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 3–32. Springer, Heidelberg (Aug 2019)
26. Feige, U., Lapidot, D., Shamir, A.: Multiple noninteractive zero knowledge proofs under general assumptions. SIAM J. Comput. **29**(1), 1–28 (Sep 1999)
27. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (Aug 1987)
28. Fischlin, M.: A note on security proofs in the generic model. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 458–469. Springer, Heidelberg (Dec 2000)
29. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) 41st ACM STOC. pp. 169–178. ACM Press (May / Jun 2009)
30. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press (May 2008)

31. Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions (extended abstract). In: 25th FOCS. pp. 464–479. IEEE Computer Society Press (Oct 1984)
32. Goldwasser, S., Micali, S.: Probabilistic encryption and how to play mental poker keeping secret all partial information. In: 14th ACM STOC. pp. 365–377. ACM Press (May 1982)
33. Holmgren, J., Lombardi, A.: Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In: Thorup, M. (ed.) 59th FOCS. pp. 850–858. IEEE Computer Society Press (Oct 2018)
34. Kalai, Y.T., Rothblum, G.N., Rothblum, R.D.: From obfuscation to the security of Fiat-Shamir for proofs. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 224–251. Springer, Heidelberg (Aug 2017)
35. Katsumata, S., Nishimaki, R., Yamada, S., Yamakawa, T.: Designated verifier/prover and preprocessing NIZKs from Diffie-Hellman assumptions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part II. LNCS, vol. 11477, pp. 622–651. Springer, Heidelberg (May 2019)
36. Kilian, J.: A note on efficient zero-knowledge proofs and arguments (extended abstract). In: 24th ACM STOC. pp. 723–732. ACM Press (May 1992)
37. Lombardi, A., Vaikuntanathan, V.: Fiat-shamir for repeated squaring with applications to PPAD-hardness and VDFs. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 632–651. Springer, Heidelberg (Aug 2020)
38. Lyubashevsky, V.: Lattice-based identification schemes secure under active attacks. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 162–179. Springer, Heidelberg (Mar 2008)
39. Lyubashevsky, V.: Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 598–616. Springer, Heidelberg (Dec 2009)
40. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 738–755. Springer, Heidelberg (Apr 2012)
41. Lyubashevsky, V., Wichs, D.: Simple lattice trapdoor sampling from a broad class of distributions. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 716–730. Springer, Heidelberg (Mar / Apr 2015)
42. Maurer, U.: Abstract models of computation in cryptography. In: IMA International Conference on Cryptography and Coding. pp. 1–12. Springer (2005)
43. Micali, S.: Computationally sound proofs. SIAM Journal on Computing **30**(4), 1253–1298 (2000)
44. Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (Apr 2012)
45. Nechaev, V.I.: Complexity of a determinate algorithm for the discrete logarithm. Mathematical Notes **55**(2), 165–172 (1994)
46. Neven, G., Smart, N.P., Warinschi, B.: Hash function requirements for schnorr signatures. Journal of Mathematical Cryptology **3**(1), 69–87 (2009)
47. Okamoto, T.: Provably secure and practical identification schemes and corresponding signature schemes. In: Brickell, E.F. (ed.) CRYPTO'92. LNCS, vol. 740, pp. 31–53. Springer, Heidelberg (Aug 1993)
48. Peikert, C.: A decade of lattice cryptography. Foundations and Trends in Theoretical Computer Science **10**(4), 283–424 (2016)

49. Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (plain) learning with errors. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part I. LNCS, vol. 11692, pp. 89–114. Springer, Heidelberg (Aug 2019)
50. Pointcheval, D., Stern, J.: Provably secure blind signature schemes. In: Kim, K., Matsumoto, T. (eds.) ASIACRYPT'96. LNCS, vol. 1163, pp. 252–265. Springer, Heidelberg (Nov 1996)
51. Quach, W., Rothblum, R.D., Wichs, D.: Reusable designated-verifier NIZKs for all NP from CDH. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part II. LNCS, vol. 11477, pp. 593–621. Springer, Heidelberg (May 2019)
52. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 239–252. Springer, Heidelberg (Aug 1990)
53. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT'97. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (May 1997)
54. Stern, J., Pointcheval, D., Malone-Lee, J., Smart, N.P.: Flaws in applying proof methodologies to signature schemes. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 93–110. Springer, Heidelberg (Aug 2002)
55. Unruh, D.: Random oracles and auxiliary input. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 205–223. Springer, Heidelberg (Aug 2007)
56. Wahby, R.S., Tzialla, I., shelat, a., Thaler, J., Walfish, M.: Doubly-efficient zkSNARKs without trusted setup. In: 2018 IEEE Symposium on Security and Privacy. pp. 926–943. IEEE Computer Society Press (May 2018)