Does Fiat-Shamir Require a Cryptographic Hash Function?

Yilei Chen[*] Alex Lombardi[†] Fermi Ma[‡] 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 about the interactive protocol (i.e., we invoke the generic group model), while in others, we argue soundness in the plain model. At a high level, the security of each resulting non-interactive protocol derives from hard problems already implicit in the original interactive protocol.

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

[*] Visa Research. Email: `chenyilei.ra@gmail.com`.

[†] 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`.

Contents

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 Π denote a 3-message (public-coin) argument system with transcripts of the form (α, β, γ). Then, given any *hash function* h, the Fiat-Shamir transform of Π 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 [FS87] 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 [FS87, Sch90, PS96], non-interactive zero-knowledge arguments (NIZKs) [BR94, CCRR18, CCH$^+$19, PS19], and succinct non-interactive arguments (SNARGs) [Kil92, Mic00, BCS16, BBC$^+$17, BBHR18b, BBHR18a, WTs$^+$18, BCR$^+$19, BBHR19].

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 [GM82]. 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** [BR94]: 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 Π" is not an application that can distinguish h from a random oracle.
- **Correlation Intractability**: In a recent line of work [KRR17, CCRR18, HL18, CCH$^+$19, PS19, BKM20, LV20], 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 [CGH98].
  - 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.

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 [CGH98] whose security can be based on standard cryptographic assumptions [CCH$^+$19, PS19, BKM20].

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.

[1] See discussion in [BR94]

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 [FS87], a Fiat-Shamir hash function could be instantiated using a pseudorandom function family [GGM84] (they give DES as an example instantiation).
- As proposed in the random oracle methodology [BR94], 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.
- For the special case of Schnorr signatures [Sch90], it was shown in [NSW09] that a form of random-prefix (second) preimage-resistance (which is implied by collision-resistance) plausibly suffices for security, which suggests using a collision-resistant hash function for Schnorr signatures.
- In the provably secure instantiations of [CCH$^+$19, PS19], the hash function families are based on flavors of *fully homomorphic encryption*, which can be instantiated from lattice assumptions [Gen09, BV11].
- In the recent work [BKM20], a (modified) *trapdoor hash function* [DGI$^+$19] 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 Π, the answer is "yes" for uninteresting reasons. For example, given any constant-round, public-coin interactive protocol Π, there is a protocol $\tilde{\Pi}$ that replaces all prover messages α_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 Π 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 [FS87, Sch90, Lyu09], and
2. Round-compressing 3-message honest-verifier zero knowledge argument systems to obtain NIZK arguments for NP [BR94, CCRR18, CCH$^+$19, PS19].

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

- Schnorr's identification scheme [Sch90].
- The Chaum-Pedersen interactive proof system for the Diffie-Hellman language [CP93].
- Lyubashevsky's lattice-based identification scheme [Lyu09].
- More generally, Σ-protocols [Dam10], 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. To give a taste of our results, here is a theorem that we formalize and prove:

Theorem 1.1 (Informal)**.** *If the Schnorr identification scheme [Sch90] or the Chaum-Pedersen [CP93] interactive proof system is instantiated with full-size challenges $\beta \leftarrow \mathbb{Z}_p$, then Fiat-Shamir for this protocol can be soundly instantiated with an extremely simple (even $\mathbb{Z}_p$-linear) hash function in the Generic Group Model (GGM).*

However, if either protocol is instantiated with $\{0,1\}$-challenges and repeated λ times in parallel, then Fiat-Shamir using a non-cryptographic hash function is unsound*, even in the GGM.*

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

1.1.1 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 1.2 (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 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** Π.

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

Definition 1.3 (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 Π does not require a cryptographic hash function (Definition 1.3), 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 generic group model [Nec94, Sho97]. 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 Π (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.

How We Give Negative Results. In order to obtain a negative result, we would like to show that for a particular protocol Π, 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 Π.

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 Π then $\mathcal{H}$ necessarily satisfies $\mathcal{P}$. Since $\mathcal{P}$ is independent of the protocol Π, this comes closer to distinguishing $\mathcal{H}$-hardness from hardness in Π.

However, there is still one issue with the above strategy: 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 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 [Blu86] instantiated in the random-oracle model.

As an added bonus, we are also sometimes able to give direct attacks on $\Pi_{\mathrm{FS},\mathcal{H}}$ relative to the oracle (i.e., in the generic group model or the random oracle model). That is, 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.

1.1.2 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.

Mini-Result: Schnorr Signatures with a Linear Fiat-Shamir Hash Function. Our first 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 observe that for signing *short* messages, 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 1.4 (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 small subset of $\mathbb{Z}_p$, i.e. $\mathcal{M} \subset \mathbb{Z}_p$ and $|\mathcal{M}|/\mathbb{Z}_p \leq \mathrm{negl}(\lambda)$. Let ℓ 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*

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

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

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

Soundness of the resulting Schnorr signature can be proved by re-purposing a security analysis due to [NSW09]; this work characterized a security property of $\mathcal{H}$ that suffices for (long-message) signatures schemes in the GGM. In our case, an *information-theoretic* property of h suffices; see Section 2 for details.

One takeaway from Theorem 1.4 is that Schnorr-like signatures can 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 in long messages.

The Chaum-Pedersen Protocol and NIZKs for NP. Next, we consider an interactive proof system due to Chaum and Pedersen [CP93] for proving membership in the Diffie-Hellman language $\mathcal{L}_{\mathrm{DH}} \coloneqq \{(g, g^u, g^v, g^{uv})\}_{g \in G, u,v \in \mathbb{Z}_p}$. 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 [CS98]). Notably, a recent line of work [CH19, KNYY19, QRW19, CKU20] 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 generic group model that a simple, fixed Fiat-Shamir hash function h suffices to compile the Chaum-Pedersen protocol into an argument for $\mathcal{L}_{\mathrm{DH}}$ satisfying a weaker 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 1.5. *Let Π^{CP} denote the Chaum-Pedersen protocol over a group G of order p. Let ℓ 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 generic group model, $(\Pi^{CP})_{FS,h}$ is a semi-adaptively sound argument system for $\mathcal{L}_{\mathrm{DH}}$.

In Section 4, we prove a stronger result: as long as h satisfies an (easily satisfied) information theoretic property, $(\Pi^{\mathrm{CP}})_{\mathrm{FS},h}$ is sound in the 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 [FLS99], and consequently NIZKs for NP in the standard model [CH19, KNYY19, QRW19, CKU20].

This gives an interesting alternative to two prior constructions of NIZKs for NP based on pairing-free groups [CCRR18, CKU20]: as in [CKU20], our NIZK implements the hidden bits model using a Fiat-Shamir NIZK for the DDH language, but we replace the correlation-intractable Fiat-Shamir hash function of [CKU20] built from exponentially KDM-secure ElGamal with the addition-mod-p function.

Lattice-Based Identification Schemes. We next turn to lattice-based analogues of the Schnorr protocol. In particular, we consider variants of Lyubashevsky's identification schemes [Lyu08, Lyu09, Lyu12], which were designed to obtain efficient signature schemes in the random oracle model via Fiat-Shamir.

We obtain a sound Fiat-Shamir instantiation for a protocol Π that is a small modification of the "basic Lyubashevsky protocol." 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 1.6 (Informal)**.** *For close variants of the [Lyu09] ID scheme based on either SIS or LWE, the non-interactive protocol resulting from the Fiat-Shamir transform using the $\mathbf{G}^{-1}$ "hash function" is average-case sound under SIS.*

As in the group setting, we obtain meaningful soundness guarantees using a deterministic hash function. We also prove Theorem 1.6 for a large class of Fiat-Shamir hash functions (that includes bit-decomposition). However, unique to the lattice setting, we manage to prove soundness in the *standard model*, relying only on the Short Integer Solution (SIS) assumption! More specifically, the SIS assumption suffices to argue *average-case* soundness, where soundness holds over randomly generated (false) instances.

Variants of our protocol Π_{FS} also show a surprising connection to Micciancio-Peikert lattice trapdoors [MP12, LW15]. Namely, the prover algorithm in Π_{FS} can be interpreted as a preimage sampling algorithm using a Micciancio-Peikert trapdoor. This highlights a potential connection between two seemingly orthogonal paths to build signatures from lattice-based assumptions: one using lattice trapdoors [GPV08, CHKP10, MP12] and the other through the Fiat-Shamir heuristic [Lyu08, Lyu09, Lyu12]. To the best of our knowledge, no such connection was known before. We discuss this connection in more detail in the technical overview.

Negative Results. To complement our positive results, we also give evidence that for some protocols, Fiat-Shamir necessarily requires a cryptographic hash function. Our negative results apply to a class of **three-message honest-verifier zero-knowledge (HVZK) arguments** (or proofs). Two prototypical examples to have in mind are:

- Blum's Hamiltonicity protocol [Blu86], 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 GGM, respectively). We give evidence that analogues to Theorem 1.4, Theorem 1.5, and Theorem 1.6 *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 1.7 (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 $\Pi^t_{\mathrm{FS},\mathcal{H}}$ is unsound in the GGM.*

  For any *instantiation of the [Blu86] 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.*

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

More generally (see Theorem 6.7), we give such an impossibility result for any 3-message HVZK protocol Π in the ROM/GGM satisfying some technical requirements. The most important requirement is that Π 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" (Definition 6.8)) such that for any protocol Π 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 Π must possess this security property. In other words, we show:

  Theorem 1.8 (Informal, see Theorem 6.9)**.** *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.

  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 1.7 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 Π, 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 work of [CCH+19, PS19] tells 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 Appendix A, we explore this connection further by showing that correlation-intractable hash functions (as constructed by [CCH+19, PS19]) suffice to instantiate Fiat-Shamir for (a variant of) the *idealized* Blum protocol.

1.2 Conclusions

The soundness of Fiat-Shamir has typically 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 related to 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, and using cryptographic hardness from the *interactive protocol.*

As mentioned before, this leads to noticeable qualitative differences from prior approaches, 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.

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 Σ-protocol appears to necessitate using a cryptographic hash function.

We view our contributions as a starting point for a more precise understanding of *when* hardness is required from a Fiat-Shamir hash function.

1.3 Related Work

To the best of the authors' knowledge, the only prior work to explicitly consider Fiat-Shamir for *non-cryptographic* hash functions is the work of Mittelbach and Venturi [MV16]. They identify a class of so-called "highly sound" protocols for which Fiat-Shamir can be soundly applied using any q-wise independent hash function.[2] Moreover, they showed that using indistiguishability obfuscation, any 3-round public coin interactive proof system can be converted into one that is "highly sound." However, the class of protocols for which their compiler works is extremely narrow; the only non-trivial protocols we are aware of satisfying their criteria are obtained through indistinguishability obfuscation.

Negative Results for Fiat-Shamir. A celebrated result of [DNRS99] shows that Fiat-Shamir *in the standard model* is not instantiable for a 3-message protocol Π that is *malicious-verifier* zero knowledge. This result can be seen as an extension of prior impossibility results [GO94, GK90] for constant-round public-coin zero knowledge.

The basic ideas present in these (and other) negative results — use a zero-knowledge simulator for the protocol to contradict the soundness of a related protocol — appear in an altered form in our negative results (Theorem 6.9, Theorem 6.7). However, in this work, we show that (in some settings) even honest-verifier zero knowledge (which is easily satisfied by many 3-message protocols) of the interactive protocol is sufficient to imply that a Fiat-Shamir hash function must be cryptographic.

Correlation Intractability and Fiat-Shamir. In a long sequence of works [KRR17, CCRR18, HL18, CCH+19, PS19, BKM20, LV20], it was shown that Fiat-Shamir in the standard model can be provably instantiated (for an interesting class of protocols) by using a Fiat-Shamir hash family $\mathcal{H}$ satisfying variants of *correlation intractability* [CGH98]. A hash family $\mathcal{H}$ is correlation intractable for a sparse relation $R(x, y)$ if given $h \leftarrow \mathcal{H}$, it is computationally hard to find an input x such that $(x, h(x)) \in R$.

There is a fairly strong established connection between correlation-intractability and Fiat-Shamir (see discussion in [CCRR18]); in fact, it is known that (under appropriate formulations) for a hash family $\mathcal{H}$, correlation intractability for *all* sparse relations is equivalent to soundly instantiating Fiat-Shamir for *all* constant-round public-coin (statistically sound) interactive proofs. This implies a weak negative result for Fiat-Shamir with information-theoretic hash functions: it says that if $\mathcal{H}$ instantiates Fiat-Shamir *simultaneously* for a large class of interactive protocols, then $\mathcal{H}$ is cryptographic.[3]

As a result, one could attempt to study the questions in this paper through the correlation intractability lens. However, our questions do not appear to translate well into the language of correlation intractability. This is mainly because we do not ask $\mathcal{H}$ to instantiate Fiat-Shamir for such a large class of protocols (such as all 3-round public coin interactive proofs) at once. For any fixed 3-message protocol Π, correlation intractability for the "transcript relation" $R_x = \{(\alpha, \beta) : \exists \gamma \text{ such that } V(x, \alpha, \beta, \gamma) = 1\}$ is too strong of a security property to exactly capture the soundness of Fiat-Shamir for Π. This is because correlation intractability does not capture the hardness of finding an accepting third message γ along with the first message α.

On a related note, the work of Dodis et al. [DRV12] shows that a property of hash function families called "entropy preservation" is necessary for the soundness for Fiat-Shamir for proofs (it is shown in [CCR16] that entropy preservation and correlation intractability are equivalent in some parameter settings). This is also a characterization of when a hash family $\mathcal{H}$ instantiates FS *simultaneously* for *all* (constant-round public coin) interactive proofs. The result of Dodis et al. does not show that entropy preservation is necessary for instantiating FS for any fixed protocol such as Blum's protocol for Hamiltonian cycles.

[2] In fact, q-wise independence was only used to obtain q-theorem zero-knowledge; soundness follows from 1-wise independence.

[3] It is not hard to see that correlation-intractable hash functions (for a fairly small class of sparse relations) imply the existence of one-way functions: in the case that h is shrinking by a factor of 2, consider the function family $f(x) = h(x) + p(x)$ for $h \leftarrow \mathcal{H}$ and p sampled from a pairwise independent hash family.

2 Technical Overview

We give an overview of our positive results for group-based protocols in Sections 2.1 and 2.2 and our positive results for lattice-based identification protocols in Section 2.3. We then highlight the intuition behind our negative results in Section 2.4.

2.1 Warm-up: Fiat-Shamir for the Schnorr Protocol

We begin with some positive results to build intuition for our approach. The results in this warm-up section are easy from a technical standpoint and we will not revisit them in the body of this paper.

Our positive results begin with 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. We will later see how to extend this intuition 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 $g^z = (g^r)(g^u)^{h(g^r)}$.

Our first goal is to identify simple, fixed choices for h such that this problem is plausibly hard.

At the very least, h should not be a constant function, i.e. $h(g^x) = c$ for all g^x, since 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.

Mini-Result: One-Time Soundness Against Generic Algorithms. This characterization of insecure choices for h turns out to be tight for generic algorithms.

Theorem 2.1. *In the generic group model (GGM), the non-interactive Schnorr protocol is one-time secure provided $h(\cdot)$ on a random input has entropy $\omega(\log \lambda)$.*

In the generic group model, group elements g^x are replaced by labels $\sigma(x)$ where σ is a random injection from $\mathbb{Z}_p$ to an exponentially-larger label space $[L]$. The attacker interacts with an oracle (who knows the truth table of σ) 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 of a group element.

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 α, β can be determined from prior oracle queries. In other words, the attacker must "know" α and β.

The attacker is trying to find z along with $\sigma(r)$ such that $z = r + u \cdot h(\sigma(r))$. But the attacker knows α and β 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 [Sho97].

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 (α, β), and for each distinct choice of (α, β), 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 $\alpha + h(\sigma(\alpha \cdot u + \beta)) = 0$; a union bound over the polynomially many (α, β) oracle queries completes the argument.

Mini-Result: Schnorr Signatures. In the Schnorr signature scheme, the Fiat-Shamir hash function additionally takes as input the message $m \in \mathcal{M}$ to be signed (in addition to the first message of the interactive protocol), i.e. $h : G \times \mathcal{M} \to \mathbb{Z}_p$.

The work of [NSW09] gives a necessary and sufficient security property for the Fiat-Shamir hash function to ensure that the compiled Schnorr signature scheme is existentially unforgeable against chosen message attacks (EUF-CMA) in the generic group model. In particular, the hash function must be "random prefix pre-image resistant". In this security game, the attacker commits to a hash output $y \in \mathbb{Z}_p$, the challenger samples a random $g^u \leftarrow G$, and the attacker wins if it can find $m \in \mathcal{M}$ such that $h(g^u, m) = y$.

We observe that if the message space $\mathcal{M}$ is a small subset of $\mathbb{Z}_p$, i.e. $\mathcal{M} \subset \mathbb{Z}_p$ and $|\mathcal{M}|/\mathbb{Z}_p \leq \text{negl}(\lambda)$, then a similar entropic property of $\mathbf{h}$ will information-theoretically satisfy random prefix pre-image resistance.

Theorem 2.2. *Suppose $\mathcal{M} \subset \mathbb{Z}_p$ and $|\mathcal{M}|/\mathbb{Z}_p \leq \text{negl}(\lambda)$. Let $h : G \times \mathcal{M} \to \mathbb{Z}_p$ be a function such that for any $m \in \mathcal{M}$, $h(g^u, m)$ has min-entropy $\log(|\mathcal{M}|) \cdot \log \lambda$ on a random $g^u \leftarrow G$. Then the resulting Schnorr signature sceme is EUF-CMA secure.*

Given the [NSW09] characterization, it suffices to prove that h statistically satisfies random prefix pre-image resistance. For any choice of y the attacker commits to in the "random prefix pre-image resistance" game, any particular choice of $m \in \mathcal{M}$ is a solution to $h(g^u, m) = y$ with probability $1/(|\mathcal{M}| \cdot 2^{\omega(\log \lambda)})$; a union bound over all $m \in \mathcal{M}$ completes the argument.

As was the case for the simple non-interactive Schnorr identification scheme in the previous section, *many simple choices of h will satisfy this definition.* For instance, a hash function h which interprets the bit-representation of g^u as an integer, adds m, and returns the result modulo p suffices.

2.2 A Non-Interactive Argument for the Diffie-Hellman Language.

Our main positive result for compiling group-based protocols concerns the Chaum-Pedersen interactive proof system for validity of a Diffie-Hellman tuples, i.e. membership in $\mathcal{L}_{\text{DH}} := \{(g, g^u, g^v, g^{uv})\}_{u,v \in \mathbb{Z}_p}$.

In the Chaum-Pedersen protocol, the prover and verifier have a shared instance (g, g^u, g^v, g^w), and the prover runs two simultaneous instances of the Schnorr protocol to convince the verifier it knows the discrete log of g^v with respect to g and the discrete log of g^w with respect to g^u. The crucial point is that the two Schnorr instances are not fully independent: there will only be a single verifier challenge $c \in \mathbb{Z}_p$ and only a single prover response $z \in \mathbb{Z}_p$. This ensures that not only does the prover know $\log_g(g^v)$ and $\log_{g^u}(g^w)$, but that these discrete logs are equal, which implies $g^w = g^{uv}$. Concretely, the protocol is:

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

A recent line of work [CH19, KNYY19, QRW19, CKU20] has shown that a non-interactive, adaptively sound, (single-theorem) zero-knowledge argument for $\mathcal{L}_{\text{DH}}$ (along with the Computational Diffie-Hellman assumption) suffices to instantiate non-interactive zero-knowledge (NIZK) arguments for all of NP. This makes the task of compiling Fiat-Shamir for this protocol, particularly with a *simple, concrete hash function*, especially compelling. For this overview, we will focus on obtaining soundness, as zero-knowledge will follow easily from standard techniques (see Section 4.3).

In an adaptively sound protocol, the cheating prover can win by convincing the verifier to accept an arbitrary NO-instance of $\mathcal{L}_{\mathrm{DH}}$. We will settle for a weaker security notion we call *semi-adaptive* soundness, but we observe that the compiler of [CKU20] actually only requires this weaker notion of soundness to construct NIZKs for NP. In semi-adaptive soundness, the cheating prover is given a random g^u, adaptively selects g^v, g^w such that $(g, g^u, g^v, g^w) \notin \mathcal{L}_{\mathrm{DH}}$, and wins if the verifier accepts.

To apply Fiat-Shamir to this protocol, the hash function should take as input the first message of the interactive protocol as well as any part of the instance that the prover controls. So for any particular $h : G^4 \to \mathbb{Z}_p$, the cheating prover's goal is to solve the following problem.

- **Input:** A group description $G = (g, p)$, a hash function $h : G^4 \to \mathbb{Z}_p$, and a random group element g^u.
- **Output:** g^v, g^w, g^r, g^s, z satisfying

$$\begin{aligned} g^w &\neq g^{uv}, \\ g^z &= (g^r)(g^u)^{h(g^v, g^w, g^r, g^s)}, \\ (g^v)^z &= (g^s)(g^w)^{h(g^v, g^w, g^r, g^s)}. \end{aligned}$$

We identify an information theoretic property on h — that it is "well-spread on repeated random inputs" — that makes this problem hard for generic attackers. For the sake of readability, we defer the definition and the discussion of this property until after we give intuition for the security proof.

Theorem 2.3. *In the generic group model, the non-interactive Chaum-Pedersen protocol is semi-adaptively sound provided $h : G^4 \to \mathbb{Z}_p$ is "well-spread on repeated random inputs".*

Our generic group proof extends the ideas laid out in the generic group security argument given for the one-time compilation of Schnorr's identification protocol.

At a high level, the argument will rely heavily on the fact that an attacker cannot solve for u. We use the fact that the group elements it outputs, $\sigma(v), \sigma(w), \sigma(r), \sigma(s)$ must each be of the form $\sigma(\alpha \cdot u + \beta)$ for some choice of α and β known to the generic group attacker. Since the attacker knows z, the last two checks will induce two equations that the attacker can satisfy (if it breaks soundness), both of which involve u.

Since the attacker cannot solve for u except with noticeable probability, we will argue that these equations must have a (net) coefficient of 0 on u. This will leads to several constraints involving the various α, β coefficients of v, w, r, s, along with z and the evaluation of the Fiat-Shamir hash function h.

If the evaluation of the Fiat-Shamir hash function h is taken to be a formal variable, the only way the attacker can satisfy the induced constraints is to give a YES-instance of $\mathcal{L}_{\mathrm{DH}}$, along with an honestly generated proof $\sigma(r), \sigma(s), z$. The goal is then to identify an information theoretic property of h that will imply the only way the attacker satisfies these constraints with non-negligible probability is to satisfy them when the evaluation of h is taken as a formal variable.

The property we ask of h is that it is "well-spread on repeated random inputs." While the attacker can pick v, w, r, s however it likes, it is not evaluating h directly on v, w, r, s, but instead on $\sigma(v), \sigma(w), \sigma(r), \sigma(s)$. Since σ is a uniformly random labeling function, the attacker can only evaluate h on inputs that are essentially random strings. There is one major caveat, however, which is that the attacker can choose to set any of v, w, r, s equal to each other. In this case, h will be run on random inputs that repeat in a pattern that the attacker can specify.

Therefore, a hash function h is well-spread on repeated random inputs if: for any choice of $i_1, i_2, i_3, i_4 \in \{1, 2, 3, 4\}^4$, if $x_1, \ldots, x_4$ are sampled as random strings, then $h(x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4})$ has $\omega(\log \lambda)$ min-entropy.

It turns out that many simple hash functions satisfy this property. For instance the function $h(g_1, g_2, g_3, g_4)$ that interprets the bit representation of each input as a positive integer and outputs the result mod p will satisfy this property. As another example, $h(g_1, g_2, g_3, g_4)$ can apply a pair-wise independent hash function outputting $\log(p)/4$ bits to each g_i independently, and return the concatenation of the results (interpreted as an element in $\mathbb{Z}_p$).

2.3 A Non-Interactive Lattice-Based Identification Scheme

Next, we describe how we obtain positive results in the lattice setting (Theorem 1.6). We consider a natural extension of Lyubashevky's three-message identification protocol [Lyu09], which can be seen as a lattice analogue to the Schnorr protocol.[4]

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 Π then executes as follows:

- The prover samples a short vector $\mathbf{t}$ and sends $\boldsymbol{\alpha} := \mathbf{At} \mod q$.
- The verifier responds by sending a random vector $\mathbf{c}$ with small entries.
- The prover responds with $\mathbf{z} := \mathbf{t} + \mathbf{Rc}$.
- 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 [Lyu09], 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 $\mathbf{h}$.[5]
- **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 2.4. *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 2.4 can be proved as follows. If the condition in Theorem 2.4 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 [MP12].

[4] Lyubashevky's original protocol [Lyu09] uses a scalar challenge; in this work we consider an extension to vector-valued challenges.

[5] $\mathbf{Y}$ 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.

Extensions. In Section 5, we study several variants of Π for the purposes of handling security against the *verifier* (e.g., zero-knowledge):

- In its most basic variant, we instantiate Π using noise flooding to ensure (single-theorem) zero-knowledge in the common random string (CRS) model. This gives a conceptually simple protocol closely related to the Schnorr protocol over groups, but at the cost of being less practically efficient. We note that to obtain zero-knowledge, we require a *family* of hash functions indexed by the CRS (although soundness can be argued for deterministic hash functions).

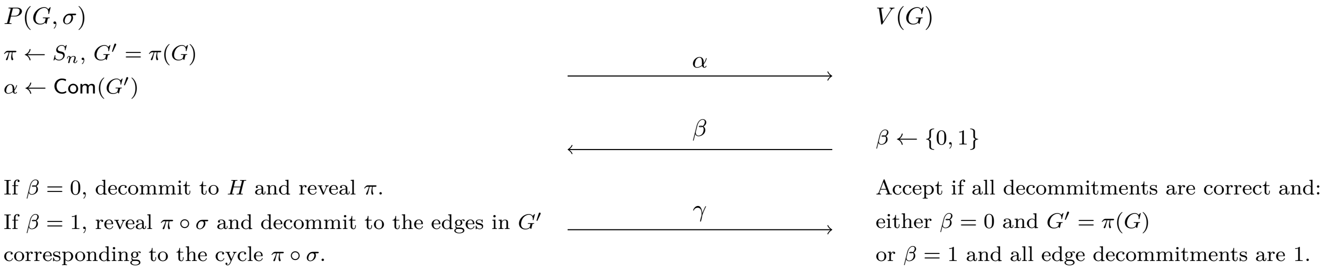
- We also consider more efficient protocols that use rejection sampling [Lyu08, Lyu09, Lyu12], where the prover aborts the execution of the protocol with some probability to ensure that the transcript is independent of his secret. Those protocols are in the plain model, but only guarantee witness indistinguishability. Note that because the prover has to run his algorithm several times in his head until it does not abort, the resulting non-interactive protocol is not directly the result of applying the Fiat-Shamir heuristic as is, but rather a "Fiat-Shamir with aborts" [Lyu09].

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 [LW15] using a Micciancio-Peikert trapdoor [MP12]. 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 other words, our protocol can be viewed as having the prover sample a preimage for some matrix with a Micciancio-Peikert trapdoor. On the flip side, the Lyubashevsky-Wichs trapdoor preimage sampling algorithm of [LW15] 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 [Lyu09]. More generally, this seems to hint at a potential connection between seemingly orthogonal paths to obtain signatures from lattice-based assumptions: one relying on lattice trapdoors and trapdoor preimage sampling ([GPV08, CHKP10, MP12]) and another through Fiat-Shamir [Lyu08, Lyu09, Lyu12].

2.4 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 [Blu86] in which the commitment scheme is instantiated with a random oracle.


$P(G, \sigma)$
$V(G)$
$\pi \leftarrow S_n$, $G' = \pi(G)$
$\alpha \leftarrow \mathsf{Com}(G')$
α
β
$\beta \leftarrow \{0, 1\}$
γ

If $\beta = 0$, decommit to H and reveal π.
If $\beta = 1$, reveal $\pi \circ \sigma$ and decommit to the edges in G' corresponding to the cycle $\pi \circ \sigma$.

Accept if all decommitments are correct and:
either $\beta = 0$ and $G' = \pi(G)$
or $\beta = 1$ and all edge decommitments are 1.

Figure 1: The Zero Knowledge Proof System Π^{Blum} for Graph Hamiltonicity.

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

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 Π^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 $\Pi^t_{\mathrm{FS},h}$ 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 $\Pi_{\mathrm{FS},f}$, 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 α_1 that can successfully answer either a "0" challenge or a "1" challenge. Therefore, if α_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 α_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 $1 \le i \le t, 1 \le \ell \le q$, 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 (α, y). In the case $h = (f, \ldots, f)$ as above, this condition follows immediately. We show in Section 6 (Lemma 6.1) that for *any* h, as long as $q = \omega(t)$, Step (2) has a solution with high probability over (α, 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" Definition 6.8, a security property which captures the analogous problems for a wide class of protocols Π.

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

- Given a challenge β, it is possible to sample a (pseudorandom) first message α along with an accepting response γ for α, 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 Section 6 for more details on the extent to which the result generalizes.

3 Preliminaries

In cryptography, the security parameter (denoted as λ) is a variable that is used to parameterize the computational complexity of the cryptographic algorithm or protocol, and the adversary's probability of breaking security. An algorithm is "efficient" if it runs in (probabilistic) polynomial time over λ.

Let $\mathbb{R}, \mathbb{Z}, \mathbb{N}$ be the set of real numbers, integers and positive integers. For $q \in \mathbb{N}_{\geq 2}$, denote $\mathbb{Z}/q\mathbb{Z}$ by $\mathbb{Z}_q$. For $n \in \mathbb{N}$, let $[n] := \{1, ..., n\}$. A vector in $\mathbb{R}^n$ (represented in column form by default) is written as a bold lower-case letter, e.g. $\mathbf{v}$. For a vector $\mathbf{v}$, the i^{th} component of $\mathbf{v}$ will be denoted by v_i. A matrix is written as a bold capital letter, e.g. $\mathbf{A}$. We denote the transpose of a matrix $\mathbf{A}$ (resp. of a vector $\mathbf{v}$) as $\mathbf{A}^T$ (resp. $(\mathbf{v}^T)$). For matrices $\mathbf{A}, \mathbf{B}$, we denote their horizontal concatenation as $[\mathbf{A} \| \mathbf{B}]$. The i^{th} column vector of $\mathbf{A}$ is denoted $\mathbf{a}_i$. The infinity norm of a vector $\mathbf{v}$ is defined as $\|\mathbf{v}\|_\infty := \max_i\{|v_i|\}$. When a variable v is drawn uniformly random from the set S we denote as $v \leftarrow \mathcal{U}(S)$ or $v \leftarrow S$.

Definition 3.1 (Fiat-Shamir transformation [FS87])**.** Given a three round public coin interactive protocol Π, the Fiat-Shamir transformation with hash function family $\mathcal{H}$ (possibly a singleton) syntactically transform Π to a non-interactive protocol $\Pi_{\mathrm{FS},\mathcal{H}}$ as follows. Sample $h \leftarrow \mathcal{H}$ and let h be the common reference string. The prover in $\Pi_{\mathrm{FS},\mathcal{H}}$ runs the prover in Π on h to obtain the first message α, then compute $\beta = h(\alpha)$, then runs the prover in Π on $h, \alpha, h(\alpha)$ to obtain the third message γ. The prover in $\Pi_{\mathrm{FS},\mathcal{H}}$ then outputs α, β, γ as the proof.

The security properties for the non-interactive protocol vary in the applications. We will explicitly define them when needed.

4 Fiat-Shamir in the Generic Group Model

4.1 Generic Group Model Preliminaries

The generic group model (GGM) [Nec94, Sho97] is an idealization of a cryptographic group in which the representation of a group element leaks no information about the underlying exponent beyond what can be learned through honest group operations. This is typically formalized by an oracle interface that implements the group operations. Each group element is represented by a randomly chosen "label," and the attacker interacts with the oracle to perform meaningful operations on the labels. A generic group attacker is measured by the number of oracle queries, but is otherwise computationally unbounded.

Definition 4.1 (Generic Group Model, Standard Formulation)**.** In the generic group model, a cyclic group of order p is represented with a label space of size $L \geq p \cdot 2^\lambda$. The generic group labeling function is a randomly sampled injection $\sigma : \mathbb{Z}_p \to [L]$. For any $x \in \mathbb{Z}_p$, the corresponding $\sigma(x) \in [L]$ is the label representing g^x.

In any application of the GGM, an attacker $\mathcal{A}$ is initialized with a list of labels $(\tau_1 = \sigma(1), \ldots, \tau_N = \sigma(x_N))$; the number N of initial labels as well as how each x_i is sampled will depend on the particular application. The attacker is usually given access to a canonical group generator, which can be formalized by requiring that $\tau = \sigma(1)$ be included in the set of initial labels.

The attacker $\mathcal{A}$ is given oracle access to the group operation oracle $\mathcal{O}_G(\cdot, \cdot)$, which on input $\tau_1, \tau_2 \in [L]$ does the following:

- If either of $\sigma^{-1}(\tau_1)$ or $\sigma^{-1}(\tau_2)$ are undefined, return $\perp$.

- Otherwise, set $x = \sigma^{-1}(\tau_1)$ and $y = \sigma^{-1}(\tau_2)$, compute $x + y \in \mathbb{Z}_p$, and return $\sigma(x + y)$.

We remark that $\mathcal{O}_G$ suffices to implement all of the standard group element manipulations. Raising a known group element to an arbitrary exponent $a \in \mathbb{Z}_p$ can be done via repeated squaring with $O(\log p)$ queries to $\mathcal{O}_G$. Computing the inverse of a group element is equivalent to raising the group element to the exponent $p - 1$, and the attacker is explicitly given p as input.

A cryptographic application is said to be (T, ϵ)-secure in the GGM if a (computationally unbounded) T-query attacker $\mathcal{A}$ cannot succeed with advantage greater than ϵ (over the randomness of the application and the labeling function σ).

Discrete Log and Linear Relations. Throughout this section, we will rely on a theorem of Shoup [Sho97] stating that discrete log is hard in the GGM. Recall that in the discrete-log problem, the attacker $\mathcal{A}$ is instantiated with labels $(\sigma(1), \sigma(x))$ for a random $x \leftarrow \mathbb{Z}_p$, and it wins if it can output x.

Theorem 4.2 (Hardness of Discrete Log [Sho97])**.** *The discrete-log problem is $(T, O(T^2/p))$-secure in the GGM.*

An almost immediate corollary of Shoup's result is that if a GGM attacker $\mathcal{A}$ is instantiated with d random group elements, it is hard to find a non-trivial linear relation among them. Formally, in the linear relation problem parameterized by $d \geq 1$, $\mathcal{A}$ is instantiated with labels $(\sigma(1), \sigma(x_1), \ldots, \sigma(x_d))$ where $x_1, \ldots, x_d$ are all uniformly random in $\mathbb{Z}_p$, and wins if it outputs a non-zero vector $\vec{\alpha} \in \mathbb{Z}_p^{d+1}$ such that $\langle \vec{\alpha}, (1, x_1, \ldots, x_d) \rangle = 0$ over $\mathbb{Z}_p$.

Theorem 4.3 (Hardness of Finding a Linear Relation)**.** *The linear relation problem with parameter d is $(T, O(dT^2/p))$-secure in the GGM.*

Proof. A T-query attacker $\mathcal{A}$ solving outputting a linear relation $\vec{\alpha}$ with advantage $\epsilon(\lambda)$ implies a T-query attacker $\mathcal{A}$ for discrete-log with advantage $\epsilon(\lambda)/d$. The reduction randomly samples $d-1$ uniformly random group elements and places the discrete-log challenge $\sigma(u)$ in a random position. At least one of the entries of $\vec{\alpha}$ other than the first entry must be non-zero, so a non-zero entry of $\vec{\alpha}$ coincides with the random position of the discrete-log challenge with probability at least $1/d$ independent of the attacker's view. If this occurs and the attacker succeeds, the reduction can solve for u. □

4.1.1 An Alternative Formulation of the GGM

For our purposes, it will be more convenient to think of the GGM as an interface that permits an attacker to perform arbitrary linear queries, but nothing else.

Definition 4.4 (Generic Group Model, Linear-Query Formulation)**.** The setup is the same as the previous formulation of the GGM, except the oracle $\mathcal{O}_G$ is replaced by a linear-query oracle $\mathcal{O}_{Lin}$.

$\mathcal{C}$ initializes $\mathcal{A}$ with the labels $\tau_1 = \sigma(x_1), \ldots, \tau_N = \sigma(x_N)$ and the group generator $\tau = \sigma(1)$. $\mathcal{O}_{Lin}$ takes as input $\alpha_1, \ldots, \alpha_N, \beta \in \mathbb{Z}_p$, and outputs

$$\sigma((\sum_{i \in [N]} \alpha_i \cdot x_i) + \beta).$$

Generic group model security proofs frequently rely on the equivalence of these two formulations. For the sake of completeness, we state this equivalence in the following claims.

Claim 4.5. *If an application is (T, ϵ)-secure in the linear-query GGM, then the application is $(T, \epsilon + O(T/2^\lambda))$-secure in the standard GGM.*

Proof. We prove that a T-query attacker $\mathcal{A}$ in the standard GGM attaining advantage ϵ implies a T-query attacker $\mathcal{A}'$ in the linear-query GGM attaining advantage $\epsilon - O(T/2^\lambda)$.

Let E be the event that the attacker $\mathcal{A}$ ever queries $\mathcal{O}_G$ on a label τ which is not the output of a prior query to $\mathcal{O}_G$, or one of the elements $\mathcal{A}$ is initialized with. Since σ is a random injection from $\mathbb{Z}_p$ to $[L]$ where $L \geq p \cdot 2^\lambda$, any label it tries which is not the result of a prior query to $\mathcal{O}_G$ will have a valid preimage under σ with probability at most $O(\frac{1}{2^\lambda})$. A union bound over all T queries shows that E occurs with probability at most $O(\frac{T}{2^\lambda})$.

Conditioned on $\neg\mathsf{E}$, any query that $\mathcal{A}$ makes to $\mathcal{O}_G$ can be perfectly replaced by a single query to $\mathcal{O}_{Lin}$. We argue this by induction on the queries. The first query that $\mathcal{A}$ makes to $\mathcal{O}_G$ is can be represented as a linear combination of (the preimages of) the initial labels $(\sigma(1), \sigma(x_1), \ldots, \sigma(x_N))$ since $\neg\mathsf{E}$ implies the inputs to $\mathcal{O}_G$ are in this list. For the inductive step, suppose each of the first i queries to $\mathcal{O}_G$ can be represented as a linear combination of (the preimages of) the initial labels $(\sigma(1), \sigma(x_1), \ldots, \sigma(x_N))$. Given $\neg\mathsf{E}$, the query (τ_1, τ_2) to $\mathcal{O}_G$ must be from the results of prior queries to $\mathcal{O}_G$ or the initial labels. But all such labels are linear combinations of the initial labels, so this must be true for query $i+1$. It is straightforward to recover the coefficients $(\alpha_1, \ldots, \alpha_N, \beta)$ for the query $i+1$ given the initial labels and the input/output transcript of the first i queries. □

Claim 4.6. *If an application is (T, ϵ)-secure in the standard GGM, then the application is $(T/(\Theta(N \log p), \epsilon)$-secure in the linear-query GGM.*

Proof. Any query to $\mathcal{O}_{Lin}$ can be simulated with $\Theta(N \log p)$ total queries to $\mathcal{O}_G$. Each $\sigma(\alpha_i x_i)$ as well as $\sigma(\beta)$ can be computed in $O(\log p)$ queries to $\mathcal{O}_G$ by repeated squaring. Combining these labels to obtain $\sigma((\sum_{i \in [N]} \alpha_i \cdot x_i) + \beta)$ takes an additional $\Theta(N)$ queries. The cost is dominated by the first step, which takes $\Theta(N \log p)$ queries. □

4.2 Chaum-Pedersen Protocol

The Chaum-Pedersen protocol (see Fig. 2) gives an interactive proof of membership for the language Diffie-Hellman tuples

$$\mathcal{L}_{\mathrm{DH}} \coloneqq \{(g, g^u, g^v, g^{uv})\}_{u,v \in \mathbb{Z}_p}.$$

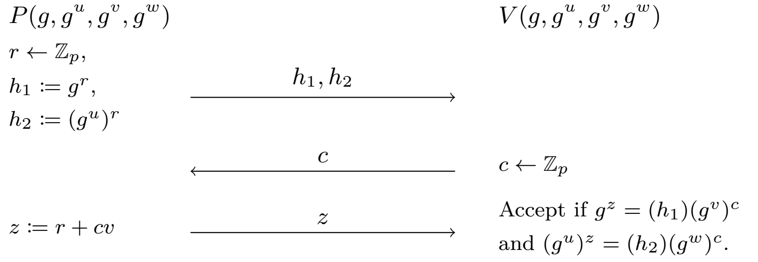


Figure 2: Protocol Π^{CP} for proving validity of a DDH tuple.

We compile the Chaum-Pedersen protocol into a non-interactive protocol satisfying *semi-adaptive* soundness for the $\mathcal{L}_{\mathrm{DH}}$ language. In contrast to fully adaptive soundness, in which the cheating prover attempts to convince the verifier to accept an arbitrary NO-instance of $\mathcal{L}_{\mathrm{DH}}$, the semi-adaptive attacker is forced to give a NO-instance whose second group element g^u is sampled at random. It then picks g^v and g^w such that $(g, g^u, g^v, g^w) \notin \mathcal{L}_{\mathrm{DH}}$, and wins if the verifier accepts.

In this section, we prove that any fixed Fiat-Shamir hash function $h : G^4 \to \mathbb{Z}_p$ satisfying the following information theoretic notion suffices to compile Chaum-Pedersen protocol into a semi-adaptively sound argument for $\mathcal{L}_{\mathrm{DH}}$.

Definition 4.7 (Well-Spread on Repeated Random Inputs)**.** A hash function $H : [L]^d \to \mathbb{Z}_p$ is well-spread on repeated random inputs if for all choices of $i_1, ..., i_d \in [d]$,

$$\mathbf{H}_\infty(H(x_{i_1}, \ldots, x_{i_d}) : (x_1, \ldots, x_d) \leftarrow [L]^d) = \omega(\log \lambda).$$

A simple example of a hash function satisfying Definition 4.7 is the Sum-Mod-p hash function H_{sum}. On input $(x_1, \ldots, x_d)$, the function H_{sum} computes $\sum_i x_i$ (identifying each input $x_i \in [L]$ with its value as an integer) and outputs the result modulo p.

For the protocol Π^{CP}, we will consider Fiat-Shamir hash function $H : g^v, g^w, h_1, h_2 \mapsto c \in \mathbb{Z}_p$, as for semi-adaptive soundness g^u is picked uniformly.

Theorem 4.8. *The protocol $(\Pi^{CP})_{FS,H}$ is semi-adaptively sound in the generic group model if the Fiat-Shamir hash function $H : [L]^4 \to \mathbb{Z}_p$ is well-spread on repeated random inputs (Definition 4.7).*

Proof. We consider the generic group model in the linear-query formulation (Definition 4.4) which suffices by Claim 4.5.

Suppose a generic group attacker $\mathcal{A}$ breaks the semi-adaptive soundness of the non-interactive protocol with advantage $\epsilon(\lambda)$. This means $\mathcal{A}$ instantiated with input $(\sigma(1), \sigma(u))$ will, with probability $\epsilon(\lambda)$ over u and σ, output $(\tau_v, \tau_w) \in [L]^2$ corresponding to a "no instance" of the DDH language, accompanied by an accepting proof $((\tau_r, \tau_s), z)$ where $(\tau_r, \tau_s) \in [L]^2$ and $z \in \mathbb{Z}_p$.

Explicitly, $\mathcal{A}$ outputs $(\tau_v, \tau_w, \tau_r, \tau_s, z)$ satisfying the following conditions with probability $\epsilon(\lambda)$:

- (Condition 1: $(\sigma(u), \tau_v, \tau_w)$ is not a valid DDH tuple) Over $\mathbb{Z}_p$, $u \cdot \sigma^{-1}(\tau_v) \neq \sigma^{-1}(\tau_w)$. If either of $\sigma^{-1}(\tau_v)$ or $\sigma^{-1}(\tau_w)$ do not exist, this condition is failed.

- (Condition 2: the verifier accepts (τ_r, τ_s, z)). The two checks the verifier performs are:

$$z = \sigma^{-1}(\tau_r) + \sigma^{-1}(\tau_v) \cdot H(\tau_v, \tau_w, \tau_r, \tau_s), \tag{2}$$

$$u \cdot z = \sigma^{-1}(\tau_s) + \sigma^{-1}(\tau_w) \cdot H(\tau_v, \tau_w, \tau_r, \tau_s). \tag{3}$$

  In the real protocol, the verifier is checking these relations in the exponent, but the cheating prover must still satisfy them over $\mathbb{Z}_p$. If either of $\sigma^{-1}(\tau_r)$ or $\sigma^{-1}(\tau_w)$ do not exist, this condition is failed.

Recall that in the linear-query formulation of the GGM (cf. Definition 4.4), any label the attacker $\mathcal{A}$ obtains from the group oracle $\mathcal{O}_{lin}$ is of the form $\sigma(\alpha \cdot u + \beta)$, where α and β are known to the attacker (since $\mathcal{A}$ explicitly provides α, β to make the query). If any of the labels $\tau_v, \tau_w, \tau_r, \tau_s$ that $\mathcal{A}$ outputs are *not* the result of a query to $\mathcal{O}_{lin}$ (or one of $\sigma(1)$ or $\sigma(u)$), then with probability $1 - O(1/2^\lambda)$ (over the randomness of σ) there will not exist a preimage under σ and the conditions will fail.

Therefore, for any attacker $\mathcal{A}$, we can define an attacker that directly outputs those coefficients with overwhelming probability. Up to renaming, we can therefore think of $\mathcal{A}$ as *directly* outputting coefficients $\alpha_v, \beta_v, \alpha_w, \beta_w, \alpha_r, \beta_r, \alpha_s, \beta_s \in \mathbb{Z}_p$ such that

$$\begin{aligned} v &= \alpha_v \cdot u + \beta_v, \\ w &= \alpha_w \cdot u + \beta_w, \\ r &= \alpha_r \cdot u + \beta_r, \\ s &= \alpha_s \cdot u + \beta_s, \end{aligned}$$

in place of $\tau_v, \tau_w, \tau_r, \tau_s$, as this can only hurt its advantage by an additive $O(1/2^\lambda)$.

We rewrite Eqs. (2) and (3) in terms of $\vec{\alpha} \coloneqq (\alpha_v, \alpha_w, \alpha_r, \alpha_s)$ and $\vec{\beta} \coloneqq (\beta_v, \beta_w, \beta_r, \beta_s)$:

$$z = (\alpha_r \cdot u + \beta_r) + (\alpha_v \cdot u + \beta_v) \cdot f(\vec{\alpha}, \vec{\beta}), \tag{4}$$

$$u \cdot z = (\alpha_s \cdot u + \beta_s) + (\alpha_w \cdot u + \beta_w) \cdot f(\vec{\alpha}, \vec{\beta}), \tag{5}$$

where

$$f(\vec{\alpha}, \vec{\beta}) \coloneqq H(\sigma(\alpha_v + u \cdot \beta_v), \sigma(\alpha_w + u \cdot \beta_w), \sigma(\alpha_r + u \cdot \beta_r), \sigma(\alpha_s + u \cdot \beta_s)).$$

In these equations, the attacker $\mathcal{A}$ outputs (or can efficiently compute) every value except for u. If these equations can be solved for u, this means the attacker can break discrete log with the same probability that can satisfy these equations. By Theorem 4.2, a T-query attacker can only break discrete log with probability $O(T^2/p)$.

Therefore, with probability $\epsilon(\lambda) - O(1/2^\lambda) - O(T^2/p)$, the attacker outputs $\vec{\alpha}, \vec{\beta}, z$ satisfying Eqs. (4) and (5), and furthermore these equations *cannot* be solved for u. In other words, these equations should not be formally solvable in u as a formal variable. This gives rise to the following four equations, where the first two state the coefficient of u must be equal on both sides of Eqs. (4) and (5), and the last two are the result of setting theconstant terms to be equal.

$$0 = \alpha_r + \alpha_v \cdot f(\vec{\alpha}, \vec{\beta}), \tag{6}$$

$$z = \alpha_s + \alpha_w \cdot f(\vec{\alpha}, \vec{\beta}), \tag{7}$$

$$z = \beta_r + \beta_v \cdot f(\vec{\alpha}, \vec{\beta}), \tag{8}$$

$$0 = \beta_s + \beta_w \cdot f(\vec{\alpha}, \vec{\beta}). \tag{9}$$

We finish the proof by showing that if $\vec{\alpha}, \vec{\beta}, z$ does not correspond to a valid DDH tuple, then over the randomness of σ, these equations can only hold with probability $O(T^4/\lambda^{\omega(1)})$. This means $\epsilon(\lambda) - O(1/2^\lambda) - O(T^2/p) \leq O(T^4/\lambda^{\omega(1)})$, from which the claimed bound of $\epsilon(\lambda) \leq O(T^4/\lambda^{\omega(1)})$ follows.

Suppose for a moment that $f(\vec{\alpha}, \vec{\beta})$ is replaced by a formal variable $\mathbf{f}$ in each of Eqs. (6) to (9). Then for any particular choice of $\vec{\alpha}, \vec{\beta}$, either (1) none of these equations have any formal dependence on $\mathbf{f}$ or (2) at least one of these equations determines $\mathbf{f}$ (if more than one equation determines $\mathbf{f}$ there may be no solution).

If (1) is the case, then the coefficients of $\mathbf{f}$ must be equal on both sides in all four equations. This gives rise to the following conditions on $\vec{\alpha}, \vec{\beta}, z$:

$$\begin{aligned} \alpha_v &= 0, \\ \alpha_r &= 0, \\ \alpha_w &= \beta_v, \\ \alpha_s &= \beta_r, \\ \beta_w &= 0, \\ \beta_s &= 0, \\ z &= \beta_r + \beta_v \cdot f(\vec{\alpha}, \vec{\beta}). \end{aligned}$$

These conditions are equivalent to $w = u \cdot v, s = u \cdot r$, and $z = r + H(\tau_v, \tau_w, \tau_r, \tau_s) \cdot v$, which precisely corresponds to an honest execution of the protocol on a valid DDH tuple.

However, since $\mathcal{A}$ is outputting $(\vec{\alpha}, \vec{\beta}, z)$ corresponding to a DDH "no instance" (with probability $\epsilon(\lambda) - O(1/2^\lambda) - O(T^2/p)$) this cannot correspond to (1). Therefore, it must be that the choice of $\vec{\alpha}, \vec{\beta}$ determines $\mathbf{f}$.

Observe that there are at most $(T+2)^4$ possible choices for $\vec{\alpha}, \vec{\beta}$, since we are already conditioning on the attacker outputting only (α, β) pairs corresponding to one of the T queries it made $\mathcal{O}_{Lin}$, or one of the two group elements $\sigma(1), \sigma(u)$ (i.e. $(\alpha, \beta) = (0, 1)$ or $(1, 0)$).

For any fixed choice of $\vec{\alpha}, \vec{\beta}$, the value $f(\vec{\alpha}, \vec{\beta})$ will be an evaluation of H on four generic group labels $(\tau_v, \tau_w, \tau_r, \tau_s)$. These generic group labels may be repeated, but the well-spread property of H (Definition 4.7) guarantees that H still has min-entropy $\omega(\log \lambda)$. So the probability that this value of H will equal the prescribed setting for $\mathbf{f}$ is at most $1/2^{\omega(\log \lambda)}$. By a union bound over all $(T+2)^4$ possible choices of $(\vec{\alpha}, \vec{\beta})$, the probability that the attacker satisfies all the equations is at most $(T+2)^4/2^{\omega(\log \lambda)}$. □

4.3 Application: NIZKs for NP

We now show that $(\Pi^{\mathrm{CP}})_{FS,H}$ can be used to obtain NIZKs for all of NP. This follows the recent line work of instantiating the hidden-bits model [FLS99] from standard assumptions [CH19, KNYY19, QRW19, CKU20]. In particular, Couteau, Katsumata and Ursu [CKU20, Theorem 28] show that any NIZK for the language

$\mathcal{L}_{\mathrm{DH}}$ is sufficient to build so-called Verifiable Pseudorandom Generators (VPRG) [CH19] (also known as hidden bits generators [QRW19]), which in turn allows to instantiate the hidden bits model [CH19, KNYY19, QRW19, CKU20].

While the statement of [CKU20, Theorem 28] specifies that the underlying NIZK for $\mathcal{L}_{\mathrm{DH}}$ be *adaptively sound*, we note that our notion of semi-adaptive soundness suffices. This is because in the proof of [CKU20, Theorem 28] the g^u component of the Diffie-Hellman tuple is randomly sampled and included in the common reference string of the VPRG; this is something which the malicious prover does not have any control over. This gives the following theorem:

Theorem 4.9 (NIZKs for $\mathcal{L}_{\mathrm{DH}}$ imply NIZKs for all of NP, adapted from [CKU20])**.** *Suppose* Π *is a semi-adaptively sound, single-theorem zero-knowledge NIZK argument for* $\mathcal{L}_{\mathrm{DH}}$*. Then, under the CDH assumption, there exists an (adaptively sound, adaptively multi-theorem) NIZK argument for all of* NP*.*

As is, our protocol $(\Pi^{\mathrm{CP}})_{FS,H}$ is in the plain model, and is therefore not zero-knowledge (assuming deciding DDH is not in BPP). However, one can generically add single-theorem zero-knowledge in the following way. We now use a common random string $\mathsf{crs} := \rho \leftarrow \mathbb{Z}_p$ and define our new hash function $H_\rho = H + \rho$. This makes H_ρ 1-wise independent, and allows to lift honest-verifier zero-knowledge of Π^{CP} to single-theorem zero-knowledge of $(\Pi^{\mathrm{CP}})_{FS,H}$ in the CRS model (by having the simulator program ρ to map the challenge c to the honest-verifier simulator challenge).

5 Lattice-Based Identification Protocols

5.1 Preliminaries

We review basic definitions and lemmas we will use throughout the section.

Lemma 5.1 (Noise flooding)**.** *Let* $B = B(\lambda)$*,* $B' = B'(\lambda)$ *be integers such that* $B'/B = \mathrm{negl}(\lambda)$*. Then for all* $x \in [-B', B']$*, the distributions* $\mathcal{U}([-B, B] + x)$ *and* $\mathcal{U}([-B, B])$ *are within negligible statistical distance from each other.*

Lemma 5.2 (Leftover Hash Lemma)**.** *Let* $\mathcal{H} = \{h : \mathcal{X} \to \mathcal{Y}\}$ *be a 2-universal hash function family. Then for any random variable* $X \in \mathcal{X}$*, for* $\epsilon > 0$ *s.t.* $\log(|\mathcal{Y}|) \leq H_\infty(X) - 2\log(1/\epsilon)$*, the distributions*

$$(h, h(X)) \text{ and } (h, \mathcal{U}(\mathcal{Y}))$$

are ϵ*-statistically close.*

Furthermore, the family $\{\mathbf{A} \in \mathbb{Z}_q^{n \times m} : \mathbf{r} \mapsto \mathbf{Ar}\}$ *is 2-universal for prime* q*.*

SIS and LWE. We first recall the short integer solution (SIS) problem.

Definition 5.3 (Short Integer Solution (SIS) [Ajt96])**.** For any $n, m, q \in \mathbb{Z}$ and $B \in \mathbb{R}$, define the short integer solution problem $SIS_{n,m,q,B}$ as follows: Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\|_\infty \leq B$, and

$$\mathbf{Ax} = \mathbf{0} \bmod q.$$

Definition 5.4 (Inhomogeneous Short Integer Solution (iSIS))**.** For any $n, m, q \in \mathbb{Z}$ and $B \in \mathbb{R}$, define the inhomogeneous short integer solution problem $iSIS_{n,m,q,B}$ as follows: Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{y} \in \mathbb{Z}_q^n$, find $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\|_\infty \leq B$, and

$$\mathbf{Ax} = \mathbf{y} \bmod q.$$

Lemma 5.5 (Hardness of (i)SIS based on the lattice problems in the worst case [Ajt96, GPV08])**.** *For any* $m = \Omega(n \log q)$*, any* $\beta > 0$*, and any sufficiently large* $q \geq \beta \cdot poly(n)$*, solving* $SIS_{n,m,q,\beta}$ *or* $iSIS_{n,m,q,\beta}$ *(where* $\mathbf{y}$ *is sampled uniformly from* $\mathbb{Z}_q^n$*) with non-negligible probability is as hard as solving* $GapSVP_\gamma$ *and* $SIVP_\gamma$ *on arbitrary* n*-dimensional lattices with overwhelming probability, for some approximation factor* $\gamma = \beta \cdot poly(n)$*.*

We recall the decisional learning with errors (LWE) problem.

Definition 5.6 (Decisional Learning with Errors (LWE) [Reg05]). For $n, m \in \mathbb{N}$ and modulus $q \geq 2$, distributions for secret vectors, public matrices, and error vectors $\theta, \pi, \chi \subseteq \mathbb{Z}_q$. An LWE sample is obtained from sampling $\mathbf{s} \leftarrow \theta^n$, $\mathbf{A} \leftarrow \pi^{n \times m}$, $\mathbf{e} \leftarrow \chi^m$, and outputting $(\mathbf{A}, \mathbf{y}^t := \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \mod q)$.

We say that an algorithm solves $LWE_{n,m,q,\theta,\pi,\chi}$ if it distinguishes the LWE sample from a random sample distributed as $\pi^{n \times m} \times \mathcal{U}(\mathbb{Z}_q^m)$ with probability greater than 1/2 plus non-negligible.

Lemma 5.7 (Hardness of LWE based on the lattice problems in the worst case [Reg05]). *Given $n \in \mathbb{N}$, for any $m = poly(n)$, $q \leq 2^{poly(n)}$. Let $\theta = \pi = \mathcal{U}(\mathbb{Z}_q)$, $\chi = D_{\mathbb{Z},s}$, the discrete Gaussian distribution of width $s \geq 2\sqrt{n}$. If there exists an efficient (possibly quantum) algorithm that breaks $LWE_{n,m,q,\theta,\pi,\chi}$, then there exists an efficient (possibly quantum) algorithm for solving $GapSVP_\gamma$ and $SIVP_\gamma$ on arbitrary n-dimensional lattices with overwhelming probability, for some approximation factor $\gamma = \tilde{O}(nq/s)$.*

The next lemma shows that LWE with the secret sampled from the error distribution is as hard as the standard LWE.

Lemma 5.8 ([ACPS09]). *For n, m, q, s chosen as in Lemma 5.7, $LWE_{n,m',q,D_{\mathbb{Z},s},\mathcal{U}(\mathbb{Z}_q),D_{\mathbb{Z},s}}$ is as hard as $LWE_{n,m,q,\mathcal{U}(\mathbb{Z}_q),\mathcal{U}(\mathbb{Z}_q),D_{\mathbb{Z},s}}$ for $m' \leq m - (16n + 4\log\log q)$.*

Throughout the paper we will denote by $LWE_{n,m,q,\chi}$ the assumption implicitly setting $\theta = \chi$, $\pi = \mathcal{U}(\mathbb{Z}_q)$.

Definition 5.9 (Gadget Matrix). We say that a matrix $\mathbf{G} \in \mathbb{Z}_q^{k \times \ell}$ is a *gadget matrix* if there exists an efficient deterministic procedure $\mathbf{G}^{-1}$, which, on input $\mathbf{X} \in \mathbb{Z}_q^k$, output a matrix $\mathbf{G}^{-1}(\mathbf{X})$ with small norm such that $\mathbf{G}\mathbf{G}^{-1}(\mathbf{X}) = \mathbf{X}$. A common choice of the gadget matrix is the following "power-of-b" matrix, where the base b is a small integer (say $b = 2$). Let $\mathbf{G} = \mathbf{I}_k \otimes \mathbf{g}^t \in \mathbb{Z}_q^{k \times k\lceil \log_b q \rceil}$ with $\mathbf{g}^t = (1, b, \ldots, b^{\lceil \log_b q \rceil - 1})$ (implicitly setting $\ell = k\lceil \log_b q \rceil$). The $\mathbf{G}^{-1}$ function is then the base-b decomposition function. By default we will consider the "power-of-two" gadget matrix, but all our results apply with any matrix $\mathbf{G}$ with the following property:

- There exists a deterministic function $\mathbf{G}^{-1}(\cdot)$, which on input $\boldsymbol{\alpha} \in \mathbb{Z}_q^k$ outputs a short $\mathbf{c}$ such that $\mathbf{G}(\mathbf{c}) = \boldsymbol{\alpha}$,

Looking ahead, if we do not use the "powers-of-two" gadge matrix, the "shortness" of $\mathbf{c} = \mathbf{G}^{-1}(\boldsymbol{\alpha})$ will slightly modify the parameters of the schemes, namely the final check of the verifier with respect to the norm of the third message, and the parameters underlying SIS problem used to argue soundness.

5.2 Identification Protocols based on SIS

We first describe a variant of Lyubashevsky identification protocol. This can be also seen as a variant of the Schnorr protocol ported to the SIS setting, using many secrets in parallel. For the sake of simplicity, we will first deal with zero-knowledge using noise flooding rather than rejection sampling; we present a version based on rejection sampling in Section 5.4.

Let n, m, q, and ℓ, B be integers.

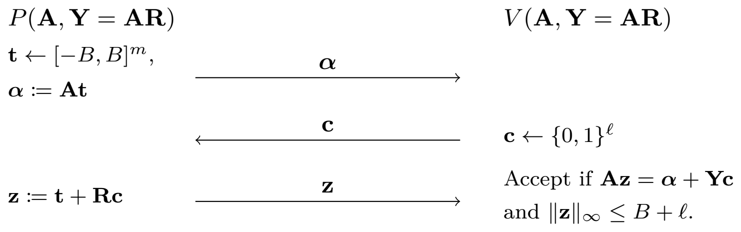

$P(\mathbf{A}, \mathbf{Y} = \mathbf{AR})$
$V(\mathbf{A}, \mathbf{Y} = \mathbf{AR})$
$\mathbf{t} \leftarrow [-B, B]^m$,
$\boldsymbol{\alpha} := \mathbf{At}$
$\boldsymbol{\alpha}$
$\mathbf{c}$
$\mathbf{c} \leftarrow \{0,1\}^\ell$
$\mathbf{z} := \mathbf{t} + \mathbf{Rc}$
$\mathbf{z}$
Accept if $\mathbf{Az} = \boldsymbol{\alpha} + \mathbf{Yc}$
and $\|\mathbf{z}\|_\infty \leq B + \ell$.

Figure 3: Identification Protocol Π^{SIS} based on SIS.

Consider the following identification protocol:

- The public key is $(\mathbf{A}, \mathbf{Y})$ where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$, and $\mathbf{Y} = \mathbf{A}\mathbf{R} \in \mathbb{Z}_q^{n \times \ell}$ where $\mathbf{R} \leftarrow \{0,1\}^{m \times \ell}$. The secret key is $\mathbf{R}$.
- The prover samples $\mathbf{t} \leftarrow [-B, B]^m$, and sends $\boldsymbol{\alpha} = \mathbf{A}\mathbf{t} \in \mathbb{Z}_q^n$ to the verifier.
- The verifier sends a challenge $\mathbf{c} \leftarrow \{0,1\}^\ell$ as the second message.
- The prover computes $\mathbf{z} = \mathbf{t} + \mathbf{R}\mathbf{c} \in \mathbb{Z}_q^m$, and sends it to the verifier .
- The verifier accepts if $\mathbf{A}\mathbf{z} = \boldsymbol{\alpha} + \mathbf{Y}\mathbf{c}$ and $\|\mathbf{z}\|_\infty \le B + \ell$.

Claim 5.10 (Completeness)**.** *The identification protocol* Π^{SIS} *is complete.*

Proof. By linearity, $\mathbf{A}\mathbf{z} = \mathbf{A}\mathbf{t} + \mathbf{A}\mathbf{R}\mathbf{c} = \boldsymbol{\alpha} + \mathbf{Y}\mathbf{c}$. Further, we have $\|\mathbf{t}\|_\infty \le B$ and $\|\mathbf{R}\mathbf{c}\|_\infty \le \ell$, so that $\|\mathbf{z}\|_\infty \le B + \ell$. □

Next, we show that Π^{SIS} satisfies special soundness. Unfortunately, we are not able to extract a short matrix $\mathbf{R}'$ such that $\mathbf{A}\mathbf{R}' = \mathbf{Y}$. Instead, we show how to obtain a short (non-zero) vector $\mathbf{r} \in \mathbb{Z}_q^{m+\ell}$ such that $[\mathbf{A}\|\mathbf{Y}] \cdot \mathbf{r} = \mathbf{0}$. Note that for uniformly random $\mathbf{A}$ and $\mathbf{Y}$, this is hard to do assuming SIS.

Claim 5.11 (Relaxed Special Soundness)**.** *Suppose that* $\boldsymbol{\alpha}, \mathbf{z}, \mathbf{z}'$ *and* $\mathbf{c} \neq \mathbf{c}'$ *such that* $(\boldsymbol{\alpha}, \mathbf{c}, \mathbf{z})$ *and* $(\boldsymbol{\alpha}, \mathbf{c}', \mathbf{z}')$ *are both accepting transcripts for* Π^{SIS}*. Then there exists an extractor* $\mathcal{E}((\boldsymbol{\alpha}, \mathbf{c}, \mathbf{z}), (\boldsymbol{\alpha}, \mathbf{c}', \mathbf{z}'))$ *that computes a non-zero element* $\mathbf{r} \in \mathbb{Z}_q^{m+\ell}$ *such that* $[\mathbf{A}\|\mathbf{Y}] \cdot \mathbf{r} = \mathbf{0}$ *and* $\|\mathbf{r}\|_\infty \le 2(B + \ell)$*.*

Proof. We have $\mathbf{z} - \mathbf{z}' = \mathbf{R}(\mathbf{c} - \mathbf{c}')$, so that $\mathbf{A}(\mathbf{z} - \mathbf{z}') = \mathbf{Y}(\mathbf{c} - \mathbf{c}')$. We distinguish two cases. Either $\mathbf{Y}(\mathbf{c} - \mathbf{c}') = \mathbf{0}$, in which case $\mathbf{r} := \begin{bmatrix} \mathbf{0} \\ \mathbf{c} - \mathbf{c}' \end{bmatrix}$ is non-zero and satisfies $[\mathbf{A}\|\mathbf{Y}] \cdot \mathbf{r} = \mathbf{0}$, where $\|\mathbf{r}\|_\infty \le 2$; or we have $\mathbf{z} - \mathbf{z}' = \mathbf{R}(\mathbf{c} - \mathbf{c}')$, so that $\mathbf{A}(\mathbf{z} - \mathbf{z}') = \mathbf{Y}(\mathbf{c} - \mathbf{c}')$. Because $\mathbf{Y}(\mathbf{c} - \mathbf{c}') \neq \mathbf{0}$, we have that $\mathbf{z} - \mathbf{z}' \neq 0$, and therefore $\mathbf{r} := \begin{bmatrix} \mathbf{z} - \mathbf{z}' \\ \mathbf{c}' - \mathbf{c} \end{bmatrix}$ is a non-zero vector such that $[\mathbf{A}\|\mathbf{Y}] \cdot \mathbf{r} = \mathbf{0}$ with $\|\mathbf{r}\|_\infty \le 2(B + \ell)$. □

Claim 5.12 (Honest-Verifier Zero-Knowledge)**.** *Suppose* $\ell/B = \mathrm{negl}(\lambda)$*. Then the identification protocol* Π^{SIS} *is statistically honest-verifier zero-knowledge.*

Proof. We define the honest-verifier simulator $\mathcal{S}$ as follows. On input $(\mathbf{A}, \mathbf{Y}, \mathbf{c})$, it samples $\mathbf{z}$ uniformly from $[-B, B]^m$, and sets $\boldsymbol{\alpha} = \mathbf{A}\mathbf{z} - \mathbf{Y}\mathbf{c}$.

For $\mathbf{c} \leftarrow \{0,1\}^\ell$, by Lemma 5.1, the resulting distribution $(\mathbf{c}, \mathbf{z})$ is statistically close to the one produced by real proofs. Given $(\mathbf{c}, \mathbf{z})$, for accepting proofs, $\boldsymbol{\alpha}$ satisfies $\boldsymbol{\alpha} = \mathbf{A}\mathbf{z} - \mathbf{Y}\mathbf{c}$ and therefore the output $(\boldsymbol{\alpha}, \mathbf{c}, \mathbf{z})$ of $\mathcal{S}$ is distributed statistically close to honestly generated proofs. □

We now show that instantiating the Fiat-Shamir heuristic on Π^{SIS} with the hash function $\mathbf{G}^{-1}(\cdot)$ (Fig. 4) preserves (average-case) soundness. In order to preserve zero-knowledge, we additionally rely on a common random string.

$$\mathsf{crs} = \boldsymbol{\rho} \leftarrow \mathbb{Z}_q^n$$

$$\begin{array}{lcl}
P(\mathbf{A}, \mathbf{Y} = \mathbf{A}\mathbf{R}, \mathbf{R}) & & V(\mathbf{A}, \mathbf{Y} = \mathbf{A}\mathbf{R}) \\
\mathbf{t} \leftarrow [-B, B]^m, & & \text{Compute } \mathbf{c} = \mathbf{G}^{-1}(\boldsymbol{\alpha} + \boldsymbol{\rho}) \\
\boldsymbol{\alpha} := \mathbf{A}\mathbf{t} & & \text{Accept if } \mathbf{A}\mathbf{z} = \boldsymbol{\alpha} + \mathbf{Y}\mathbf{c} \\
\mathbf{c} = \mathbf{G}^{-1}(\boldsymbol{\alpha} + \boldsymbol{\rho}) & \xrightarrow{\quad(\boldsymbol{\alpha}, \mathbf{z})\quad} & \text{and } \|\mathbf{z}\|_\infty \le B + \ell \\
\mathbf{z} = \mathbf{t} + \mathbf{R}\mathbf{c} & & \text{and } (\mathbf{c}, \mathbf{z}) \neq (\mathbf{0}, \mathbf{0}).
\end{array}$$

Figure 4: Non-interactive Identification Protocol $(\Pi^{\mathrm{SIS}})_{\mathrm{FS}, \mathbf{G}^{-1}}$ based on SIS.

Claim 5.13 (Completeness)**.** *The protocol* $(\Pi^{\mathrm{SIS}})_{\mathrm{FS}, \mathbf{G}^{-1}}$ *is complete.*

Proof. This follows by completeness of the interactive variant Π^{SIS}. $\square$

Claim 5.14 (Average-case soundness)**.** *Under the* $iSIS_{n,m+\ell,B+\ell}$ *assumption, we have that for all efficient cheating prover* P^* *for* $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$*:*

$$\Pr_{\mathsf{crs}\leftarrow\mathbb{Z}_q^{\ell},\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m},\mathbf{Y}\leftarrow\mathbb{Z}_q^{n\times\ell}}[(P^*(\mathsf{crs},\mathbf{A},\mathbf{Y})\leftrightarrow V(\mathsf{crs},\mathbf{A},\mathbf{Y}))=\mathrm{Accept}]\leq\mathrm{negl}(n).$$

In particular, $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$ *is a one-time secure identification scheme.*

Proof. Accepting proofs $(\boldsymbol{\alpha},\mathbf{c},\mathbf{z})$ for $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$ satisfy $\mathbf{Az}=\boldsymbol{\alpha}+\mathbf{Yc}$ where $\|\mathbf{z}\|_\infty\leq B+\ell$. This can be rewritten as

$$[\mathbf{A}\parallel\mathbf{G}+\mathbf{Y}]\begin{bmatrix}\mathbf{z}\\-\mathbf{G}^{-1}(\boldsymbol{\alpha}+\boldsymbol{\rho})\end{bmatrix}=\boldsymbol{\rho}.$$

Let $(\mathbf{B},\boldsymbol{\rho})$ be an inhomogeneous SIS instance where $\mathbf{B}=[\mathbf{B}_1\|\mathbf{B}_2]\leftarrow\mathbb{Z}_q^{n\times(m+\ell)}$, and $\boldsymbol{\rho}\leftarrow\mathbb{Z}_q^n$ and let $P^*(\mathsf{crs},\mathbf{A},\mathbf{Y})$ be a cheating prover breaking average-case soundness of $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$ with probability ϵ over the randomness of $(\mathsf{crs},\mathbf{A},\mathbf{Y})\leftarrow\mathbb{Z}_q^n\times\mathbb{Z}_q^{n\times m}\times\mathbb{Z}_q^{n\times\ell}$. Then, any accepting transcript $(\boldsymbol{\alpha},\mathbf{c},\mathbf{z})$ produced by P^* on input $(\boldsymbol{\rho},\mathbf{B}_1,\mathbf{B}_2-\mathbf{G})$ (which is distributed uniformly) induces an inhomogeneous SIS solution $\mathbf{r}=\begin{bmatrix}\mathbf{z}\\-\mathbf{G}^{-1}(\boldsymbol{\alpha}+\boldsymbol{\rho})\end{bmatrix}$ which is non-zero, as $(\mathbf{z},\mathbf{c})\neq(\mathbf{0},\mathbf{0})$, and such that $\|\mathbf{r}\|_\infty\leq B+\ell$. $\square$

Note that the proof of Claim 5.14 does not strongly rely on the randomness of ρ. In particular, we could set $\boldsymbol{\rho}=\mathbf{0}$ and still argue soundness of $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$, by relying directly on SIS instead of its inhomogeneous version in the proof. In other words, using the *single, deterministic* Fiat-Shamir hash function $\mathbf{G}^{-1}(\cdot)$ preserves soundness of Π^{SIS}; it is only for zero-knowledge that we consider a (slightly modified) *family of* hash functions $\mathbf{G}_{\boldsymbol{\rho}}^{-1}(\boldsymbol{\alpha})=\mathbf{G}^{-1}(\boldsymbol{\alpha}+\boldsymbol{\rho})$.

Next, we argue zero-knowledge of our construction. Note that the way we add the CRS to our protocol is technically different from the one we use in the group-based setting. In more details, defining $\widetilde{\mathbf{G}^{-1}}_{\boldsymbol{\rho}}(\boldsymbol{\alpha}):=\mathbf{G}^{-1}(\boldsymbol{\alpha})+\boldsymbol{\rho}$ would break the structural requirement that we use to argue soundness. Instead, we define $\mathbf{G}^{-1}{}_{\boldsymbol{\rho}}(\boldsymbol{\alpha}):=\mathbf{G}^{-1}(\boldsymbol{\alpha}+\boldsymbol{\rho})$, and we directly argue (single-theorem) zero-knowledge without using the fact that Π^{SIS} is honest-verifier zero-knowledge.

Claim 5.15 (Zero-Knowledge)**.** *The protocol* $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$ *is (single-theorem) statistically zero-knowledge.*

Proof. We define our simulator $\mathbf{S}$ as follows. On input $(\mathbf{A},\mathbf{Y})$, it samples $\mathbf{u}\leftarrow\mathbb{Z}_q^n$, and sets $\mathbf{c}=\mathbf{G}^{-1}(\mathbf{u})$. It samples $\mathbf{z}$ uniformly from $[-B,B]^m$, and sets $\boldsymbol{\rho}=[\mathbf{A}\|\mathbf{G}+\mathbf{Y}]\begin{bmatrix}\mathbf{z}\\-\mathbf{c}\end{bmatrix}$. It sets $\boldsymbol{\alpha}=\mathbf{u}-\boldsymbol{\rho}$, and outputs $(\mathsf{crs}=\boldsymbol{\rho},(\boldsymbol{\alpha},\mathbf{c},\mathbf{z}))$.

Let us justify that the simulated distribution is statistically close to the real one. In the real distribution, $\boldsymbol{\rho}$ is distributed uniformly, so $\boldsymbol{\alpha}+\boldsymbol{\rho}$ is distributed uniformly. The simulated $\mathbf{z}$ is distributed statistically close to its honestly generated counterpart, by Lemma 5.1, even conditioned on $\mathbf{c}$ and $\mathbf{u}$. Given $\mathbf{z}$ and $\mathbf{c}=\mathbf{G}^{-1}(\mathbf{u})$, the simulated $\boldsymbol{\rho}$ is entirely determined as $\boldsymbol{\rho}=[\mathbf{A}\|\mathbf{G}+\mathbf{Y}]\begin{bmatrix}\mathbf{z}\\-\mathbf{c}\end{bmatrix}$, where $\boldsymbol{\rho}$ is (taken alone) statistically close to uniform by the leftover hash lemma (over the randomness of $\mathbf{z}$). This in turn defines $\boldsymbol{\alpha}$ as $\mathbf{u}-\boldsymbol{\rho}$, which makes the distribution output by $\mathcal{S}$ statistically close to honestly generated proofs overall. $\square$

Parameters. To argue security of Π^{SIS} and $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$, we used the following properties:

- $\mathbf{G}\in\mathbb{Z}_q^{n\times\ell}$ is a gadget matrix. It suffices to set $\ell=n\lceil\log q\rceil$ to satisfy this property when instantiating $\mathbf{G}$ as the "powers-of-two" matrix. We stress that one could use any gadget matrix satisfying the requirements of Definition 5.9, albeit with slightly different parameters depending on the gadget matrix.

- $\ell/B\leq\mathrm{negl}(n)$ to argue zero-knowledge in Claims 5.12 and 5.15;

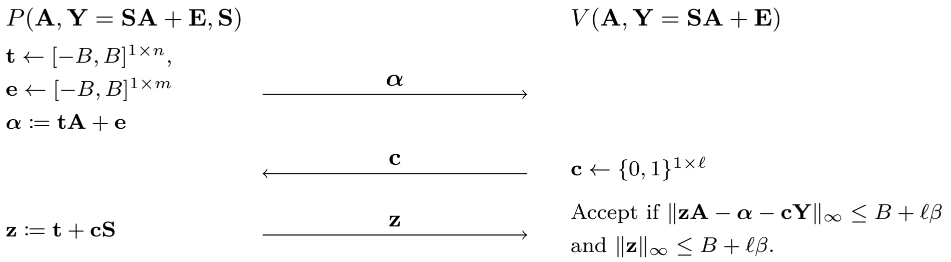
- $(\mathbf{A}, \mathbf{AR})$ (resp. $(\mathbf{A}, \mathbf{Az})$) are statistically close to uniform, to argue that relaxed special soundness of Claim 5.11 is non-vacuous (resp. zero-knowledge of $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$ in Claim 5.15). By the leftover hash lemma it suffices to set $m = 2n \log q$;
- $iSIS_{n,m+\ell,q,B+\ell}$ is hard, to argue soundness of $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$ in Claim 5.14.

Overall, setting $m = 2n\lceil \log q \rceil$, $\ell = n\lceil \log q \rceil$, $q = 2^{n^{\epsilon}}$ for any $0 < \epsilon < 1$, and any $B = n^{\omega(1)}$, our scheme is secure under $iSIS_{n,m+\ell,q,B+\ell}$ (where statistical zero-knowledge holds with statistical distance $\approx \ell/B + q^{n/2}$), and therefore under the hardness of $GapCVP$ and $SIVP$ with sub-exponential approximation factors.

5.3 Identification Protocols based on LWE

Next, we show LWE counterparts to the identification schemes above. We will consider here the Hermite Normal Form of LWE [ACPS09], where the secret is sampled from the error distribution. Looking ahead, doing so will make the third message of the protocol short, which will be crucial to analyze the soundness of our non-interactive version.

Let n, m, q, and ℓ, B be integers, and let χ be a β-bounded error distribution for some integer β.



$P(\mathbf{A}, \mathbf{Y} = \mathbf{SA} + \mathbf{E}, \mathbf{S})$

$V(\mathbf{A}, \mathbf{Y} = \mathbf{SA} + \mathbf{E})$

$\mathbf{t} \leftarrow [-B, B]^{1\times n}$,

$\mathbf{e} \leftarrow [-B, B]^{1\times m}$

$\boldsymbol{\alpha} := \mathbf{tA} + \mathbf{e}$

$\xrightarrow{\boldsymbol{\alpha}}$

$\xleftarrow{\mathbf{c}}$

$\mathbf{c} \leftarrow \{0,1\}^{1\times \ell}$

$\mathbf{z} := \mathbf{t} + \mathbf{cS}$

$\xrightarrow{\mathbf{z}}$

Accept if $\|\mathbf{zA} - \boldsymbol{\alpha} - \mathbf{cY}\|_\infty \leq B + \ell\beta$

and $\|\mathbf{z}\|_\infty \leq B + \ell\beta$.

Figure 5: Identification Protocol Π^{LWE} based on LWE.

Consider the following identification protocol:

- The public key is $(\mathbf{A}, \mathbf{Y})$ where $\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$, and $\mathbf{Y} = \mathbf{SA} + \mathbf{E} \in \mathbb{Z}_q^{\ell\times m}$ where $\mathbf{S} \leftarrow \chi^{\ell\times n}$ and $\mathbf{E} \leftarrow \chi^{\ell\times m}$. The secret key is $\mathbf{S} \in \mathbb{Z}_q^{\ell\times n}$.
- The prover samples $\mathbf{t} \leftarrow [-B, B]^{1\times n}$, $\mathbf{e} \leftarrow \chi^{1\times m}$, and sends $\boldsymbol{\alpha} = \mathbf{tA} + \mathbf{e} \in \mathbb{Z}_q^{1\times m}$ to the verifier.
- The verifier sends a challenge $\mathbf{c} \leftarrow \{0,1\}^{1\times \ell}$ as the second message.
- The prover computes $\mathbf{z} = \mathbf{t} + \mathbf{cS} \in \mathbb{Z}_q^{1\times n}$, and sends it to the verifier.
- The verifier accepts if $\|\mathbf{zA} - \boldsymbol{\alpha} - \mathbf{cY}\|_\infty \leq (\ell + 1)\beta$ and $\|\mathbf{z}\|_\infty \leq B + \ell\beta$.

Claim 5.16 (Completeness)**.** *The identification protocol Π^{LWE} is complete.*

Proof. We have $\mathbf{zA} - \boldsymbol{\alpha} - \mathbf{cY} = -\mathbf{e} - \mathbf{cE}$, where $\|\mathbf{e}\|_\infty \leq B$ and $\|\mathbf{E}\|_\infty \leq \beta$, and therefore $\|\mathbf{zA} - \boldsymbol{\alpha} - \mathbf{cY}\|_\infty \leq B + \ell\beta$. Similarly, $\|\mathbf{t}\|_\infty \leq B$ and $\|\mathbf{S}\|_\infty \leq \beta$, so that $\|\mathbf{z}\|_\infty \leq B + \ell\beta$. □

Next, we show that there are no (even inefficient) cheating strategies succeeding over random instances $(\mathbf{A}, \mathbf{Y})$.

Claim 5.17 (Average-Case Soundness)**.** *Suppose that $m \geq 2n \log q$ and $B + \ell\beta \leq q/2$.*

Then identification protocol Π^{LWE} is average-case statistically sound. Namely, for all (potentially inefficient) cheating provers P^*:

$$\Pr_{\mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m}, \mathbf{Y}\leftarrow\mathbb{Z}_q^{\ell\times m}} [(P^*(\mathbf{A}, \mathbf{Y}) \leftrightarrow V(\mathbf{A}, \mathbf{Y})) = \mathrm{Accept}] \leq \mathrm{negl}(n),$$

where $P^*(\mathbf{A}, \mathbf{Y}) \leftrightarrow V(\mathbf{A}, \mathbf{Y})$ denotes the output of the verifier after interacting with P^*.

Proof. By the leftover hash lemma, the distribution $(\mathbf{Y}, \mathbf{cY})$ is statistically close to $(\mathbf{Y}, \mathbf{U})$ where $\mathbf{U} \leftarrow \mathbb{Z}_q^{1\times m}$. Let $\boldsymbol{\alpha}^* = \boldsymbol{\alpha}^*(\mathbf{A}, \mathbf{Y})$ be the first message sent by P^*. Then $(\mathbf{A}, \mathbf{Y}, \boldsymbol{\alpha}^*, \mathbf{cY})$ is statistically close to $(\mathbf{A}, \mathbf{Y}, \boldsymbol{\alpha}^*, \mathbf{U})$.

Fix $\boldsymbol{\alpha}^* \in \mathbb{Z}_q^{1\times m}$. For a fixed $\mathbf{z} \in \mathbb{Z}_q^{1\times m}$, the probability over $\mathbf{u} \leftarrow \mathbb{Z}_q^{1\times m}$ that $\|\mathbf{zA} - \boldsymbol{\alpha}^* - \mathbf{u}\|_\infty \leq B + \ell\beta$ is at most $((\ell + 1)\beta)^m/q^m$.

By union bound over $\mathbf{z}$, the probability over $\mathbf{u} \leftarrow \mathbb{Z}_q^{1\times m}$ that there exists $\mathbf{z} \in \mathbb{Z}_q^{1\times n}$ such that $\|\mathbf{zA} - \boldsymbol{\alpha}^* - \mathbf{u}\|_\infty \leq B + \ell\beta$ is at most

$$q^n \cdot (B + \ell\beta)^m/q^m \leq q^n/2^m \leq q^{-n}$$

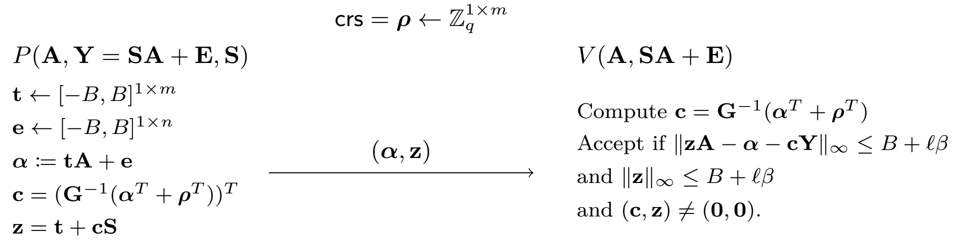
which is negligible, so that with overwhelming probability no prover message in step 3 can make the verifier accept. □

Claim 5.18 (Honest-Verifier Zero-Knowledge). *Suppose $(\ell\beta)/B \leq \mathrm{negl}(\lambda)$. The identification protocol Π^{LWE} is statistically honest-verifier zero-knowledge.*

Proof. We define our honest-verifier simulator $\mathcal{S}$ as follows. On input $(\mathbf{A}, \mathbf{Y}, \mathbf{c})$, it samples $\mathbf{z} \leftarrow [-B, B]^{1\times n}$. It samples $\mathbf{e} \leftarrow [-B, B]^{1\times m}$, sets $\boldsymbol{\alpha} = \mathbf{zA} - \mathbf{cY} + \mathbf{e}$, $\mathbf{c} = \mathbf{G}^{-1}(\boldsymbol{\alpha})$, and outputs $(\boldsymbol{\alpha}, \mathbf{c}, \mathbf{z})$.

By Lemma 5.1, the distribution of $\mathbf{z}$ is statistically close to the one produced by real proofs. Then, for accepting proofs, $\boldsymbol{\alpha}$ is distributed as $\boldsymbol{\alpha} = \mathbf{tA} + \mathbf{e} = \mathbf{zA} - \mathbf{cY} + \mathbf{cE} + \mathbf{e}$ where $\mathbf{Y} = \mathbf{SA} + \mathbf{E}$, $\mathbf{E} \leftarrow \chi^{\ell\times m}$, and $\mathbf{e} \leftarrow [-B, B]^{1\times m}$. But by Lemma 5.1, for all $\mathbf{c} \in \{0, 1\}^{1\times\ell}$, this distribution is statistically close to $\boldsymbol{\alpha} = \mathbf{zA} - \mathbf{cY} + \mathbf{e}$ where $\mathbf{e} \leftarrow [-B, B]^{1\times m}$: this is the distribution output by the simulator $\mathcal{S}$. □

Next, we show that instantiation the Fiat-Shamir heuristic on Π^{LWE} with the hash function $\mathbf{G}^{-1}(\cdot)$ preserves (average-case) soundness. As for the SIS version, we additionally rely on a common random string to argue zero-knowledge.



$\mathsf{crs} = \boldsymbol{\rho} \leftarrow \mathbb{Z}_q^{1\times m}$

$P(\mathbf{A}, \mathbf{Y} = \mathbf{SA} + \mathbf{E}, \mathbf{S})$

$\mathbf{t} \leftarrow [-B, B]^{1\times m}$

$\mathbf{e} \leftarrow [-B, B]^{1\times n}$

$\boldsymbol{\alpha} := \mathbf{tA} + \mathbf{e}$

$\mathbf{c} = (\mathbf{G}^{-1}(\boldsymbol{\alpha}^T + \boldsymbol{\rho}^T))^T$

$\mathbf{z} = \mathbf{t} + \mathbf{cS}$

$\xrightarrow{(\boldsymbol{\alpha}, \mathbf{z})}$

$V(\mathbf{A}, \mathbf{SA} + \mathbf{E})$

Compute $\mathbf{c} = \mathbf{G}^{-1}(\boldsymbol{\alpha}^T + \boldsymbol{\rho}^T)$

Accept if $\|\mathbf{zA} - \boldsymbol{\alpha} - \mathbf{cY}\|_\infty \leq B + \ell\beta$

and $\|\mathbf{z}\|_\infty \leq B + \ell\beta$

and $(\mathbf{c}, \mathbf{z}) \neq (\mathbf{0}, \mathbf{0})$.

Figure 6: Non-interactive Identification Protocol $(\Pi^{\mathrm{LWE}})_{\mathrm{FS},\mathbf{G}^{-1}}$ based on LWE.

Notice that now $\boldsymbol{\alpha}, \boldsymbol{\rho} \in \mathbb{Z}_q^{1\times m}$ are row vectors. Therefore, $(\mathbf{G}^{-1}(\boldsymbol{\alpha}^T + \boldsymbol{\rho}^T)) \in \mathbb{Z}_q^\ell$ is a column vector and $\mathbf{c} = (\mathbf{G}^{-1}(\boldsymbol{\alpha}^T + \boldsymbol{\rho}^T))^T \in \mathbb{Z}_q^{1\times\ell}$ is in turn a row vector. In other words, in our syntax, $\mathbf{G}^{-1}(\cdot)$ expands column vectors to column vectors (instead of row vectors to row vectors), which introduces the transposes in the hash function $\mathbf{G}^{-1}(\cdot)$.

Claim 5.19 (Completeness). *The protocol $(\Pi^{\mathrm{LWE}})_{\mathrm{FS},\mathbf{G}^{-1}}$ is complete.*

Proof. This follows by completeness of the interactive variant Π^{LWE}. □

Claim 5.20 (Average-case soundness). *Under the $iSIS_{m,\,n+\ell+m,\,q,\,B+\ell\beta}$ assumption, we have that for all efficient cheating prover P^*:*

$$\Pr_{\mathsf{crs}\leftarrow\mathbb{Z}_q^{1\times m}, \mathbf{A}\leftarrow\mathbb{Z}_q^{n\times m}, \mathbf{Y}\leftarrow\mathbb{Z}_q^{\ell\times m}}[(P^*(\mathsf{crs}, \mathbf{A}, \mathbf{Y}) \leftrightarrow V(\mathsf{crs}, \mathbf{A}, \mathbf{Y})) = \mathrm{Accept}] \leq \mathrm{negl}(n).$$

In particular, $(\Pi^{\mathrm{LWE}})_{\mathrm{FS},\mathbf{G}^{-1}}$ is a one-time secure identification scheme.

Proof. Accepting proofs $(\boldsymbol{\alpha}, \mathbf{c}, \mathbf{z})$ for $(\Pi^{\mathrm{LWE}})_{\mathrm{FS},\mathbf{G}^{-1}}$ satisfy $\|\mathbf{z}\mathbf{A} - \boldsymbol{\alpha} - \mathbf{c}\mathbf{Y}\|_\infty \leq B + \ell\beta$ where $\|\mathbf{z}\|_\infty \leq B + \ell$. This can be rewritten as

$$[\mathbf{z} \,\|\, -\mathbf{c} \,\|\, \mathbf{z}\mathbf{A} - \boldsymbol{\alpha} - \mathbf{c}\mathbf{Y}] \begin{bmatrix} \mathbf{A} \\ \mathbf{Y} + \mathbf{G}^T \\ -\mathbf{I} \end{bmatrix} = \boldsymbol{\rho},$$

where $\mathbf{c} = (\mathbf{G}^{-1}(\boldsymbol{\alpha}^T + \boldsymbol{\rho}^T))^T$.

Let $P^*(\mathbf{A}, \mathbf{Y})$ be a cheating prover breaking average-case soundness of Π^{LWE} over the randomness of $(\mathsf{crs}, \mathbf{A}, \mathbf{Y}) \leftarrow \mathbb{Z}_q^{1\times m} \times \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{\ell\times m}$. Let $(\mathbf{B}, \boldsymbol{\tau}^T)$ where $\mathbf{B} = [\mathbf{B}_1 \| \mathbf{B}_2 \| \mathbf{B}_3] \leftarrow \mathbb{Z}_q^{m\times(n+\ell+m)}$ and $\boldsymbol{\tau} \leftarrow \mathbb{Z}_q^{1\times m}$ be an inhomogeneous SIS instance.

We define a reduction as follows. If $\mathbf{B}_3$ is not invertible mod q, the reduction aborts. Otherwise, it computes $\mathbf{C} = -\mathbf{B}_3^{-1}\mathbf{B} = [\mathbf{C}_1 \| \mathbf{C}_2 \| -\mathbf{I}]$ where $\mathbf{C}_1 \in \mathbb{Z}_q^{m\times n}, \mathbf{C}_2 \in \mathbb{Z}_q^{m\times \ell}$, and computes $\boldsymbol{\rho}^T = -\mathbf{B}_3^{-1}\boldsymbol{\tau}^T$ so that $-\mathbf{B}_3\boldsymbol{\rho}^T = \boldsymbol{\tau}^T$.

Then, any accepting transcript $(\boldsymbol{\alpha}, \mathbf{c}, \mathbf{z})$ produced by P^* on input $(\boldsymbol{\rho}, \mathbf{C}_1^T, (\mathbf{C}_2 - \mathbf{G}^T)^T)$ (which is distributed uniformly as $\mathbf{B}_3^{-1}$ is invertible) gives $\mathbf{r} = \begin{bmatrix} \mathbf{z}^T \\ -\mathbf{c}^T \\ \left(\mathbf{z}\mathbf{A} - \boldsymbol{\alpha} - \mathbf{c}\mathbf{Y}\right)^T \end{bmatrix} \in \mathbb{Z}_q^{n+\ell+m}$ which is non-zero, as $(\mathbf{z}, \mathbf{c}) \neq (\mathbf{0}, \mathbf{0})$ such that $\|\mathbf{r}\|_\infty \leq B + \ell\beta$. Furthermore, we have $\mathbf{C}\mathbf{r} = [\mathbf{C}_1 \| \mathbf{C}_2 \| -\mathbf{I}] \cdot \mathbf{r} = \boldsymbol{\rho}^T$ so that $\mathbf{B}\mathbf{r} = -\mathbf{B}_3\mathbf{C}\mathbf{r} = -\mathbf{B}_3\boldsymbol{\rho}^T = \boldsymbol{\tau}^T$, and therefore $\mathbf{r}$ is an inhomogeneous SIS solution for $(\mathbf{B}, \boldsymbol{\tau}^T)$. □

As for the protocols based on SIS, the randomness of the CRS $\boldsymbol{\rho}$ is only used for zero-knowledge, and could be set to $\mathbf{0}$ if we only cared about soundness. So as for the SIS case, using the *single, deterministic* Fiat-Shamir hash function $\mathbf{G}^{-1}(\cdot)$ preserves soundness of Π^{LWE}; it is only for zero-knowledge that we consider a (slightly modified) *family of* hash functions $(\mathbf{G}_{\boldsymbol{\rho}}^{-1})^T(\boldsymbol{\alpha}) = (\mathbf{G}^{-1}(\boldsymbol{\alpha}^T + \boldsymbol{\rho}^T))^T$. Note that, as in the SIS version, we directly argue zero-knowledge of the non-interactive protocol instead of relying on the honest-verifier zero-knowledge property of the interactive version.

Claim 5.21 (Zero-Knowledge)**.** *Suppose $\ell\beta/B \leq \mathrm{negl}(n)$, let δ be the uniform distribution over $[-B, B]^m$. Under the $LWE_{n,m,q,\delta}$ assumption,*[6][7] $(\Pi^{\mathrm{LWE}})_{\mathrm{FS},\mathbf{G}^{-1}}$ *is (single-theorem) computationally zero-knowledge.*

We define our simulator $\mathbf{S}$ as follows. On input $(\mathbf{A}, \mathbf{Y})$, it samples $\mathbf{u} \leftarrow \mathbb{Z}_q^m$, and sets $\mathbf{c} = (\mathbf{G}^{-1}(\mathbf{u}))^T$. It samples $\mathbf{z}$ and $\mathbf{e}$ uniformly from $[-B, B]^m$, and sets $\boldsymbol{\rho} = [\mathbf{z} \| -\mathbf{c} \| \mathbf{e}] \begin{bmatrix} \mathbf{A} \\ \mathbf{Y} + \mathbf{G}^T \\ -\mathbf{I} \end{bmatrix}$. It sets $\boldsymbol{\alpha} = \mathbf{u}^T - \boldsymbol{\rho}$, and outputs $(\mathsf{crs} = \boldsymbol{\rho}, (\boldsymbol{\alpha}, \mathbf{c}, \mathbf{z}))$.

First, $\mathbf{u}$ is statistically close to uniform over $\mathbb{Z}_q^m$ over the randomness of $\boldsymbol{\rho}$ alone. Then, by Lemma 5.1, $\mathbf{z}$ is distributed statistically close to $\mathbf{z} + \mathbf{c}\mathbf{S}$ even conditioned on $\mathbf{c}$ and $\mathbf{u}$. Similarly, $\mathbf{e}$ is distributed statistically close to $\mathbf{z}\mathbf{A} - \boldsymbol{\alpha} - \mathbf{c}\mathbf{Y} = \mathbf{e} + \mathbf{c}\mathbf{E}$ even conditioned on $\mathbf{c}$ and $\mathbf{u}$. Now $\boldsymbol{\rho}$ is entirely determined as

$$\boldsymbol{\rho} = [\mathbf{z} \,\|\, -\mathbf{c} \,\|\, \mathbf{z}\mathbf{A} - \boldsymbol{\alpha} - \mathbf{c}\mathbf{Y}] \begin{bmatrix} \mathbf{A} \\ \mathbf{Y} + \mathbf{G}^T \\ -\mathbf{I} \end{bmatrix}.$$

By the $LWE_{n,m,q,\chi}$ assumption, $\boldsymbol{\rho}$ is computationally indistinguishable from uniform. This in turn determines $\boldsymbol{\alpha} = \mathbf{u}^T - \boldsymbol{\rho}$, and therefore the distribution output by $\mathcal{S}$ is computationally indistinguishable to honestly generated proofs.

Parameters. To argue security of Π^{LWE} and $(\Pi^{\mathrm{LWE}})_{\mathrm{FS},\mathbf{G}^{-1}}$, we used the following properties:

- $\mathbf{G} \in \mathbb{Z}_q^{m\times\ell}$ is a gadget matrix. It suffices to set $\ell = m\lceil\log q\rceil$ to satisfy this property when instantiating $\mathbf{G}$ as the "powers-of-two" matrix. We stress that we could technically use any gadget matrix satisfying the requirements of Definition 5.9, albeit with slightly different parameters.

[6] Recall that this refers to the HNF form of LWE, where the secret is also taken from the distribution δ.

[7] This assumption is in particular implied by $LWE_{n,m,q,\chi}$ for any β-bounded distribution χ such that $\beta/B \leq \mathrm{negl}(n)$.

- $B + \ell\beta \leq q/2$ and $m \geq 2n \log\ q$ to argue average-case soundness of Π^{LWE};
- $\ell\beta/B \leq \mathrm{negl}(n)$ to argue zero-knowledge in Claims 5.18 and 5.21;
- $LWE_{n,m,q,[-B,B]^m}$ holds to argue zero-knowledge of $(\Pi^{\mathrm{LWE}})_{\mathrm{FS},\mathbf{G}^{-1}}$ in Claim 5.21. Note that this holds assuming $LWE_{n,m,q,\chi}$ for any β-bounded distribution χ such that $\beta/B \leq \mathrm{negl}(n)$;
- $SIS_{n,m+\ell,q,B+\ell}$ holds, to argue soundness of $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$ in Claim 5.14;
- $LWE_{n,m,q,\chi}$ to argue that the base language is hard.

Overall, we can set $m = 2n\lceil\log q\rceil$, $\ell = m\lceil\log q\rceil$, $q = 2^{n^\epsilon}$ for any $0 < \epsilon < 1$, any $B = n^{\omega}(1) < q/4$, and χ a β-bounded distribution such that $\beta/B \leq \mathrm{negl}(n)$.

Then our scheme is secure assuming both the $LWE_{n,m,q,\chi}$ and $SIS_{n,m+\ell,q,B+\ell}$ assumptions (and where statistical zero-knowledge holds with statistical distance $\approx \ell\beta/B + q^{-n/2}$), and is therefore under the (quantum) hardness of $GapCVP$ and $SIVP$ with sub-exponential approximation factors.

Attacks on Worst-Case Soundness. Our claims for soundness for Π^{LWE} (Claim 5.17) and $(\Pi^{\mathrm{LWE}})_{\mathrm{FS},\mathbf{G}^{-1}}$ (Claim 5.20) hold over *random* instances. One can naturally ask if they satisfy a standard notion of *worst-case* soundness, which require that no cheating prover should convince a verifier on *any* false instance. Here, by false instance, we mean any instance $(\mathbf{A}, \mathbf{Y})$ such that there does not exist $\mathbf{E}$ (nor $\mathbf{S}$) with norm at most β such that $\mathbf{Y} = \mathbf{SA} + \mathbf{E}$.

We show here that they do not satisfy worst-case soundness as is, by showing an attack on particular instances Π^{LWE} that breaks soundness with probability 1/2, and a full attack on particular instances of $(\Pi^{\mathrm{LWE}})_{\mathrm{FS},\mathbf{G}^{-1}}$.

Pick $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ uniformly at random. Suppose the first $\ell - 1$ rows of $\mathbf{Y}$ are LWE samples, that is, are set as $\mathbf{Y}_i = \mathbf{S}_i\mathbf{A} + \mathbf{E}_i$ for some short $\mathbf{S}_i, \mathbf{E}_i$, and suppose the last row of $\mathbf{Y}$ is $(0, \ldots, 0, q/2)$. Then, with high probability over the randomness of $\mathbf{A}$, $(0, \ldots, 0, q/2)$ cannot be written as $\mathbf{sA} + \mathbf{e}$ for any short $\mathbf{e}$, and therefore defines a false instance.[8]

Then, if $\mathbf{c}$ is set such that $\mathbf{c}_\ell = 0$, then a cheating prover can convince the verifier using his knowledge of $\mathbf{S}_i$, by running the honest prover (which would not use $\mathbf{S}_\ell$ in that case). This gives a cheating prover strategy with success probability (negligibly close to) 1/2.

The same strategy also applies for $(\Pi^{\mathrm{LWE}})_{\mathrm{FS},\mathbf{G}^{-1}}$, but now the cheating prover can sample $\boldsymbol{\alpha} \neq 0$ honestly until the last coordinate of $\mathbf{G}^{-1}(\boldsymbol{\alpha})$ is zero, in which case he succeeds with probability close to 1 (notice that under the LWE assumption, $\mathbf{G}^{-1}(\boldsymbol{\alpha})$ is distributed computationally close to $\mathbf{G}^{-1}(\mathbf{u})$ where $\mathbf{u} \leftarrow \mathbb{Z}_q^{1\times m}$).

5.4 More Efficient Protocols via Rejection Sampling

One drawback of the previous identification protocols is that zero-knowledge is argued using noise flooding. This requires the modulus q to be super-polynomially larger than the secret (namely, $\mathbf{R}$ in the SIS versions and $\mathbf{S}$ in the LWE ones), and in particular q has to be super-polynomial. This leads to quite inefficient schemes in practice.

Here, we describe variants of Π^{SIS} (Fig. 3) and $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$ (Fig. 4) that are compatible with a polynomial modulus q, using the rejection sampling technique of [Lyu09, Lyu12, LW15]. In a nutshell, instead of flooding the dependence of the response $\mathbf{z}$ in the secret, the prover now uses a much smaller masking term, but aborts the protocol with some probability. This will ensure that the distribution of the resulting response is independent of his secret.

Unfortunately, this results in downgrading security from zero-knowledge to witness indistinguishability: this is essentially because sampling from this secret-independent distribution is hard without any secrets. While this is meaningful in the SIS regime, it is vacuous for LWE languages as the witness there is unique (with overwhelming probability over $\mathbf{A}$). We therefore focus on the SIS variants in this section.

[8] This can be seen by a union bound on the q^n balls centered on points $\mathbf{zA}$, *e.g.* [GPV08, Lemma 5.3].

Our interactive identification scheme only features weak properties: the prover has some chance of aborting the execution of the protocol, compromising both completeness and witness-indistinguishability. Instead, we obtain our non-interactive variant using the *Fiat-Shamir with aborts* technique of [Lyu09], where the prover only sends a complete execution over to the verifier.

Interestingly this unveils a connection between lattice trapdoors and identification schemes. Indeed, the transcript of our non-interactive protocol $(\Pi^{\mathsf{SIS-Rej}})_{\mathsf{FS},\mathbf{G}^{-1}}$ (Fig. 8) exactly matches the output of the trapdoor presampling algorithm of [LW15] where the target is $\mathbf{0}$. We develop on that connection at the end of the section.

Let n, m, q, and ℓ, B be integers. Let $P_{\mathbf{t}}$ and $P_{\mathbf{z}}$ be two probability distributions over $\mathbb{Z}_q^m$, and let $M > 0$ be a real. We first present an interactive identification protocol based on rejection sampling.

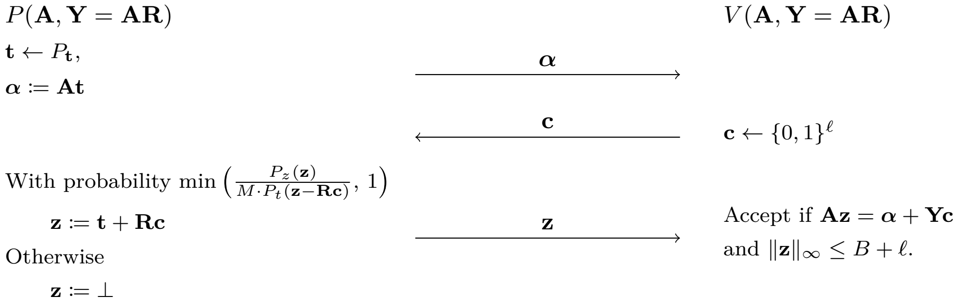

$P(\mathbf{A}, \mathbf{Y} = \mathbf{AR})$
$V(\mathbf{A}, \mathbf{Y} = \mathbf{AR})$
$\mathbf{t} \leftarrow P_{\mathbf{t}}$,
$\boldsymbol{\alpha} := \mathbf{At}$
$\boldsymbol{\alpha}$
$\mathbf{c}$
$\mathbf{c} \leftarrow \{0,1\}^{\ell}$
With probability $\min\left(\frac{P_z(\mathbf{z})}{M \cdot P_t(\mathbf{z}-\mathbf{Rc})}, 1\right)$
$\mathbf{z} := \mathbf{t} + \mathbf{Rc}$
$\mathbf{z}$
Accept if $\mathbf{Az} = \boldsymbol{\alpha} + \mathbf{Yc}$
and $\|\mathbf{z}\|_\infty \leq B + \ell$.
Otherwise
$\mathbf{z} := \perp$

Figure 7: Identification Protocol $\Pi^{\mathsf{SIS-Rej}}$ based on SIS.

Completeness holds whenever the prover sends $\mathbf{z} \neq \perp$, and relaxed special soundness follows from a proof nearly identical to Claim 5.11.

The advantages of using rejection sampling comes at the cost of downgrading zero-knowledge to witness-indistinguishability. The proof of the following claim is essentially in [Lyu09, Section 3.1] for the following distributions.

Claim 5.22 (Witness Indistinguishability)**.** *Suppose $P_{\mathbf{t}}$ is a m-dimensional discrete gaussian with parameter σ, and let $I = [-(mn\sigma - \ell), mn\sigma - \ell]^m$. Set $M = 1/P_{\mathbf{t}}(I)$ and define $P_{\mathbf{z}}$ as $P_{\mathbf{z}}(\mathbf{z}) = P_{\mathbf{t}}(\mathbf{z})$ if $\mathbf{z} \in I$ and* 0 *otherwise. Then, conditioned on $\mathbf{z}$ being sent in the third round, $\Pi^{\mathsf{SIS-Rej}}$ is witness-indistinguishable.*

The protocol $\Pi^{\mathsf{SIS-Rej}}$ has quite a few drawbacks: it only achieves weak completeness and weak witness-indistinguishability. A natural idea to boost completeness would be to repeat the protocol until some $\mathbf{z} \neq \perp$ is sent. However, this is in general breaks witness indistinguishability: even though the third message $\mathbf{z} = \mathbf{t} + \mathbf{Rc}$ itself does not reveal which secret $\mathbf{R}$ is used, the *probability* of sending $\mathbf{z} \neq \perp$ *does* depend $\mathbf{R}$. In other words, seeing aborted transcripts could break security.

The key idea, introduced by [Lyu09], consists in applying the Fiat-Shamir heuristic regardless of the weak properties of the base protocol. Now, for the resulting non-interactive protocol, the prover can keep producing transcripts in his head until some outputs some $\mathbf{z} \neq \perp$: this allows us to obtain (statistical) completeness. Furthermore, the fact the prover only sends the one accepting transcript allows us to argue witness indistinguishability. The resulting protocol is therefore not directly the result of the Fiat-Shamir heuristic itself, but of a *Fiat-Shamir with aborts.*

We now define our protocol $(\Pi^{\mathsf{SIS-Rej}})_{\mathsf{FS},\mathbf{G}^{-1}}$ in Fig. 8.

Completeness and average-case soundness for $(\Pi^{\mathsf{SIS-Rej}})_{\mathsf{FS},\mathbf{G}^{-1}}$ follow from arguments nearly identical to the ones of Section 5.2, where we implicitly set $\boldsymbol{\rho} = \mathbf{0}$: we do not need any common random string as the rejection sampling will ensure witness indistinguishability.

Claim 5.23 (Completeness)**.** *Suppose $P_{\mathbf{t}}$ is a B-bounded distribution for some B. Then the expected running time of the prover is at most $2M$, and the protocol $(\Pi^{\mathsf{SIS-Rej}})_{\mathsf{FS},\mathbf{G}^{-1}}$ is complete.*

$P(\mathbf{A}, \mathbf{Y} = \mathbf{AR})$ $\qquad\qquad\qquad\qquad$ $V(\mathbf{A}, \mathbf{Y} = \mathbf{AR})$

$\mathbf{t} \leftarrow P_{\mathbf{t}}$,

$\boldsymbol{\alpha} := \mathbf{At}$

$\mathbf{c} = \mathbf{G}^{-1}(\boldsymbol{\alpha})$

With probability $\min\left(\frac{P_z(\mathbf{z})}{M \cdot P_t(\mathbf{z}-\mathbf{Rc})} 1\right)$: $\xrightarrow{(\boldsymbol{\alpha}, \mathbf{z})}$ Compute $\mathbf{c} = \mathbf{G}^{-1}(\boldsymbol{\alpha})$
Accept if $\mathbf{Az} = \boldsymbol{\alpha} + \mathbf{Yc}$
and $\|\mathbf{z}\|_\infty \leq B + \ell$.

$\quad \mathbf{z} := \mathbf{t} + \mathbf{Rc}$

Otherwise Repeat from the beginning.

Figure 8: Identification Protocol $(\Pi^{\mathrm{SIS-Rej}})_{\mathrm{FS},\mathbf{G}^{-1}}$ based on SIS.

Claim 5.24 (Average-case soundness)**.** *Suppose the distribution $P_{\mathbf{z}}$ is B-bounded for some B. Then, under the $SIS_{n,m+\ell,q,B+\ell}$ assumption, we have that for all efficient cheating prover P^* for $(\Pi^{\mathrm{SIS-Rej}})_{\mathrm{FS},\mathbf{G}^{-1}}$:*

$$\Pr_{\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}, \mathbf{Y} \leftarrow \mathbb{Z}_q^{n \times \ell}} [(P^*(\mathbf{A}, \mathbf{Y}) \leftrightarrow V(\mathbf{A}, \mathbf{Y})) = \mathrm{Accept}] \leq \mathrm{negl}(n).$$

In particular, $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$ is a one-time secure identification scheme.

It remains to argue witness indistinguishability. This proof of the following claim is identical to the one of [LW15, Section 3].

Claim 5.25 (Witness-Indistinguishability)**.** *Suppose that the distributions $\mathbf{At}$ and $\mathbf{Az}$ are statistically close to uniform* mod q*, where $\mathbf{t} \leftarrow P_{\mathbf{t}}$ and $\mathbf{z} \leftarrow P_{\mathbf{z}}$.*[9]

Suppose furthermore that, over the randomness of $\boldsymbol{\alpha} \leftarrow \mathbb{Z}_q^n$, $\mathbf{c} = \mathbf{G}^{-1}(\boldsymbol{\alpha})$, and $\mathbf{z} \leftarrow P_{\mathbf{z}}$ conditioned on $\mathbf{Az} = \mathbf{u} + \mathbf{Yc}$:

$$\Pr\left[\frac{P_{\mathbf{z}}(\mathbf{z})}{P_{\mathbf{t}}(\mathbf{z} - \mathbf{Rc})} \leq M\right] \geq 1 - \mathrm{negl}(n).$$

Then the protocol $(\Pi^{\mathrm{SIS-Rej}})_{\mathrm{FS},\mathbf{G}^{-1}}$ is statistically witness-indistinguishable.

Proof. Consider the following (inefficient) simulator $\mathcal{S}$. It first generates $\boldsymbol{\alpha} \leftarrow \mathbb{Z}_q^n$, and sets $\mathbf{c} = \mathbf{G}^{-1}(\boldsymbol{\alpha})$. It then samples $\mathbf{z} \leftarrow P_{\mathbf{z}}$ conditioned on $\mathbf{Az} = \mathbf{u} + \mathbf{Yc}$. Notice that this last step is inefficient. Finally, the simulator outputs $(\boldsymbol{\alpha}, \mathbf{c}, \mathbf{z})$.

The proof of [LW15, Theoreom 3.1] exactly shows that the resulting distribution is statistically indistinguishable from a honestly generated transcript (setting their target $\mathbf{t}$ as $\mathbf{0}$). □

Instantiations and parameters. As in Section 5.2, we can use the "powers-of-two" gadget matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times \ell}$, which sets $\ell = n\lceil \log q \rceil$.. We stress that we could technically use any gadget matrix satisfying the requirements of Definition 5.9, albeit with slightly different parameters.

To ensure the first hypothesis of Claim 5.25, we can set $m = n\lceil \log q \rceil$ as well (and require that the distributions $P_{\mathbf{t}}$ and $P_{\mathbf{z}}$ have enough min-entropy to apply the leftover hash lemma).

The main parameters left to instantiate are the distributions $P_{\mathbf{t}}$ and $P_{\mathbf{z}}$. [LW15] proposes two instantiations that can directly be used to instantiate our theorems.

One is to set $P_{\mathbf{t}}$ to be the uniform distribution over the cube $[-(m+1)\ell, (m+1)\ell)]^m$, and $P_{\mathbf{z}}$ as the uniform distribution over $[-m\ell, m\ell]^m$. This sets $M \approx e$, and therefore the prover will run the loop $1/M = 1/e$ times in expectation. This leads to a very simple "rejection sampling" step: the prover sends $\mathbf{z} \neq \perp$ if and only if $\|bz\|_\infty \leq m\ell$. This makes the proof rely on the hardness of $SIS_{n,m+\ell,q,(m+1)\ell}$.

Another possible choice is to set $P_{\mathbf{t}}$ and $P_{\mathbf{z}}$ as discrete gaussians over with the same parameter $\sigma = \Theta(\ell\sqrt{\lambda})$, where λ denotes the security parameter. One can set $M = e^{1+1/\lambda}$, which makes the prover run the loop < 3.5 times in expectation. This makes the proof rely on the hardness of $SIS_{n,m+\ell,q,\Theta(\ell\sqrt{\lambda})}$.

In all cases security is implied by the (quantum) hardness of $GapCVP$ and $SIVP$ with sub-exponential approximation factors.

[9] By the leftover hash lemma, this holds if $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times m}$ and $P_{\mathbf{t}}$, $P_{\mathbf{z}}$ have sufficiently high min-entropy.

Connection with the Trapdoor Preimage Sampling Algorithm of [LW15]. It turns out that the transcript of the protocol $(\Pi^{\mathrm{SIS-Rej}})_{\mathrm{FS},\mathbf{G}^{-1}}$ (Fig. 8) matches the output of the trapdoor preimage sampling algorithm of [LW15]. In their context, the goal is to sample a short $\mathbf{r} \in \mathbb{Z}_q^{m+\ell}$ such that

$$[\mathbf{A} \parallel \mathbf{G} + \mathbf{AR}] \cdot \mathbf{r} = \mathbf{0},$$

such that the distribution of $\mathbf{r}$ is independent of the trapdoor $\mathbf{R}$. One could already do so using the techniques of [MP12]. The main observation of [LW15] is that there exists a much simpler preimage sampling algorithm which only uses rejection sampling, instead of relying both on discrete gaussian sampling and lattice convolution theorems. It turns out that their algorithm is exactly the prover algorithm in $(\Pi^{\mathrm{SIS-Rej}})_{\mathrm{FS},\mathbf{G}^{-1}}$, with the syntactical difference of outputting $\mathbf{r} = \begin{bmatrix} \mathbf{z} \\ -\mathbf{c} \end{bmatrix}$ instead of $(\boldsymbol{\alpha}, \mathbf{c}, \mathbf{z})$ (note that $\boldsymbol{\alpha}$ can be recovered as $\boldsymbol{\alpha} = \mathbf{Gc}$). In other words, the prover of $(\Pi^{\mathrm{SIS-Rej}})_{\mathrm{FS},\mathbf{G}^{-1}}$ samples a short preimage of $\mathbf{0}$ under $[\mathbf{A} \| \mathbf{G} + \mathbf{AR}]$: this is easy to do using his knowledge of $\mathbf{R}$, and witness indistinguishability ensures that the output distribution is independent of $\mathbf{R}$.

Using a common random string, in a similar way as $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$ (Fig. 4), namely, setting it as some arbitrary $\boldsymbol{\rho} \leftarrow \mathbb{Z}_q^n$ and setting the challenge as $\mathbf{c} = \mathbf{G}^{-1}(\boldsymbol{\alpha} + \boldsymbol{\rho})$, the output transcript now satisfies:

$$[\mathbf{A} \parallel \mathbf{G} + \mathbf{Y}] \begin{bmatrix} \mathbf{z} \\ -\mathbf{c} \end{bmatrix} = \boldsymbol{\rho}.$$

This exactly recovers the preimage sampling algorithm of [LW15] for abitrary targets $\boldsymbol{\rho}$. Notice that the randomness of $\boldsymbol{\rho}$ was used in $(\Pi^{\mathrm{SIS}})_{\mathrm{FS},\mathbf{G}^{-1}}$ to argue zero-knowledge. Here, rejection sampling allows to argue witness indistinguishability, and setting $\boldsymbol{\rho}$ to any arbitrary value does not affect any of the properties of $(\Pi^{\mathrm{SIS-Rej}})_{\mathrm{FS},\mathbf{G}^{-1}}$, while completeness ensures that the equation above holds for accepting transcripts.

6 Negative Results for Fiat-Shamir with Non-Cryptographic Hash Functions

In this section, we give evidence that in contrast to our positive results (Section 4, Section 5), Fiat-Shamir for certain protocols *necessarily requires* a cryptographic hash function. Our prototypical example of such an interactive protocol is Blum's protocol for graph Hamiltonicity [Blu86], but our results extend to a broad class of 3-message HVZK argument systems.

Our results have two different forms:

- We show that even if one is willing to use an oracle (such as a random oracle or a generic group oracle) to instantiate the 3-message protocol (such as Blum), there is an unconditional break of soundness in the resulting Fiat-Shamir protocol for any hash function h that *does not make use of the oracle*. This stands in contrast to our results in Section 4, where idealized (GGM) assumptions about 3-message protocols *did* suffice for the soundness of Fiat-Shamir with an oracle independent hash function.
- We describe a concrete security property (which we call "mix-and-match resistance" (Definition 6.8)) such that for any protocol Π in a large class $\mathcal{C}$, any hash function (family) $\mathcal{H}$ that instantiates Fiat-Shamir for Π must possess this security property. This result also holds relative to natural oracle distributions $\mathcal{O}$, which further establishes that the "mix-and-match resistance" property of $\mathcal{H}$ is not "borrowing hardness" from the protocol. This stands in contrast to our results in Section 5, where a simple and non-cryptographic hash function was provably sufficient to instantiate Fiat-Shamir in the standard model.

The two kinds of results are closely related. As described in the technical overview (Section 2), our attacks in the random oracle model (for example) make only a polynomial number of queries to the oracle but require solving some oracle-independent problem in unbounded time. Our concrete security property is then the claim that this oracle-independent problem cannot be solved in polynomial time.

Of course, for this methodology to work, we have to ensure that the oracle-independent problem above actually has an information-theoretic solution. We begin (Section 6.1) with a technical lemma that will guarantee such an information-theoretic solution. We then apply this lemma to prove impossibility results for instantiating Fiat-Shamir for the Blum protocol (Section 6.2) and then state and prove our two general negative results (Section 6.3 and Section 6.4).

6.1 Main Information-Theoretic Lemma

Let $A^{(1)}, \ldots, A^{(t)}$ be arbitrary $q \times w$ binary matrices, and let $f : \{0,1\}^{wt} \to \Sigma^t$ be an arbitrary function. Finally, let $y^{(1)}, \ldots, y^{(t)} \leftarrow \Sigma^q$ be i.i.d. uniformly random elements of Σ^q.

For any vector $v \in [q]^t$, fix the notation $A[v] = (A^{(1)}_{v_1}, \ldots, A^{(t)}_{v_t})$ and $y[v] = (y^{(1)}_{v_1}, \ldots, y^{(t)}_{v_t})$.

Lemma 6.1. *If $q \geq t|\Sigma|\lambda$, then with $1 - O\left(\frac{1}{\lambda t}\right)$ probability, there exists a vector $v \in [q]^t$ such that $f(A[v]) = y[v]$.*

Proof. For every vector v, define the random variable $X_v = \chi\Big(f(A[v]) = y[v]\Big)$ (i.e., the indicator variable for v being a solution to our problem). Define $X = \sum_v X_v$. We want to show that $X > 0$ with high probability.

To do this, we apply the second moment method (Chebyshev's inequality). We first compute

$$\begin{aligned}
\mathbf{E}\left[X\right] &= \sum_{v \in [q]^t} \mathbf{E}[X_v] \\
&= \sum_{v \in [q]^t} \Pr\Big[f(A[v]) = y[v]\Big] \\
&= \sum_{v \in [q]^t} |\Sigma|^{-t} = \left(\frac{q}{|\Sigma|}\right)^t.
\end{aligned}$$

The third equality holds because for any vector v, the random variable $y[v]$ is uniform over Σ^t. We next compute the second moment of X, as follows

$$\begin{aligned}
\mathbf{E}\left[X^2\right] &= \sum_{v,w \in [q]^t} \mathbf{E}[X_v X_w] \\
&= \sum_{d=0}^{t} \sum_{\substack{v,w \in [q]^t \\ \delta_{\mathrm{H}}(v,w) = d}} \mathbf{E}[X_v X_w],
\end{aligned}$$

where $\delta_{\mathrm{H}}(v, w)$ denotes Σ-Hamming distance (the number of symbols on which v and w disagree). We claim that for every v, w such that $\delta_{\mathrm{H}}(v, w) = d$, $\mathbf{E}[X_v X_w] \leq 2^{-t-d}$. This can be seen by the calculation

$$\begin{aligned}
\mathbf{E}\left[X_v X_w\right] &= \Pr\Big[f(A[v]) = y[v] \text{ AND } f(A[w]) = y[w]\Big] \\
&= |\Sigma|^{-t} \Pr\Big[f(A[w]) = y[w] \mid f(A[v]) = y[v]\Big] \\
&\leq |\Sigma|^{-t-d},
\end{aligned}$$

where the last inequality follows from the fact that $y[w]$ has $d \log |\Sigma|$-bits of min-entropy given $y[v]$.

Therefore, we complete the calculation

$$
\begin{aligned}
\mathbf{E}[X^2] &= \sum_{d=0}^{t} \sum_{\substack{v,w\in[q]^t\\ \delta_{\mathrm{H}}(v,w)=d}} \mathbf{E}[X_v X_w] \\
&\le \sum_{d=0}^{t} \sum_{\substack{v,w\in[q]^t\\ \delta_{\mathrm{H}}(v,w)=d}} |\Sigma|^{-t-d} \\
&\le \sum_{d=0}^{t} \binom{t}{d} q^{t+d} |\Sigma|^{-t-d} \\
&= \left(\frac{q}{|\Sigma|}\right)^{2t} \sum_{d=0}^{t} \binom{t}{d} \left(\frac{|\Sigma|}{q}\right)^{t-d} .
\end{aligned}
$$

Thus, we can bound

$$
\begin{aligned}
\mathrm{Var}[X] &= \mathbf{E}[X^2] - \mathbf{E}[X]^2 \\
&\le \left(\frac{q}{|\Sigma|}\right)^{2t} \sum_{d=0}^{t-1} \binom{t}{d} \left(\frac{|\Sigma|}{q}\right)^{t-d} \\
&= \left(\frac{q}{|\Sigma|}\right)^{2t-1} \left(1 + \frac{|\Sigma|}{q}\right)^{t-1} .
\end{aligned}
$$

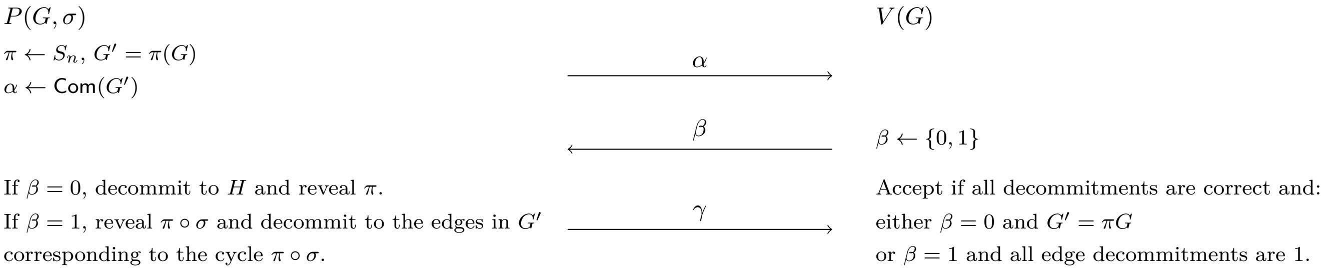
We conclude that when $q \ge t|\Sigma|\lambda$,

$$
\frac{\mathrm{Var}[X]}{\mathbf{E}[X]^2} = O\left(\frac{1}{\lambda t}\right),
$$

which implies that $\Pr[X > 0] \ge 1 - O\left(\frac{1}{\lambda t}\right)$ by Chebyshev's inequality. □

6.2 Negative Result for Blum in the Random Oracle Model

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 [Blu86] in which the commitment scheme is instantiated with a random oracle. 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.



$P(G, \sigma)$

$\pi \leftarrow S_n$, $G' = \pi(G)$

$\alpha \leftarrow \mathsf{Com}(G')$

$V(G)$

$\xrightarrow{\alpha}$

$\xleftarrow{\beta}$

$\beta \leftarrow \{0,1\}$

If $\beta = 0$, decommit to H and reveal π.
If $\beta = 1$, reveal $\pi \circ \sigma$ and decommit to the edges in G' corresponding to the cycle $\pi \circ \sigma$.

$\xrightarrow{\gamma}$

Accept if all decommitments are correct and:
either $\beta = 0$ and $G' = \pi G$
or $\beta = 1$ and all edge decommitments are 1.

Figure 9: The Zero Knowledge Proof System Π^{Blum} for Graph Hamiltonicity.

The Blum protocol Π is described in Fig. 9. For this example, we instantiate $\mathsf{Com}(b; r) = \mathcal{O}(x, r)$ as an idealized bitwise commitment scheme in the random oracle model. Π is repeated t times in parallel to obtain soundness error 2^{-t}. We now give a polynomial-query attack on $\Pi^t_{\mathrm{FS},H}$ for any hash function H that does not invoke the oracle[10] $\mathcal{O}$.

Let H denote a candidate Fiat-Shamir hash function for the above protocol Π_{Blum} when iterated t times in parallel. Consider the following attack on the Fiat-Shamir protocol $\Pi^t_{\mathrm{FS},H}$:

1. For $1 \le i \le t, 1 \le \ell \le q$, sample a random bit $y^{(i)}_\ell \leftarrow \{0,1\}$ and sample message $\alpha^{(i)}_\ell$: if $y^{(i)}_\ell = 0$, sample $\alpha^{(i)}_\ell$ as in the honest protocol, while if $y^{(i)}_\ell = 1$, and sample $\alpha^{(\ell)}_i$ as a commitment to a cycle graph.
2. Find $v \in [q]^t$ such that $H(\alpha[v]) = y[v]$.
3. Output $\alpha[v]$ as well as the necessary decommitments to $\alpha[v]$ (either the entire graph or just the edges in the cycle).

By Lemma 6.1, as long as $q = \omega(t)$, Step (2) has a solution with high probability over (α, y) (because the joint distribution (α, y) is statistically close to uniform). Therefore, this constitutes a poly-query attack on the protocol $\Pi^t_{\mathrm{FS},H}$ in the random oracle model. Moreover, if the computational problem in Step (2) (which does not depend on $\mathcal{O}$) can be solved *efficiently*, then there is a poly-*time* attack on $\Pi^t_{\mathrm{FS},H}$.

6.3 A General Polynomial-Query Attack

We now generalize our negative result for Π_{Blum} to a broader class of interactive arguments. Namely, we consider a class of 3-message public-coin honest-verifier zero-knowledge arguments relative to an arbitrary oracle (or efficiently simulatable oracle distribution[11]) $\mathcal{O}$ and give polynomial-query attacks on resulting Fiat-Shamir protocols for any (oracle-independent) hash function h.

Definition 6.2 (HVZK Arguments relative to an Oracle)**.** An interactive argument system $\Pi^{\mathcal{O}(\cdot)} = (P^{\mathcal{O}(\cdot)}, V^{\mathcal{O}(\cdot)})$ for a language L (with witness relation R_L) built relative to an oracle distribution $\mathcal{O}$ is an *HVZK argument system* relative to $\mathcal{O}$ if it satisfies the following properties:

- **Completeness**: For any $(x, w) \in R_L$, at the end of an interaction $\langle P^{\mathcal{O}(\cdot)}(x, w), V^{\mathcal{O}(\cdot)}(x)\rangle$, the verifier outputs 1 with probability $1 - \mathrm{negl}(\lambda)$.
- **Soundness error** ϵ: For any $x \notin L$ and any *efficient* $P^{*\mathcal{O}(\cdot)}$, $V^{\mathcal{O}(\cdot)}(x)$ (in an interaction with P^*) outputs 1 with probability at most ϵ.
- **Honest-Verifier Zero Knowledge**: There exists a polynomial-time simulator $\mathsf{Sim}^{\mathcal{O}(\cdot)}$ such that for every $(x, w) \in R_L$, the following indistinguishability holds:

$$\mathsf{Sim}^{\mathcal{O}(\cdot)}(x) \approx \mathsf{view}_V \langle P^{\mathcal{O}(\cdot)}(x, w), V^{\mathcal{O}(\cdot)}(x; r)\rangle.$$

  That is, the simulator outputs a verifier view that is indistinguishable from the honest verifier's view in an interaction with an honest prover.

We emphasize that the Simulator is only given query access to $\mathcal{O}$; it may not program the oracle.

Two Variants: We say that $\Pi^{\mathcal{O}}(\cdot)$ satisfies HVZK against **query-bounded** adversaries if simulation indistinguishability holds with respect to all polynomial-query distinguishers. We say that $\Pi^{\mathcal{O}(\cdot)}$ satisfies HVZK against **polynomial-time** adversaries if the indistinguishability holds with respect to all polynomial-time distinguishers.

For our negative results, we focus on protocols $\Pi^{\mathcal{O}(\cdot)}$ satisfying the following conditions

[10] Such an assumption is necessary, or else $\mathcal{O}$ can be used to instantiate a standard Random-Oracle based Fiat-Shamir hash function.

[11] An oracle distribution is efficiently simulatable if a polynomial-query interaction with $\mathcal{O}$ can be simulated (up to negligible statistical distance) in polynomial time. This captures models such as the random oracle model and generic group model.

- **Public Coin**: The verifier messages are assumed to be sampled publicly (no internal verifier state) and uniformly at random. This restriction is necessary for Fiat-Shamir to be well-defined syntactically.
- **3-Messages**: We assume that Π consists of only three rounds of interaction. This is mainly for simplicity of the analysis.
- **Small Challenge Space**: We assume that the verifier's message is an element of a polynomial-size alphabet Σ. Fiat-Shamir is then applied to the protocol Π^t repeated $t = \omega(1)$ times in parallel.

We note that for such protocols, the honest-verifier zero knowledge property is equivalent to **special honest-verifier zero knowledge**:

Definition 6.3 (Special Honest-Verifier Zero Knowledge)**.** A 3-message public-coin protocol $\Pi^{\mathcal{O}(\cdot)}$ is special honest-verifier zero knowledge if there exists a simulator $\mathsf{Sim}(x,\beta) \to (\alpha,\gamma)$ such that for all $(x,w) \in R_L$ and all verifier messages β, $\mathsf{Sim}(x,\beta)$ is (computationally/query-bounded) indistinguishable from the distribution $\{(\alpha, \mathsf{state}) \leftarrow P(x,w), \gamma \leftarrow P(\mathsf{state},\beta) : (\alpha,\gamma)\}$.

We now prove our two negative results on Fiat-Shamir using information-theoretic hash functions. For our first result, we generalize the polynomial-query attack on Blum. This attack requires one further property of the protocol Π: a variant of "zero-knowledge" that even holds for false statements:

Definition 6.4 (Challenge Hiding)**.** For a 3-message special honest-verifier zero knowledge protocol Π, we say that the SHVZK simulator Sim is *challenge hiding* if for all x (not necessarily true statements) and all challenges $\beta, \beta' \in \Sigma$, the following (computational/query-bounded) indistinguishability holds:

$$\Big\{(\alpha,\gamma) \leftarrow \mathsf{Sim}(x,\beta) : \alpha\Big\} \approx \Big\{(\alpha',\gamma') \leftarrow \mathsf{Sim}(x,\beta') : \alpha'\Big\}.$$

That is, simulated first messages hide their corresponding challenges.

The above definition is a worst-case notion, meaning that we require that the property holds for *every* false statement (and every true statement). We also consider an average-case variant:

Definition 6.5 (Average-Case Challenge Hiding)**.** Let Π denote a 3-message special honest-verifier zero knowledge protocol Π for a language L, and let $\mathcal{D}$ be a distribution on NO-instances. We say that the SHVZK simulator Sim is *challenge hiding on average* if the following two distributions are (computationally/query-bounded) indistinguishable:

$$\Big\{x \leftarrow \mathcal{D}, \beta \leftarrow \Sigma, (\alpha,\gamma) \leftarrow \mathsf{Sim}(x,\beta) : (x,\alpha,\beta)\Big\} \approx \Big\{x \leftarrow \mathcal{D}, \beta, \beta' \leftarrow \Sigma, (\alpha,\gamma) \leftarrow \mathsf{Sim}(x,\beta') : (x,\alpha,\beta)\Big\}.$$

Remark 6.6. As long as the oracle distribution $\mathcal{O}$ is efficiently simulatable (such as a random oracle or a GGM oracle), any Special HVZK protocol Π with simulator Sim is challenge-hiding against polynomial-time adversaries *for at least one false statement* x assuming that the underlying language L is hard. Moreover, if L is decisionally hard-on-average – meaning that there are computationally indistinguishable distributions $\mathcal{D}_{\mathrm{Yes}} \approx_c \mathcal{D}_{\mathrm{No}}$ on YES-instances and NO-instances, respectively – then (Π, Sim) is average-case challenge hiding for the distribution $\mathcal{D}_{\mathrm{No}}$. However, challenge hiding against query-bounded adversaries does not follow formally from hardness of the underlying language and SHVZK.

Theorem 6.7. *Suppose that* $\Pi := \Pi^{\mathcal{O}(\cdot)}$ *is a 3-message public-coin HVZK argument system (with simulator* Sim*) relative to an (efficiently simulatable) oracle distribution* $\mathcal{O}$ *satisfying* query-bounded *HVZK. Moreover, suppose that*

1. *The underlying language* $L \notin \mathsf{BPP}$*,*
2. *The Verifier's challenge space* Σ *is polynomial-size, and*

3. (Π, Sim) *is challenge hiding (Definition 6.4).*

Then, for any t and any hash function h (that does not query the oracle), the protocol $\Pi^t_{\mathrm{FS},h}$ is unsound relative to $\mathcal{O}$. Alternatively, if

1′. *L is hard-on-average for a distribution $\mathcal{D}$ on no-instances*

2. *Σ is polynomial-size, and*

3′. (Π, Sim) *is $\mathcal{D}$-average-case challenge hiding,*

then $\Pi^t_{\mathrm{FS},h}$ is unsound as above.

Proof. We prove the "worst-case" variant of the theorem; the "average-case" variant follows by an almost identical argument.

We describe a polynomial-query attack on $\Pi^t_{\mathrm{FS},h}$. Given the oracle $\mathcal{O}$ and an instance x, do the following, with parameter $q = t|\Sigma|\lambda$:

1. For $1 \le i \le t$, $1 \le \ell \le q$:
   - Sample a uniformly random challenge $\beta_i^{(\ell)} \leftarrow \Sigma$
   - Sample fake transcripts $(\alpha_i^{(\ell)}, \gamma_i^{\ell)}) \leftarrow \mathsf{Sim}^{\mathcal{O}(\cdot)}(x, \beta_i^{(\ell)})$ using the special honest-verifier zero-knowledge simulator.
2. Given (α, β), search for a vector $v \in [q]^t$ such that $h(\alpha[v]) = \beta[v]$.
3. If such a v exists, output $(\alpha[v], \beta[v], \gamma[v])$.

We claim that for some $x \notin L$, this attack outputs an accepting transcript with high probability. To prove this, we have to show two things occur (with non-negligible probability): Step (2) successfully finds a vector v as described, and that the resulting transcript is accepting.

We first show that the former event occurs for *every* x. To see this, consider the following hybrid experiment (cut off at step (2)):

1. For $1 \le i \le t$, $1 \le \ell \le q$:
   - Sample i.i.d. uniformly random challenges $\beta_i^{(\ell)}, \tilde{\beta}_i^{(\ell)} \leftarrow \Sigma$.
   - Sample fake transcripts $(\alpha_i^{(\ell)}, \gamma_i^{\ell)}) \leftarrow \mathsf{Sim}^{\mathcal{O}(\cdot)}(x, \tilde{\beta}_i^{(\ell)})$ using the special honest-verifier zero-knowledge simulator.
2. Given (α, β), search for a vector $v \in [q]^t$ such that $h(\alpha[v]) = \beta[v]$.

We claim that the probability that such a vector v exists in the hybrid experiment is indistinguishable from the analogous probability in the real attack. This follows from the Challenge-Hiding property of (Π, Sim) (since in both experiments, only polynomially many queries are made to $\mathcal{O}$).

Moreover, in the hybrid experiment, the probability that such a vector v exists is $1 - O(\frac{1}{\lambda t})$ by Lemma 6.1. Therefore, we conclude that for all statements x, our attack successfully outputs a tuple $(\alpha[v], \beta[v], \gamma[v])$ such that $h(\alpha[v]) = \beta[v]$ with high probability.

To complete the proof of Theorem 6.7, we show that there exists some $x \notin L$ such that with probability $1 - \mathrm{negl}(\lambda)$ over the randomness of each $(\alpha_i^{(\ell)}, \gamma_i^{(\ell)}) \leftarrow \mathsf{Sim}(x, \beta_i^{(\ell)})$, the transcript $(\alpha_i^{(\ell)}, \beta_i^{(\ell)}, \gamma_i^{(\ell)})$ is accepting. This follows from the SHVZK simulation security of Π, the completeness of Π, as well as the hardness of L. In more detail, by the completeness and SHVZK of Π, we know that simulated transcripts $(\alpha_i^{(\ell)}, \beta_i^{(\ell)}, \gamma_i^{(\ell)})$ are accepting with probability $1 - \mathrm{negl}(\lambda)$ whenever $x \in L$. Since this property can be verified in polynomial *time*, we conclude that if $L \notin \mathsf{BPP}$, there exists some $x \notin L$ such that simulated transcripts are accepting with probability $1 - \mathrm{negl}(\lambda)$ as well.

Thus, we conclude that our polynomial-query attack breaks the soundness of $\Pi^t_{\mathrm{FS},h}$, completing the proof of Theorem 6.7. □

As a corollary to Theorem 6.7, we obtain explicit polynomial-query attacks on the soundness of $\Pi^t_{\mathrm{FS},h}$ – for *any* hash function h – for a large class of interactive protocols Π. For example, any "commit-challenge-response" style argument system [Blu86, GMW87, IKOS07] instantiated using a commitment scheme that is hiding against bounded-query adversaries in the ROM satisifes the hypotheses of Theorem 6.7, and so Fiat-Shamir cannot be instantiated for such protocols if the Fiat-Shamir hash function does not depend on the random oracle. An analogous result holds for the "single bit challenge" variant of the Schnorr identification protocol [Sch90] in the generic group model.

6.4 A General "Cryptography is Necessary" Result

We move on to our second result, which states that for a broad class of interactive protocols, any sound Fiat-Shamir hash function h (or family $\mathcal{H}$) necessarily satisfies a cryptographic security property. This result holds both in the standard model and relative to any efficiently simulatable oracle distribution (which makes the negative result even stronger). The security property we consider is a computational hardness assumption about making Lemma 6.1 *effective*.

Definition 6.8 ((q, Σ)-Mix-and-Match Resistance)**.** A hash function (family) $\mathcal{H}$ with output space Σ^t is *mix-and-match resistant* with parameters (q, Σ) if a computationally bounded adversary cannot win the following game with non-negligible probability:

- The challenger samples a hash function $H \leftarrow \mathcal{H}$.
- The challenger samples t uniformly random $q \times w$ matrices $A^{(1)}, \ldots, A^{(t)}$ as well as uniformly random $y^{(1)}, \ldots, y^{(t)} \leftarrow \Sigma^q$.
- The challenger sends (H, A, y) to the adversary.
- The adversary outputs a string $v \in [q]^t$.
- The adversary wins if $y[v] = H(A[v])$.

By Lemma 6.1, we know that for $q \geq t|\Sigma|\lambda$, an unbounded adversary can win the mix-and-match resistance security game with probability $1 - o(1)$. We emphasize that the matrices $A^{(1)}, \ldots, A^{(t)}$ are uniformly random in Definition 6.8 so that mix-and-match resistance is a single, universal security property that will not depend on the protocols Π discussed below.

Theorem 6.9. *Suppose that $\Pi := \Pi^{\mathcal{O}(\cdot)}$ is a 3-message public-coin HVZK argument system (with simulator* Sim*) relative to an efficiently simulatable oracle distribution $\mathcal{O}$. Moreover, suppose that*

1. *The underlying language $L \notin \mathsf{BPP}$,*
2. *The challenge space Σ is polynomial-size, and*
3. *First messages are pseudorandom: that is, for every $(x, w) \in R_L$, the first message $\alpha \leftarrow P(x, w)$ is computationally pseudorandom.*

Finally, suppose that a hash function family $\mathcal{H}$ (which does not make use of the oracle $\mathcal{O}$) securely instantiates the Fiat-Shamir heuristic for Π^t.

Then, $\mathcal{H}$ is (q, Σ)-mix-and-match resistant (Definition 6.8) with $q = t|\Sigma|\lambda$. Alternatively, if

1. *The underlying language L is hard-on-average for a a distribution $\mathcal{D}_{Yes}$ on pairs (x, w),*
2. *The challenge space Σ is polynomial-size, and*
3. *First messages are pseudorandom-on-average: that is, the distribution $(x, \alpha \leftarrow P(x, w))$ is computationally indistinguishable from $(x, \$)$, for $(x, w) \leftarrow \mathcal{D}_{Yes}$.*

Then, the same conclusion holds.

Proof. We prove the "worst-case" variant of the theorem; the "average-case" variant follows by an almost identical argument.

Let $\mathcal{H}$ be a hash function family with the appropriate input/output lengths, and assume that $\mathcal{H}$ is not mix-and-match resistant. Then, there is a polynomial-time algorithm $\mathcal{A}$ breaking the mix-and-match security game for $\mathcal{H}$. Assuming that $L \notin \mathsf{BPP}$, we use $\mathcal{A}$ to break the soundness of $\Pi^t_{\mathrm{FS},\mathcal{H}}$ (relative to $\mathcal{O}$) in polynomial time.

The attack $P^{*\mathcal{O}(\cdot)}$ is as follows for an arbitrary instance x and hash function $H \leftarrow \mathcal{H}$:

1. For $1 \le i \le t$, $1 \le \ell \le q$, sample fake transcripts $(\alpha_i^{(\ell)}, \beta_i^{(\ell)}, \gamma_i^{\ell)}) \leftarrow \mathsf{Sim}^{\mathcal{O}(\cdot)}(x, \beta_i^{(\ell)})$ using the special honest-verifier zero-knowledge simulator on a uniformly random $\beta_i^{(\ell)} \leftarrow \Sigma$.
2. Call $\mathcal{A}(\alpha, \beta)$ to obtain a vector $v \in [q]^t$.
3. Output $(\alpha[v], \beta[v], \gamma[v])$.

It now suffices to show that assuming Π is HVZK, Π has pseudorandom first messages, and $L \notin \mathsf{BPP}$, there exists $x \notin L$ such that $P^{*\mathcal{O}(\cdot)}(x)$ outputs an accepting transcript with non-negligible probability.

To prove this, we consider a sequence of claims that each suffice.

Claim 6.10. *For all $x \in L$, $P^{*\mathcal{O}(\cdot)}(x)$ outputs an accepting transcript with non-negligible probability.*

Assuming Claim 6.10, since P^* is efficient, we conclude that if $L \notin \mathsf{BPP}$, there exists an $x \notin L$ such that $P^{*\mathcal{O}(\cdot)}(x)$ outputs an accepting transcript with non-negligible probability. Otherwise, P^* can be used as an experiment to decide L.

Thus, it suffices to prove Claim 6.10. Let $(x, w) \in R_L$ be an arbitrary instance-witness pair. We now consider the following hybrid algorithm Hybrid, which is a modification (changes in red) of P^*.

1. For $1 \le i \le t$, $1 \le \ell \le q$, sample real transcripts $(\alpha_i^{(\ell)}, \beta_i^{(\ell)}, \gamma_i^{\ell)}) \leftarrow \langle P^{\mathcal{O}(\cdot)}(x, w), V^{\mathcal{O}(\cdot)} \rangle$ (playing the role of both the prover and verifier).
2. Call $\mathcal{A}(\alpha, \beta)$ to obtain a vector $v \in [q]^t$.
3. Output $(\alpha[v], \beta[v], \gamma[v])$.

By the honest-verifier zero-knowledge of Π, the algorithm Hybrid outputs an accepting transcript with the same probability as that of $\mathcal{A}$ (up to negligible difference).

Finally, we note that in an execution of Hybrid, $\mathcal{A}(\alpha, \beta)$ is being called on a joint distribution that is computationally indistinguishable from uniform (since α is pseudorandom and β is independent of α). Therefore, the call to $\mathcal{A}$ in Hybrid outputs a v such that $H(\alpha[v]) = \beta[v]$ with non-negligible probability. Whenever this condition holds, the transcript $(\alpha[v], \beta[v], \gamma[v])$ is accepting, so we conclude that Hybrid (and the actual cheating prover P^*) outputs an accepting transcript with non-negligible probability. This completes the proofs of Claim 6.10 and Theorem 6.9. □

Note that Theorem 6.9 applies to protocols in the random oracle model, in the generic group model, and in the standard model (that is, the reduction makes black-box calls to the HVZK simulator but not to the oracle $\mathcal{O}$ itself). Therefore, this negative result applies to many 3-message argument systems, such as:

- Blum's Hamiltonicity protocol [Blu86], when the commitment scheme Com outputs pseudorandom values (i.e. Naor commitments [Nao90] in the CRS model or Blum commitments [Blu81]) in the plain model).
- The [GMW87] 3-coloring protocol with either of the above choices of commitment scheme.

- The [IKOS07] "MPC-in-the-head" proof system, for any MPC protocol, when the commitment scheme is instantiated as above.
- The $\{0,1\}$-challenge variant of Schnorr's identification scheme [Sch90], even in the generic group model.
- The [GMR85] proof system for Quadratic Residuosity.
- A simple proof system for bounded distance decoding (BDD) problem based on the natural "instance-dependent commitment scheme [BMO90, IOS97]" for BDD. On a (worst-case) instance $(A, y = sA+e)$ of this language, the prover sends a "commitment" $s'A + e'$ for random s' and random noise e' that floods e. On a challenge bit b, the prover replies with $s' + bs$. This can also be thought of as a simplification of the [MV03] proof system for gap-CVP (restricted to BDD instances). Under the LWE assumption, this protocol has pseudorandom first messages on random instances, so Theorem 6.9 applies.

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

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, Heidelberg, August 2009.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.

[BBC$^+$17] Eli Ben-Sasson, Iddo Bentov, Alessandro Chiesa, Ariel Gabizon, Daniel Genkin, Matan Hamilis, Evgenya Pergament, Michael Riabzev, Mark Silberstein, Eran Tromer, and Madars Virza. Computational integrity with a public random string from quasi-linear PCPs. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 551–579. Springer, Heidelberg, April / May 2017.

[BBHR18a] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *ICALP 2018*, volume 107 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl, July 2018.

[BBHR18b] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. Cryptology ePrint Archive, Report 2018/046, 2018. https://eprint.iacr.org/2018/046.

[BBHR19] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable zero knowledge with no trusted setup. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 701–732. Springer, Heidelberg, August 2019.

[BCR$^+$19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 103–128. Springer, Heidelberg, May 2019.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 31–60. Springer, Heidelberg, October / November 2016.

[BKM20] Zvika Brakerski, Venkata Koppula, and Tamer Mour. Nizk from lpn and trapdoor hash via correlation intractability for approximable relations, 2020. ePrint:2020/258.

[Blu81] Manuel Blum. Coin flipping by telephone. In Allen Gersho, editor, *CRYPTO'81*, volume ECE Report 82-04, pages 11–15. U.C. Santa Barbara, Dept. of Elec. and Computer Eng., 1981.

[Blu86] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2. Citeseer, 1986.

[BMO90] Mihir Bellare, Silvio Micali, and Rafail Ostrovsky. Perfect zero-knowledge in constant rounds. In *22nd ACM STOC*, pages 482–493. ACM Press, May 1990.

[BR94] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 232–249. Springer, Heidelberg, August 1994.

[BV11] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 97–106. IEEE Computer Society Press, October 2011.

[CCH$^+$19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1082–1090. ACM Press, June 2019.

[CCR16] Ran Canetti, Yilei Chen, and Leonid Reyzin. On the correlation intractability of obfuscated pseudorandom functions. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 389–415. Springer, Heidelberg, January 2016.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 91–122. Springer, Heidelberg, April / May 2018.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th ACM STOC*, pages 209–218. ACM Press, May 1998.

[CH19] Geoffroy Couteau and Dennis Hofheinz. Designated-verifier pseudorandom generators, and their applications. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 562–592. Springer, Heidelberg, May 2019.

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer, Heidelberg, May / June 2010.

[CKU20] Geoffroy Couteau, Shuichi Katsumata, and Bogdan Ursu. Non-interactive zero-knowledge in pairing-free groups from weaker assumptions. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 442–471. Springer, Heidelberg, May 2020.

[CP93] David Chaum and Torben P. Pedersen. Wallet databases with observers. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 89–105. Springer, Heidelberg, August 1993.

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 13–25. Springer, Heidelberg, August 1998.

[Dam10] Ivan Damgard. On sigma-protocols, lecture notes, faculty of science aarhus university, department of computer science, 2010.

[DGI$^+$19] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor hash functions and their applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 3–32. Springer, Heidelberg, August 2019.

[DNRS99] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry J. Stockmeyer. Magic functions. In *40th FOCS*, pages 523–534. IEEE Computer Society Press, October 1999.

[DRV12] Yevgeniy Dodis, Thomas Ristenpart, and Salil P. Vadhan. Randomness condensers for efficiently samplable, seed-dependent sources. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 618–635. Springer, Heidelberg, March 2012.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM J. Comput.*, 29(1):1–28, September 1999.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009.

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th FOCS*, pages 464–479. IEEE Computer Society Press, October 1984.

[GK90] Oded Goldreich and Hugo Krawczyk. Sparse pseudorandom distributions. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 113–127. Springer, Heidelberg, August 1990.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *14th ACM STOC*, pages 365–377. ACM Press, May 1982.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 171–185. Springer, Heidelberg, August 1987.

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

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

[HL18] Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In Mikkel Thorup, editor, *59th FOCS*, pages 850–858. IEEE Computer Society Press, October 2018.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007.

[IOS97] Toshiya Itoh, Yuji Ohta, and Hiroki Shizuya. A language-dependent cryptographic primitive. *Journal of Cryptology*, 10(1):37–50, December 1997.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th ACM STOC*, pages 723–732. ACM Press, May 1992.

[KNYY19] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. Designated verifier/prover and preprocessing NIZKs from Diffie-Hellman assumptions. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 622–651. Springer, Heidelberg, May 2019.

[KRR17] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 224–251. Springer, Heidelberg, August 2017.

[LV20] Alex Lombardi and Vinod Vaikuntanathan. Fiat-shamir for repeated squaring with applications to ppad-hardness and vdfs, 2020. To appear in CRYPTO 2020.

[LW15] Vadim Lyubashevsky and Daniel Wichs. Simple lattice trapdoor sampling from a broad class of distributions. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 716–730. Springer, Heidelberg, March / April 2015.

[Lyu08] Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In Ronald Cramer, editor, *PKC 2008*, volume 4939 of *LNCS*, pages 162–179. Springer, Heidelberg, March 2008.

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 598–616. Springer, Heidelberg, December 2009.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Heidelberg, April 2012.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000.

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Heidelberg, April 2012.

[MV03] Daniele Micciancio and Salil P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 282–298. Springer, Heidelberg, August 2003.

[MV16] Arno Mittelbach and Daniele Venturi. Fiat-Shamir for highly sound protocols is instantiable. In Vassilis Zikas and Roberto De Prisco, editors, *SCN 16*, volume 9841 of *LNCS*, pages 198–215. Springer, Heidelberg, August / September 2016.

[Nao90] Moni Naor. Bit commitment using pseudo-randomness. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 128–136. Springer, Heidelberg, August 1990.

[Nec94] V. I. Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994.

[NSW09] Gregory Neven, Nigel P Smart, and Bogdan Warinschi. Hash function requirements for schnorr signatures. *Journal of Mathematical Cryptology*, 3(1):69–87, 2009.

[PS96] David Pointcheval and Jacques Stern. Provably secure blind signature schemes. In Kwangjo Kim and Tsutomu Matsumoto, editors, *ASIACRYPT'96*, volume 1163 of *LNCS*, pages 252–265. Springer, Heidelberg, November 1996.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 89–114. Springer, Heidelberg, August 2019.

[QRW19] Willy Quach, Ron D. Rothblum, and Daniel Wichs. Reusable designated-verifier NIZKs for all NP from CDH. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 593–621. Springer, Heidelberg, May 2019.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, August 1990.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997.

[WTs+18] Riad S. Wahby, Ioanna Tzialla, abhi shelat, Justin Thaler, and Michael Walfish. Doubly-efficient zkSNARKs without trusted setup. In *2018 IEEE Symposium on Security and Privacy*, pages 926–943. IEEE Computer Society Press, May 2018.

A Correlation Intractability and the Idealized Blum Protocol

In this section, we show that correlation intractability for efficiently computable functions [CCH+19, PS19] implies a sound instantiation of Fiat-Shamir for a variant of the idealized Blum protocol (Section 6.2).

First, we recall a minor modification of the Blum protocol (as in [CCH+19, PS19]) and instantiate the commitment scheme with a random oracle, as in Section 6.2.

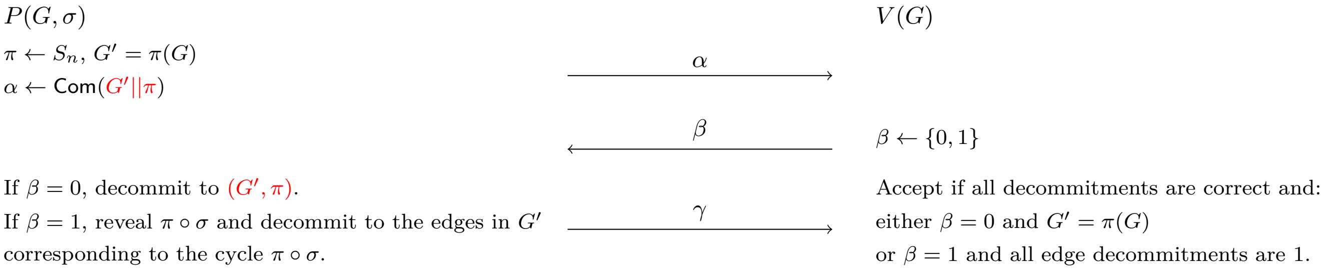

$P(G, \sigma)$
$V(G)$
$\pi \leftarrow S_n$, $G' = \pi(G)$
$\alpha \leftarrow \mathsf{Com}(G' || \pi)$
α
β
$\beta \leftarrow \{0, 1\}$
If $\beta = 0$, decommit to (G', π).
If $\beta = 1$, reveal $\pi \circ \sigma$ and decommit to the edges in G' corresponding to the cycle $\pi \circ \sigma$.
γ
Accept if all decommitments are correct and:
either $\beta = 0$ and $G' = \pi(G)$
or $\beta = 1$ and all edge decommitments are 1.

Figure 10: A Modified Idealized Blum Protocol Π

That is, we require the prover to additionally commit to the permutation π and decommit to π if $\beta = 0$. In this case, the verifier checks that π is a valid permutation and that $G' = \pi(G)$. The reason this modification is made is so that given a (partial) decommitment to the first message α, it is possible to efficiently decide which challenge is answerable using this decommitment. In the original Blum protocol, the analogous computation requires solving a graph isomorphism problem.

As before, we instantiate $\mathsf{Com}(b; r) = \mathcal{O}(b, r)$ using a random oracle. Concretely, we set $|r| = \lambda = \lambda(n)$ and $|\mathcal{O}(b, r)| = \kappa = \kappa(n)$ to be arbitrary polynomial functions in $n = |V(G)|$. The protocol above is then repeated $t = t(n)$ times in parallel to obtain negligible soundness error. We then prove:

Theorem A.1. *Suppose that for every (efficiently computable) $s(n) = \mathrm{poly}(n)$, there exists a hash family $\mathcal{H} = \{h_k : \{0,1\}^{m(n)\kappa(n)t(n)} \to \{0,1\}^{t(n)}\}_{k\in\{0,1\}^{\ell(n)}}$ (for $m(n) = n^2 + n$) that is correlation intractable for all functions computable by size $s(n)$ circuits.*

Then, for an appropriate fixed choice of function $s(\cdot)$, the same hash family $\mathcal{H}$ soundly instantiates the Fiat-Shamir heuristic for the protocol Π^t in the random oracle model.

By Theorem 6.9, we also obtain the following corollary.

Corollary A.2. *Under the hypothesis of Theorem A.1, the hash family $\mathcal{H}$ is also (q, Σ) mix-and-match resistant (Definition 6.8) for $\Sigma = \{0,1\}$ and arbitrary $q = \mathrm{poly}(n)$.*

We now prove Theorem A.1.

Proof. Let $\mathcal{H}$ be a family of correlation-intractable hash functions with parameters as above (for $s = s(n)$ chosen appropriately large). Since correlation-intractable hash functions imply the existence of one-way functions, we additionally let $F_s : \{0,1\}^{\kappa(n)-1} \to \{0,1\}$ be a PRF family computable by a family of circuits of size $s(n)$.

Now, suppose that an efficient adversary $\mathcal{A}^{\mathcal{O}(\cdot)}$, given a non-Hamiltonian graph G and random hash function h, breaks the soundness of $\Pi^t_{\mathrm{FS},\mathcal{H}}$ on G.

Let $\tau = \tau(\mathcal{A}, \mathcal{O})$ denote the transcript of $\mathcal{O}$-queries made by $\mathcal{A}$; that is, for every i, $\tau_i = (b_i, r_i, c_i)$ where (b_i, r_i) is the ith query made by $\mathcal{A}$ to $\mathcal{O}$, and $c_i = \mathcal{O}(b_i, r_i)$. Finally, let $(\alpha^*, \beta^*, \gamma^*)$ denote the output of $\mathcal{A}$.

Given an arbitrary first message α and transcript τ, we say that a challenge β is a **bad challenge** for (α, τ) if the following conditions hold:

- For every i such that $\beta_i = 0$, the string of commitments $\alpha_i = (c_{i,0}, \dots, c_{i,m})$ is entirely contained within the transcript τ, and the corresponding bits $\{b_{i,j}\}$ consist of a permutation π and the graph $\pi(G)$.
- For every i such that $\beta_i = 1$, the transcript τ contains a substring of α_i consisting of commitments to a cycle.

We now note a sequence of facts about the execution of $\mathcal{A}$.

Claim A.3. *The probability that $\mathcal{A}^{\mathcal{O}(\cdot)}$ wins with output $(\alpha^*, \beta^*, \gamma^*)$* and *$\beta^*$ is* not *a bad challenge for (α^*, τ) is negligible.*

This claim follows from binding properties of the (random oracle) commitment scheme. This is because if β^* is not bad for (α^*, τ) but $(\alpha^*, \beta^*, \gamma^*)$ is accepting, then γ^* contains decommitments to bits that are not present in τ; this means that $\mathcal{A}^{\mathcal{O}(\cdot)}$ solves an (unconditionally) hard problem in the random oracle model.

Claim A.4. *The probability that (α^*, τ) has* multiple *bad challenges associated to it is negligible.*

This again follows from binding properties of the commitment scheme, and the fact that G is not Hamiltonian. Since G is Hamiltonian, if no string c appears twice (for two different choices of (b, r)) in the transcript τ, bad challenges for any (α, τ) are unique (as each α_i cannot have an opening to both a permutation of G and a Hamiltonian graph simultaneously). However, τ only contains the same commitment string c twice with negligible probability, since it is (unconditionally) hard to find $\mathcal{O}$-collisions.

Thus, given a transcript τ and message α, we define the efficiently computable "transcript bad-challenge function" $f(\tau, \alpha)$ as follows:

- If α_i is present in τ as a commitment to (G', π) and $G' = \pi(G)$, set $\beta_i = 0$.

- Otherwise, set $\beta_i = 1$.
- Output $\beta = (\beta_1, \ldots, \beta_t)$.

By the above analysis, we conclude:

Claim A.5. *With non-negligible probability, the adversary $\mathcal{A}^{\mathcal{O}}(G, h)$ outputs $(\alpha^*, \beta^*, \gamma^*)$ such that*

- $\beta^* = h(\alpha^*) = f(\alpha^*, \tau)$*, and*
- τ *contains all necessary decommitments to answer the challenge* β^**.*

Note that Claim A.5 is an efficiently decidable property of $(\tau, \alpha^*, \beta^*)$. Thus, Claim A.5 *also* holds if we replace the truly random oracle $\mathcal{O}$ with the following oracle distribution $\mathcal{O}'$:

- $\mathcal{O}'$ has a hard-coded random seed s for the PRF $F_s : \{0,1\}^{\kappa(n)-1} \to \{0,1\}$
- $\mathcal{O}'(b, r)$ samples a uniformly random $r' \leftarrow \{0,1\}^{\kappa(n)-1}$ and outputs $(r', F_s(r') \oplus b)$.

This follows directly from the pseudorandomness property of the PRF family. Finally, we define the following efficiently computable function $g_s : \{0,1\}^{m(n)\kappa(n)t(n)} \to \{0,1\}^{t(n)}$.

- Input: $\alpha = (\alpha_1, \ldots, \alpha_n)$
- For all i, let $\alpha_i = (c_{i,1}, \ldots, c_{i,m(n)})$ and $c_{i,j} = r'_{i,j}||b'_{i,j}$. Compute $b_{i,j} = F_s(r'_{i,j}) \oplus b'_{i,j}$.
- Let $\tilde{\tau}$ denote a transcript containing triples of the form $(b_{i,j}, r_{i,j}, c_{i,j})$ where $r_{i,j}$ are arbitrary.
- Output $f(\alpha, \tilde{\tau})$.

We claim that $\mathcal{A}^{\mathcal{O}'(\cdot)}$ breaks the correlation intractability of $\mathcal{H}$ with respect to the function g_s. Indeed, whenever the conditions of Claim A.5 hold, we also claim that $h(\alpha^*) = g_s(\alpha^*)$. To see this, we note that any commitment $c = (r', b')$ occurring as (b, r, c) in the transcript τ must satisfy the property $b' = F_s(r') \oplus b$. Thus, the ith bit $f(\alpha^*, \tau)_i = 0$ if and only if the ith bit $g_s(\alpha^*)_i = 0$.

We conclude that $\mathcal{A}^{\mathcal{O}'(\cdot)}$, which can be implemented efficiently given the PRF seed s, contradicts the correlation intractability of $\mathcal{H}$ with respect to g_s. Therefore, the protocol $\Pi^t_{\mathrm{FS},\mathcal{H}}$ is indeed sound in the ROM. □