Tree Encodings II: Correlation Intractability for all Batched Relations

Damiano Abram* Giulio Malavolta† Lawrence Roy‡

Abstract

The Fiat-Shamir transform is a central tool in cryptography and understanding its soundness is both a theoretically challenging and practically pressing question. Correlation intractable hash functions offer a method to instantiate Fiat-Shamir in the standard model. In short, a hash function is correlation-intractable for a relation $\mathcal{R}$ if it is computationally hard to find an input x such that $\mathcal{R}(x, \mathsf{Hash}(x)) = 1$.

In this work we present the first construction of a correlation-intractable hash function family, where the complexity of the hash does not depend on the complexity of the relation. Besides being better aligned with the way Fiat-Shamir is used in practice, our construction implies correlation intractability for all (possibly inefficient) batched searchable relations, i.e., searchable relations that can be decomposed as a direct product of other searchable relations. Using complexity leveraging, we then compile our construction into correlation intractability for all sufficiently sparse batched relations. All of our results follow from the hardness of the decomposed short-integer solution (DSIS) problem, the natural search analogue of decomposed LWE, which we show to be at least as hard.

As a direct consequence from prior work, our result implies that sufficiently many parallel repetitions of any perfectly complete public-coin three-message proof for a language outside BPP are not zero-knowledge. Moreover, we obtain non-interactive zero-knowledge (NIZKs) arguments for NP from the sub-exponential hardness of DSIS.

1 Introduction

The Fiat–Shamir transform [FS87] is an influential method for eliminating interaction from public-coin protocols. By deriving the verifier's challenges as the output of a hash function applied to the prover's commitment, Fiat–Shamir enables efficient non-interactive proofs and signatures and is used pervasively in modern cryptographic systems.

Understanding the security of Fiat–Shamir in the standard model has long been a long-standing problem. It is known that Fiat–Shamir is not sound in general [GK03] and recent attacks on practically-deployed protocols [KRS25] underscore the importance of understanding when the transform guarantees soundness. The notion of correlation intractability [CGH98] was proposed as a unifying abstraction capturing some of the properties required of the hash function in Fiat–Shamir. For a binary relation $\mathcal{R}(x, y)$, a hash family $(\mathsf{Setup}, \mathsf{Hash})$ is *correlation-intractable* for $\mathcal{R}$ if, given a hash function $\mathsf{hk} \xleftarrow{\$} \mathsf{Setup}$, it is computationally hard to find an input x such that $\mathcal{R}(x, \mathsf{Hash}(\mathsf{hk}, x))$

*University of Edinburgh
†Bocconi University
‡Aarhus University, IBM Research Zürich

is true. For this definition to be achievable, we additionally require that the relation $\mathcal{R}$ is sparse: for any x, all but a negligible fraction of y do not satisfy the relation with x. A typical example are searchable relations, where there exists a function f such that, for every x, $(x, y) \in \mathcal{R}$ only if $y = f(x)$. We say that the relation is efficiently searchable if f is efficiently computable.

In recent years, there has been substantial progress in constructing correlation-intractable hash functions for efficiently searchable relations [CCR16, KRR17, CCRR18, HL18, CCH+19, PS19, LV22] and even the Cartesian product of efficiently-checkable relations [HLR21]. This paradigm has led to enormous advances in establishing the security of cryptographic protocols in the standard model, leading to new constructions of non-interactive zero-knowledge proofs [PS19, JJ21], batch [CJJ21, DGKV22, PP22], and succinct [CJJ22, KLVW23, BBK+23] arguments, to mention a few.

Nevertheless, our understanding of the Fiat-Shamir transformation remains limited: for one, the complexity of *all known* correlation-intractable hashes in the standard model scales with the time needed to check the relation. While this is seemingly a technical point, there are multiple reasons why this aspect is unsatisfactory: (i) for starters, it places a lower bound on the concrete efficiency of the hash function, moreover (ii) it limits correlation-intractability to *efficiently searchable* relations, whereas interactive protocols use in practice do no necessarily enjoy this property. Finally, (iii) it does not reflect how the Fiat-Shamir transform is used in real-world protocols, where the complexity of the hash is typically independent of that of the underlying protocol.

Taken together, the above limitations underscore our lack of understanding of the soundness of the Fiat-Shamir transform, which is one of the most pressing problems in the theory of modern cryptography. The goal of this work is to make progress on this question.

1.1 Our Results

In this work we propose a construction of correlation intractable hash functions for *all sufficiently sparse batched* relations, i.e., relations that consist of the direct product of other relations

$$\mathcal{R} := \mathcal{R}^{(0)} \times \cdots \times \mathcal{R}^{(c-1)}.$$

We do not place *any computational restriction* on the relations that we consider, with the constraints that their sparsity is bounded away from 1 by a non-negligible amount and that c must be a sufficiently large polynomial in the security parameter. In particular, we obtain the first correlation intractability hash functions for *not efficiently searchable* relations, with provable security against a falsifiable assumption.

A necessary condition for this to be possible is that the complexity of our hash function cannot depend on the complexity of the relation $\mathcal{R}$. Thus, in order to achieve our main result, we need to overcome a technical barrier present in, as far as we are aware, all prior work on correlation intractability in the standard model. In the batched settings, this answers affirmatively an open question by Lombardi [Lom25].

We prove that our hash function family is correlation intractable assuming the (sub-exponential) hardness of the decomposed short integer solution (DSIS) problem, a problem that we introduce in this work. DSIS is the natural dual of the decomposed LWE problem [AMR25a] and in this work we show that it is at least as hard as decomposed LWE. In turn, it is known that the hardness of the decomposed LWE problem is implied by the hardness of the succinct LWE problem [Wee24, AMR25a].

We summarize our results with the following theorem statement.

Theorem 1 (Informal)**.** *Assuming the sub-exponential hardness of the DSIS problem, there exists a correlation-intractable hash function for all sufficiently sparse batched relations.*

Let us now be more precise about our hardness assumption. We consider the DSIS problem with two parameter settings, in both cases against an adversary finding a short solution with norm bounded by 2^{k^β}, for some constant $\beta > 0$, where k denotes the dimension of the lattice.

In the standard sub-exponential settings, i.e., where the assumption postulates hardness against attackers running in time 2^{k^γ} for any constant $0 < \gamma < 1$, we obtain correlation intractability for batched *searchable relations* (see Corollary 1 for the exact parameters).

To prove correlation intractability for all (possibly non-searchable) batched relations, we require a slightly stronger assumption. That is, we require that DSIS is hard to break with probability greater than 2^{-k^ν} for attackers running in time 2^{k^μ}, where $\mu, \nu > 0$ are constants such that $\min\{\mu, \beta\} + \nu > 1 + \alpha$, $m, \sigma \leq k^{O(1)}$, and $\log q - \log B = \Theta(k^\alpha)$. The exact parameters of the assumption can be found in Corollary 2.

We also note that, scaling down the parameters, we obtain, under the *polynomial* DSIS assumption, a correlation intractable function for any *efficiently searchable* batched relation with an a priori bound on the depth of its search circuits. Note that this does not follow from prior work, since the complexity of our hash function is *independent* of the circuit size of the relation.

Leveraging a connection from prior work [DNRS99, HLR21], we obtain the following implication as an immediate corollary: Sufficiently many parallel repetitions of any perfectly complete public-coin three-round proof for a language outside BPP with a non-trivial soundness gap are not zero-knowledge. This improves upon [HLR21] that showed the same implication for the special case of the GMW protocol [GMW86]. On the positive side, following the approach of [CCH+19], we obtain non-interactive zero-knowledge (NIZKs) arguments for NP, assuming the hardness of DSIS. To the best of our knowledge, this is the first NIZK in the standard model based on (a variant of) the short integer solution problem with a classical reduction.[1]

Theorem 2 (Informal)**.** *Assuming the sub-exponential hardness of the DSIS problem, every language in* NP *has an adaptively sound computational zero-knowledge non-interactive argument in the common reference string model.*

1.2 Comparison with Prior Work

Among the large literature on correlation intractability, we highlight the work of Canetti et al. [CCRR18], which also proposes a correlation-intractable hash function that plausibly supports inefficient relations. Compared to our work, the assumption from [CCRR18] (strong KDM security) quantifies over all functions supported by correlation intractability. In other words, strong KDM security is really a family of assumptions, one for each function f, and different functions correspond to different assumptions. This is undesirable from a cryptanalytic standpoint, since there is no unified source of hardness that can be tested.

Moreover, strong KDM security for an inefficient f requires an inefficient challenger, i.e., the experiment requires the challenger to compute the function f. This goes against the win-win spirit of assumptions in cryptography: If the assumption does not hold, one cannot extract an efficient algorithm to solve an interesting problem. On the other hand, DSIS has a polynomial-time

[1] The work of Regev [Reg05] shows that an LWE solver implies the existence of an SIS solver, but the reduction requires quantum operations, even if the original LWE solver is a classical algorithm.

challenger: If our assumption is broken, we obtain a sub-exponential time algorithm to solve DSIS, which would be an interesting algorithmic statement.

2 Technical Overview

We construct correlation intractability for all sufficiently sparse batched relations in two steps: first, we present a basic construction for batched searchable relations from *decomposed SIS*; then, we use complexity leveraging to compile the basic solution into a correlation intractable hash function for all sufficiently sparse batched relations.

Background: Lattice Encodings. We recall that a BGG^+-encoding [BGG+14] of a bit x consists of the vector

$$\mathbf{c}^\intercal = \mathbf{s}^\intercal(\mathbf{M} - x \cdot \mathbf{G}_q) + \mathbf{e}^\intercal$$

where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^k$ is a random secret key, $\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{k \times n}$, where $n := k \log q$, is a public key, $\mathbf{e} \xleftarrow{\$} \chi_\sigma^n$ is a low-norm noise term, and

$$\mathbf{G}_q := \mathbf{I}_k \otimes \mathbf{g}_q^\intercal = \begin{bmatrix} \mathbf{g}_q^\intercal & & & \\ & \mathbf{g}_q^\intercal & & \\ & & \ddots & \\ & & & \mathbf{g}_q^\intercal \end{bmatrix}, \qquad \mathbf{g}_q := \begin{bmatrix} 1 \\ 2 \\ \vdots \\ 2^{\log q - 1} \end{bmatrix}$$

is the gadget matrix [MP12]. These can be easily generalised to encodings of large vectors $\mathbf{x} \in \{0,1\}^L$ by setting

$$\mathbf{c}^\intercal = \mathbf{s}^\intercal(\mathbf{M} - \mathbf{x}^\intercal \otimes \mathbf{G}_q) + \mathbf{e}^\intercal,$$

note that this leads to an increase of the size of the public parameters $\mathbf{M}$ (which becomes of size $k \times Ln$) and of the encoding itself, which has size Ln. The key property of BGG^+-encodings is the well-known equation

$$\mathbf{M} \cdot \mathbf{K}_f - f(\mathbf{x})^\intercal \otimes \mathbf{G}_q = (\mathbf{M} - \mathbf{x}^\intercal \otimes \mathbf{G}_q) \cdot \mathbf{H}_{f,\mathbf{x}},$$

where $\mathbf{K}_f$ and $\mathbf{H}_{f,\mathbf{x}}$ are low-norm matrices that can be deterministically derived from the public parameters, f, and (only in the case of $\mathbf{H}_{f,\mathbf{x}}$) the value $\mathbf{x}$ itself. This provides a recipe to convert an encoding of $\mathbf{x}$ into an encoding of $f(\mathbf{x})$ under the same secret key $\mathbf{s}$, namely compute:

$$\mathbf{c}^\intercal \cdot \mathbf{H}_{f,\mathbf{x}} \approx \mathbf{s}^\intercal \cdot (\mathbf{M}_f - f(\mathbf{x})^\intercal \otimes \mathbf{G}_q)$$

ignoring noise terms, where $\mathbf{M}_f := \mathbf{M} \cdot \mathbf{K}_f$ is a matrix that depends only on $\mathbf{M}$ and the function f. Crucially, this operation only requires knowledge of $\mathbf{x}$, f and the public parameters, no information about the secret key $\mathbf{s}$ is necessary.

Background: Compressed Lattice Encodings. In [AMR25a, AMR25b] the authors show how to compress the encoding of $\mathbf{x}$ and the public parameters to size $\mathsf{poly}(\lambda, \log L)$ without sacrificing the homomorphic properties. There are two variants of this compression: The approach of [AMR25b] is provably secure under the hardness of LWE [Reg05], however, at the same time it requires modifications to the structure of BGG^+-encodings. On the other hand, [AMR25a] preserves the elegant mathematical structure of the BGG^+-encodings, but security can only be proven under

decomposed LWE. In this work we adopt the second approach, which we recall next. In more detail, the compressed encoding of $\mathbf{x}$ consists of two terms. The first one is

$$\mathbf{c}_0^\intercal = \mathbf{s}^\intercal \cdot (\mathbf{M} - \mathbf{h}^\intercal \otimes \mathbf{G}_q) + \mathbf{e}_0^\intercal$$

where $\mathbf{h}$ is a particular Merkle hash of the input $\mathbf{x}$ [Mer88]. This can also be viewed as a standard BGG^+-encoding of $\mathbf{h}$. The second term is an LWE sample

$$\mathbf{c}_1^\intercal = \mathbf{s}^\intercal \mathbf{Z} + \mathbf{e}_1^\intercal$$

where $\mathbf{Z}$ is the decomposed LWE matrix.

The interesting property is that the compressed encoding of $\mathbf{x}$ can be expanded into a standard BGG^+-encoding of any entry $\mathbf{x}[i]$ in time $\mathsf{poly}(\lambda, \log L)$. This essentially corresponds to a read operation in a RAM. The property is summarised by the following key equation

$$\mathbf{M} \cdot \mathbf{P}_i - \mathbf{x}[i] \cdot \mathbf{G}_q = (\mathbf{M} - \mathbf{h}^\intercal \otimes \mathbf{G}_q) \cdot \mathbf{P}_i + \mathbf{Z} \cdot \mathbf{Q}_{i,\mathbf{x}} \tag{1}$$

where $\mathbf{P}_i$ and $\mathbf{Q}_{i,\mathbf{x}}$ are low-norm matrices that can be deterministically derived in time $\mathsf{poly}(\lambda, \log L)$ from the public parameters, the index i, and (only in the case of $\mathbf{Q}_{i,\mathbf{x}}$) the value $\mathbf{x}$ itself.[2] We conclude that

$$\mathbf{c}_0^\intercal \cdot \mathbf{P}_i + \mathbf{c}_1^\intercal \cdot \mathbf{Q}_{i,\mathbf{x}} \approx \mathbf{s}^\intercal \cdot (\mathbf{M}_i - \mathbf{x}[i] \cdot \mathbf{G}_q)$$

where $\mathbf{M}_i := \mathbf{M} \cdot \mathbf{P}_i$ depends only on the public parameters and the index i, again ignoring noise terms.

Of course, once we have a BGG^+-encoding of the entries of $\mathbf{x}$, we can continue to apply standard BGG^+ homomorphic operations. Given that the hash $\mathbf{h}$ used in the compressed encodings is a Merkle hash, we can also change the value of any entry of $\mathbf{x}$, obtaining a compressed encoding of the updated string in time $\mathsf{poly}(\lambda, \log L)$: to do that it is sufficient to homomorphically recompute the hashes on the Merkle path that lead to the modified entry.

$$\mathbf{M} \cdot \mathbf{K}_f - \mathbf{h}_f^\intercal \otimes \mathbf{G}_q = (\mathbf{M} - \mathbf{h}^\intercal \otimes \mathbf{G}_q) \cdot \mathbf{H}_{f,\mathbf{x},0} + \mathbf{Z} \cdot \mathbf{H}_{f,\mathbf{x},1}, \tag{2}$$

where $\mathbf{h}_f$ is the Merkle hash of $f(\mathbf{x})$, and $\mathbf{K}_f, \mathbf{H}_{f,\mathbf{x},0}$ and $\mathbf{H}_{f,\mathbf{x},1}$ are low-norm matrices that can be derived in time $T \cdot \mathsf{poly}(\lambda, \log L)$, where T is the RAM complexity of f. As before, $\mathbf{K}_f$ depends only on f and the public parameters, whereas $\mathbf{H}_{f,\mathbf{x},0}$ and $\mathbf{H}_{f,\mathbf{x},1}$ depend also on $\mathbf{x}$.

Correlation Intractability for Searchable Relations. Consider any function $f : [L/n] \to \{0,1\}^n$ and define the searchable relation

$$\mathcal{R}_f := \{(x, y) \in [L/n] \times \{0,1\}^n | y = f(x)\}.$$

We present a simple recipe to build correlation intractability for $\mathcal{R}_f$ using the compressed encodings of [AMR25a]. The hash key consists of the public parameters of the compressed encodings, namely the random matrix $\mathbf{M}$ and the decomposed LWE matrix $\mathbf{Z}$. To hash a value $x \in [L/n]$, just compute

$$\mathbf{G}^{-1}(\mathbf{M} \cdot \begin{bmatrix} \mathbf{P}_{nx} & \mathbf{P}_{nx+1} & \cdots & \mathbf{P}_{nx+n-1} \end{bmatrix} \cdot \mathbf{r}),$$

[2] We assume that we have RAM access to $\mathbf{x}$.

where $\mathbf{r}$ is the low-norm vector that satisfies $(\mathbf{y}^\intercal \otimes \mathbf{G}_q) \cdot \mathbf{r} = \mathbf{G}_q \cdot \mathbf{y}$ for every $\mathbf{y} \in \{0,1\}^n$. That is it![3]

To see why this is correlation intractable, consider what happens if we substitute $\mathbf{M}$ with $\mathbf{M} \cdot \mathbf{K}_F + \mathbf{M}'$ where $\mathbf{M}'$ is sampled at random and F is the function that maps $\mathbf{0}^L$ to the vector

$$\mathbf{y} := \begin{bmatrix} f(0) \\ \vdots \\ f(L/n-1) \end{bmatrix}.$$

Of course, from the point of view of the adversary nothing has changed, so the probability of finding an input x that hashes to $\mathbf{y}_x := f(x)$ remains the same. However, if the attack is successful, we obtain that

$$(\mathbf{M} \cdot \mathbf{K}_F + \mathbf{M}') \cdot \begin{bmatrix} \mathbf{P}_{nx} & \mathbf{P}_{nx+1} & \cdots & \mathbf{P}_{nx+n-1} \end{bmatrix} \cdot \mathbf{r} = \mathbf{G}_q \cdot \mathbf{y}_x. \tag{3}$$

We use the linearity of the Merkle hash, under which the hash of $\mathbf{0}^L$ is $\mathbf{0}$. By applying Eq. (2) on $\mathbf{x} = \mathbf{0}^L$, we then have that

$$\mathbf{M} \cdot \mathbf{K}_F - \mathbf{h}_F^\intercal \otimes \mathbf{G}_q = \mathbf{M} \cdot \mathbf{H}_{F,\mathbf{0},0} + \mathbf{Z} \cdot \mathbf{H}_{F,\mathbf{0},1}, \tag{4}$$

where $\mathbf{h}_F$ is the hash of $\mathbf{y}$. Moreover, by applying Eq. (1) on the compressed encoding of $\mathbf{y}$ given by $\mathbf{M} \cdot \mathbf{K}_F - \mathbf{h}_F^\intercal \otimes \mathbf{G}_q$, we have that

$$\mathbf{M} \cdot \mathbf{K}_F \cdot \mathbf{P}_i = \mathbf{y}[i] \cdot \mathbf{G}_q + (\mathbf{M} \cdot \mathbf{K}_F - \mathbf{h}_F^\intercal \otimes \mathbf{G}_q) \cdot \mathbf{P}_i + \mathbf{Z} \cdot \mathbf{Q}_{i,\mathbf{y}}. \tag{5}$$

Putting Eq. (4) and Eq. (5) together, we obtain that

$$\mathbf{M} \cdot \mathbf{K}_F \cdot \mathbf{P}_i = \mathbf{y}[i] \cdot \mathbf{G}_q + \mathbf{M} \cdot \mathbf{H}_{i,0} + \mathbf{Z} \cdot \mathbf{H}_{i,1},$$

where $\mathbf{H}_{i,0}$ and $\mathbf{H}_{i,1}$ are low-norm matrices, and therefore

$$\mathbf{M} \cdot \mathbf{K}_F \cdot \begin{bmatrix} \mathbf{P}_{nx} & \mathbf{P}_{nx+1} & \cdots & \mathbf{P}_{nx+n-1} \end{bmatrix} \cdot \mathbf{r} = \underbrace{(\mathbf{y}_x^\intercal \otimes \mathbf{G}_q) \cdot \mathbf{r}}_{\mathbf{G}_q \cdot \mathbf{y}_x} + \mathbf{M} \cdot \mathbf{h}_0 + \mathbf{Z} \cdot \mathbf{h}_1,$$

where $\mathbf{h}_0$ and $\mathbf{h}_1$ are low-norm vectors. Plugging the above into Eq. (3) provides us with low-norm vectors $\mathbf{h}_0$, $\mathbf{h}_1$ and $\mathbf{h}_2$ such that

$$\mathbf{M} \cdot \mathbf{h}_0 + \mathbf{Z} \cdot \mathbf{h}_1 + \mathbf{M}' \cdot \mathbf{h}_2 = \mathbf{0}.$$

Since $\mathbf{M}$ and $\mathbf{M}'$ are random, it is not hard to prove that this adversary would break the SIS variant of decomposed LWE, i.e., decomposed SIS (DSIS).

An Issue with Compactness. Unfortunately, the construction described above has a fundamental problem, namely that the hash function is not compressing. Indeed, the norm of the SIS solution $\mathbf{h}_0, \mathbf{h}_1, \mathbf{h}_2$ increases proportionally to the space complexity of F. That means that in order to achieve correlation intractability we need $q > L$ and so $n = k \log q > \log L$. Designing correlation intractability for non-compressing functions is trivial: just use the constant function that outputs an element that is outside of the image of f.

[3] Here, $\mathbf{G}^{-1}(\mathbf{v})$ denotes the bit decomposition of the vector $\mathbf{v} \in \mathbb{Z}_q^k$. Notice that $\mathbf{G}_q \cdot \mathbf{G}^{-1}(\mathbf{v}) = \mathbf{v}$.

Nevertheless our construction has two desirable properties. (i) The first one is that hashing and generating the key require time $\mathsf{poly}(\lambda, \log L)$, so one can, in principle, instantiate our construction with a superpolynomial L, e.g., 2^ℓ. Notice that this would require the reduction to run in superpolynomial time (the evaluation of F requires time proportional to L), but in the technical section we show that using a recent variant of the compressed encodings [AMR26] we can actually homomorphically evaluate F in time polynomial in λ, $\log L$, and the size of a circuit computing f. (ii) The second interesting property is that the hash parameters can be chosen independently of the circuit size, for a fixed supported depth bound. For arbitrary f, there always exists a circuit of size $2^{O(\ell)} \cdot \mathsf{poly}(\lambda)$ that outputs its truth table. Therefore, assuming the subexponential hardness of DSIS, and choosing the lattice dimension sufficiently large relative to ℓ, we can prove that our hash function is correlation intractable even for searchable relations where f is inefficient. To make our result interesting, we just need to find a way to achieve compression.

Correlation Intractability for Batched Searchable Relations. To understand how to solve the issue, let us take a step back and compare with known approaches. Previous constructions of correlation intractability [PS19, LV22, BKM20] computationally hid the description of the function f in the key, so that, hashing x, the attacker would obliviously compute $f(x) + \mathsf{PRF}_K(x)$. Then, they argued that with overwhelming probability over K, 0 is not in the image of PRF_K. In such case, there exists no x such that $f(x) = \mathsf{Hash}(\mathsf{hk}, x) = f(x) + \mathsf{PRF}_K(x)$. Notice that this would intrinsically require the complexity of the hash function to grow proportionally to that of f.

Our approach instead differs in two ways. First of all, the reduction hides the description of f in the hash key *information theoretically*, and we rely on a computational argument to argue that finding an x such that $\mathsf{PRF}_K(x) = 0$ is hard. Second, we split the computation of $f(x)$ into two parts: an input-independent, inefficient preprocessing that is performed by the reduction and hidden in the hash key (this corresponds to the homomorphic computation of the truth table of f), and a lightweight computation that depends on x which is performed in an encrypted way by the hasher (this roughly corresponds to homomorphically reading the x-th entry of the truth table). This ensures that the complexity of the hash function is independent of that of f, indeed, there is only a polylogarithmic dependence on the time and space complexity of the computation that the reduction embeds in the key. Unfortunately, in the construction we illustrated, these scale proportionally to L, therefore exponentially in the length of the input x!

What happens, however, if we consider a different class of functions, where the evaluation of f on x can be split into an inefficient input-independent preprocessing with subexponential complexity in the length of the input, and a lightweight computation that depends on x, but not on f? Our recipe is likely to give a correlation intractable hash function for a non-trivial class of inefficiently searchable relations.

One such class is batched functions: suppose that x is split into c components $x_0, \ldots, x_{c-1}$ of equal length and we tackle the relation $\mathcal{R}_{f^c}$ where

$$f^c(x) := (f(x_0), \ldots, f(x_{c-1})).$$

Then, $f^c(x)$ can be computed as follows: in the preprocessing phase, we compute the truth table of f with exponential complexity in the length of a single component x_i; then, once x is provided, we simply read the entries in position $x_0, \ldots, x_{c-1}$. Observe that now that if c is sufficiently large, we can ensure that the size of the hash is strictly smaller than the length of the input. This is the key to achieve compression.

To summarise, if $f : [L/a] \to \{0,1\}^a$ and $a \cdot c = n$, our correlation intractable hash function for $\mathcal{R}_{f^c}$ becomes now

$$\mathbf{G}^{-1}(\mathbf{M} \cdot \begin{bmatrix} \mathbf{P}_{ax_0} & \cdots & \mathbf{P}_{ax_0+a-1} & \cdots & \mathbf{P}_{ax_{c-1}} & \cdots & \mathbf{P}_{ax_{c-1}+a-1} \end{bmatrix} \cdot \mathbf{r}).$$

The size of the input is now $c \cdot \log(L/a)$, whereas the output has size $n = \mathsf{poly}(\lambda, \log L)$. In the technical sections, we will generalise this approach to any batched searchable relation $\mathcal{R}_{f'}$, where

$$f'(x_0, \ldots, x_{c-1}) = (f_0(x_0), \ldots, f_{c-1}(x_{c-1}))$$

where $f_0, \ldots, f_{c-1} : [L/a] \to \{0,1\}^a$ are arbitrary functions. Furthermore, to improve the compression rate, we will add a layer of rounding after computing the hash.

On the Necessity of Decomposed SIS. The informed reader might wonder at this point whether using the LWE-based compressed encodings of [AMR25b] could be a viable strategy to obtain a similar result. Alas, this does not appear to work. Indeed, following the above reasoning (but with the updated key equations) one obtains a short integer solution for the structured matrix

$$\mathbf{I}_T \otimes \begin{bmatrix} \mathbf{M} & & & \\ & \mathbf{Z}' & & \\ & & \ddots & \\ & & & \mathbf{Z}' \end{bmatrix} + \Pi \left(\mathbf{I}_T \otimes \begin{bmatrix} \mathbf{G}_q & & & \\ & \mathbf{G}'_q & & \\ & & \ddots & \\ & & & \mathbf{G}'_q \end{bmatrix} \right)$$

where T is a parameter that depends on the complexity of the function describing the relation, $\mathbf{G}'_q = \mathsf{vec}(\mathbf{I}_t)^\intercal \otimes \mathbf{G}_q$, $\mathbf{Z}' := \mathbf{Z} + \mathbf{G}'_q$ and Π is the map that takes the last k rows on the input matrix and puts them on top switching their sign.

Correlation Intractability for All Sparse Batched Relations. Once we have correlation intractable hash functions for all batched searchable relations, generalising the result to all sufficiently sparse batched relations can be done via complexity leveraging.

The idea is that if the hash function is correlation intractable with respect to every single relation in a family, it should also be correlation intractable with respect to their union, but the advantage degrades proportionally to the number of elements in the family. Now, every batched relation $\mathcal{R}$ can be decomposed as the union of N batched (inefficiently) searchable relations, where N grows proportionally to

$$\max_x \Big| \{y | (x,y) \in \mathcal{R}\} \Big|.$$

If our construction for searchable relations were secure with advantage that is a negligible fraction of $1/N$, we would be done. Alas, in many cases, N is too big for this argument to work. That is why we rely on a subexponential PRF to further decrease N, following the example of [HLR21]. Specifically, the final construction is obtained by plugging the output of the correlation intractable hash function for batched searchable relations into a PRF (the key is public). We show that our construction from DSIS is correlation intractable for the relation

$$\mathcal{R}_{\mathsf{PRF}} := \{(x,z) | (x, \mathsf{PRF}(z)) \in \mathcal{R}\}.$$

The reason why this time the argument goes through is that many elements y such that $(x, y) \in \mathcal{R}$ will not belong to the image of the PRF, so, by carefully choosing the parameters, we can ensure that

$$\max_x \left| \{z | (x, z) \in \mathcal{R}_{\mathsf{PRF}}\} \right| \ll N.$$

At this point, it is a matter of tuning the parameters of DSIS and the PRF. The final result is that this can be done assuming that there exist constants $\alpha, \beta, \mu, \nu > 0$ such that $\min\{\mu, \beta\} + \nu > 1 + \alpha$, $\log B = \Theta(k^\beta)$, $m, \sigma \leq k^{O(1)}$, $\log q - \log B = \Theta(k^\alpha)$, and all adversaries that run in time 2^{k^μ} cannot break decomposed SIS with probability greater than 2^{-k^ν}, where k is the lattice dimension. Since standard SIS is conjectured to achieve this level of security, it is plausible that the same holds for decomposed SIS.

3 Preliminaries

Notation. For any positive integer n, we use $[n]$ to denote the set $\{0, 1, \ldots, n-1\}$. We adopt the convention in which indexes start from 0. All logarithms are in base 2 and, whenever $\log q$ is used as a dimension, it denotes $\lceil \log q \rceil$. All adversaries are allowed to be non-uniform.

We represent vectors using lower-case bold font, e.g., $\mathbf{v}$, whereas matrices will be denoted using upper-case bold font, e.g. $\mathbf{M}$. We use the standard mathematical notation for which vectors correspond to columns. Row vectors will be represented through transposition, e.g., $\mathbf{v}^\intercal$. We represent the i-th entry of a vector $\mathbf{v}$ by $\mathbf{v}[i]$.

We use $\lceil \cdot \rfloor$ to denote rounding to the closest integer. Given an element $x \in \mathbb{Z}_q$ and an integer $p < q$, we use $\lceil x \rfloor_p$ to denote an element $y \in \mathbb{Z}_p$ that minimises the absolute value of the centred representative of $x - y \cdot \lceil q/p \rceil \in \mathbb{Z}_q$, breaking ties according to the canonical representation of y. Notice that the minimum is at most $\frac{1}{2} \cdot \lceil q/p \rceil$. When applied to vectors, we assume that $\lceil \cdot \rfloor_p$ is computed entry by entry. We use $\mathsf{Trunc}_n : \{0,1\}^* \to \{0,1\}^n$ to denote the function that truncates the input string after the n-th entry. We use $\mathsf{Pad}_n : \{0,1\}^{\leq n} \to \{0,1\}^n$ to denote the function that pads the input string with zeros until it reaches length n. We often abuse notation, and we provide as input to Trunc_n and Pad_n objects that are not binary strings. In these situations, we mean that Trunc_n and Pad_n are provided with the canonical binary string representation of those objects.

We use χ_σ to denote the centred discrete Gaussian distribution over $\mathbb{Z}$ with probability mass proportional to $e^{-\pi x^2/\sigma^2}$, and use superscripts to denote product distributions. Samples used in matrices over $\mathbb{Z}_q$ are reduced modulo q. Let $\rho_{\sigma,q} := \max_{a \in \mathbb{Z}_q} \Pr_{x \xleftarrow{\$} \chi_\sigma}[x = a \bmod q]$. We will use the standard bounds $\max_x \Pr[\chi_\sigma = x] \leq 1/(\sigma - 1)$ and $\Pr_{x \xleftarrow{\$} \chi_\sigma}[|x| > \kappa\sigma] \leq 2e^{-\pi\kappa^2}$ for $\sigma > 1$.

We recall the definition of a pseudorandom function.

Definition 1 (Pseudorandom Function)**.** *A pseudorandom function with domain $\mathcal{X} = (\mathcal{X}_\lambda)_{\lambda \in \mathbb{N}}$ and range $\mathcal{Y} = (\mathcal{Y}_\lambda)_{\lambda \in \mathbb{N}}$ is a pair of PPT algorithms* $(\mathsf{Setup}, \mathsf{Eval})$ *with the following syntax:*

Setup**:** *This algorithm is randomised, takes as input the security parameter 1^λ and outputs a PRF key K.*

Eval**:** *This algorithm is deterministic, takes as input a key K and an input $x \in \mathcal{X}_\lambda$. The output is a value $y \in \mathcal{Y}_\lambda$.*

Let $s : \mathbb{N} \to \mathbb{N}$ and $\epsilon : \mathbb{N} \to [0,1]$ be functions of the security parameter. We say that the PRF is (s,ϵ)-secure if, for every PPT adversary $\mathcal{A}$ and every sufficiently large $\lambda \in \mathbb{N}$, it holds that

$$\left| \Pr\left[\mathcal{A}^F(1^\lambda, 1^s) = 1 \middle| \begin{matrix} K \xleftarrow{\$} \mathsf{Setup}(1^\lambda) \\ F \leftarrow \mathsf{Eval}(K, \cdot) \end{matrix} \right] - \Pr\left[\mathcal{A}^F(1^\lambda, 1^s) = 1 \middle| F \xleftarrow{\$} \mathcal{Y}_\lambda^{\mathcal{X}_\lambda} \right] \right| \leq \epsilon(\lambda).$$

4 Decomposed Short Integer Solution

In this section, we introduce the *decomposed short integer solution* assumption (DSIS). This is just the SIS version of decomposed LWE [AMR25a], and so its hardness is implied by that of DLWE. We start by defining the public-coin algorithm $\mathsf{DSample}$, which is used to generate the decomposed LWE matrices.

Definition 2 (DSample)**.** *The algorithm $\mathsf{DSample}$ takes as input positive integers $1^k, 1^m, 1^t$, q and σ, and performs the following operations:*

1. *For every $i \in [t]$, it samples $\mathbf{A}_i \xleftarrow{\$} \mathbb{Z}_q^{k \times m}$ and defines*

$$\mathbf{A} := \begin{bmatrix} \mathbf{A}_0 \\ \vdots \\ \mathbf{A}_{t-1} \end{bmatrix}$$

2. *It samples $\mathbf{B} \xleftarrow{\$} \chi_\sigma^{m \times tn}$ where $n := k \log q$.*

3. *It computes*

$$\mathbf{Z} := \begin{bmatrix} \mathbf{A}_0 \cdot \mathbf{B} - \mathbf{u}_0^\intercal \otimes \mathbf{G}_q & \cdots & \mathbf{A}_{t-1} \cdot \mathbf{B} - \mathbf{u}_{t-1}^\intercal \otimes \mathbf{G}_q \end{bmatrix}$$

   where $\mathbf{u}_i$ denotes the i-th element of the standard basis of $\mathbb{R}^t$.

The output consists of $\mathbf{A}, \mathbf{B}$ and $\mathbf{Z}$.

Throughout all sections, we will use the notation $n := k \log q$. Next, we formalise our assumption. This essentially states that finding short vectors in the kernel of the DLWE matrix $\mathbf{Z}$ is hard. We conjecture DSIS to achieve a similar level of security to standard SIS.

Definition 3 (Decomposed SIS)**.** *Let $k = k(\lambda), m = m(\lambda), t = t(\lambda)$, $q = q(\lambda)$, $\sigma = \sigma(\lambda)$, $s := s(\lambda)$, and $B = B(\lambda)$ be efficiently computable functions of the security parameter, where k, m and t are polynomially bounded. Let $\epsilon : \mathbb{N} \to [0,1]$ be another function.*

We say that the $(k, m, t, q, \sigma, B, s, \epsilon)$-DSIS assumption holds if, for every PPT adversary $\mathcal{A}$ and for every sufficiently large $\lambda \in \mathbb{N}$, we have

$$\Pr\left[\begin{matrix} 0 < \|\mathbf{w}\|_\infty \leq B \\ \mathbf{Z} \cdot \mathbf{w} = \mathbf{0} \end{matrix} \middle| \begin{matrix} (\mathbf{A}, \mathbf{B}, \mathbf{Z}) \xleftarrow{\$} \mathsf{DSample}(1^k, 1^m, 1^t, q, \sigma) \\ \mathbf{w} \xleftarrow{\$} \mathcal{A}(1^\lambda, 1^s, \mathbf{A}, \mathbf{B}, \mathbf{Z}) \end{matrix} \right] \leq \epsilon(\lambda).$$

Relation to Decomposed LWE. We recall that the decomposed LWE assumption states that it is hard to distinguish between an LWE sample with respect to the matrix $\mathbf{Z}$ and a random vector of appropriate dimension.

Definition 4 (Decomposed LWE [AMR25a])**.** *Let $k = k(\lambda), m = m(\lambda), t = t(\lambda)$, $q = q(\lambda)$, $\sigma = \sigma(\lambda)$, $\sigma' = \sigma'(\lambda)$ and $s := s(\lambda)$ be efficiently computable functions of the security parameter, where k, m and t are polynomially bounded. Let $\epsilon : \mathbb{N} \to [0,1]$ be another function.*

We say that the $(k, m, t, q, \sigma, \sigma', s, \epsilon)$-DLWE assumption holds if, for every PPT adversary $\mathcal{A}$ and every sufficiently large $\lambda \in \mathbb{N}$, we have

$$\left| \Pr\left[b = b' \middle| \begin{array}{l} b \xleftarrow{\$} \{0,1\} \\ \mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^k, \mathbf{e} \leftarrow \chi_{\sigma'}^{t^2 n} \\ (\mathbf{A}, \mathbf{B}, \mathbf{Z}) \xleftarrow{\$} \mathsf{DSample}(1^k, 1^m, 1^t, q, \sigma) \\ \mathbf{v}_0 \leftarrow \mathbf{Z}^\intercal \mathbf{r} + \mathbf{e}, \mathbf{v}_1 \xleftarrow{\$} \mathbb{Z}_q^{t^2 n} \\ b' \xleftarrow{\$} \mathcal{A}(1^\lambda, 1^s, \mathbf{A}, \mathbf{B}, \mathbf{Z}, \mathbf{v}_b) \end{array} \right] - \frac{1}{2} \right| \leq \epsilon(\lambda).$$

It is easy to see that DLWE implies DSIS as, given a short vector $\mathbf{w}$ such that $\mathbf{Z}\mathbf{w} = \mathbf{0}$, we can easily distinguish between a DLWE sample $\mathbf{v}_0$ and a random vector $\mathbf{v}_1$ by computing the inner product with $\mathbf{w}$. Indeed,

$$\mathbf{w}^\intercal \cdot \mathbf{v}_0 = \mathbf{w}^\intercal \mathbf{Z}^\intercal \mathbf{r} + \mathbf{w}^\intercal \mathbf{e} = \mathbf{w}^\intercal \mathbf{e}$$

will be small in norm. On the other hand, $\mathbf{w}^\intercal \mathbf{v}_1$ is uniformly random over $\mathbb{Z}_q$. This is formalised in the following theorem.

Theorem 3. *Suppose that q is a prime, $B < q$, and let $\kappa := \kappa(\lambda)$ be any function of the security parameter such that the expression below is positive. Assuming the hardness of $(k, m, t, q, \sigma, \sigma', s, \epsilon')$-DLWE, the $(k, m, t, q, \sigma, B, s, \epsilon)$-DSIS assumption holds for*

$$\epsilon' := \frac{1}{2} \cdot \epsilon \cdot \left(1 - 2t^2 n \cdot e^{-\pi\kappa^2} - \frac{2B \cdot \kappa \cdot \sigma' \cdot t^2 \cdot n + 1}{q} \right).$$

Extended Decomposed SIS. Next, we show that DSIS remains hard even if we extend the decomposed LWE matrix $\mathbf{Z}$ with an arbitrary number of random columns. This result implies that the hardness of DSIS implies that of SIS. We prove this result as [AMR25a] proved that DLWE remains hard even if the adversary is provided with additional random LWE samples using the same secret.

Theorem 4 (Extended DSIS)**.** *Let q be prime, let $r := r(\lambda)$ be polynomially bounded, and let $\kappa := \kappa(\lambda)$ be a function of the security parameter such that*

$$m \log(1/\rho_{\sigma,q}) \geq n + 2\kappa + 2\log r, \qquad n + 2\kappa + 2\log r \leq nt/2.$$

Assuming the hardness of $(k, m, t, q, \sigma, B, s, \epsilon)$-DSIS, for every PPT adversary $\mathcal{A}$ and every sufficiently large $\lambda \in \mathbb{N}$, we have

$$\Pr\left[\begin{array}{l} 0 < \|\mathbf{w}\|_\infty \leq B' \\ \begin{bmatrix} \mathbf{Z} & \mathbf{M} \end{bmatrix} \cdot \mathbf{w} = \mathbf{0} \end{array} \middle| \begin{array}{l} (\mathbf{A}, \mathbf{B}, \mathbf{Z}) \xleftarrow{\$} \mathsf{DSample}(1^k, 1^m, 1^t, q, \sigma) \\ \mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{k \times r} \\ \mathbf{w} \xleftarrow{\$} \mathcal{A}(1^\lambda, 1^s, \mathbf{A}, \mathbf{B}, \mathbf{Z}, \mathbf{M}) \end{array} \right] \leq \epsilon(\lambda) + O(2^{-\kappa}),$$

where

$$B' = B \cdot \frac{1}{r \cdot \frac{n+2\kappa+2\log r}{nt} + \sqrt{(\kappa+\log(nt))r} + 1}.$$

Proof. Consider the hybrid in which, instead of providing $\mathcal{A}$ with $\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{k\times r}$, we sample $\mathbf{R} \xleftarrow{\$} \mathsf{Ber}_p^{nt\times r}$ where $p = \frac{n+2\kappa+2\log r}{nt}$, and we set $\mathbf{M} \leftarrow \mathbf{Z}_0 \cdot \mathbf{R}$, where $\mathbf{Z}_0 = \mathbf{A}_0 \cdot \mathbf{B} - \mathbf{u}_0^\intercal \otimes \mathbf{G}_q$. Here, Ber_p denotes the Bernoulli distribution with mean p. We show that this hybrid is $O(2^{-\kappa})$-statistically indistinguishable from the original game by the leftover hash lemma.

Consider two distinct vectors $\mathbf{x}, \mathbf{x}' \in \{0,1\}^{nt}$ and let $\mathbf{d} := \mathbf{x} - \mathbf{x}'$. Fix an entry of $\mathbf{d}$ equal to 1 or -1. Conditioned on all other columns of $\mathbf{B}$, the probability that $\mathbf{Bd} = \mathbf{0}$ is at most $\rho_{\sigma,q}^m$. If $\mathbf{Bd} \neq \mathbf{0}$, then $\mathbf{A}_0\mathbf{Bd}$ is uniform over $\mathbb{Z}_q^k$. Therefore, the family $\mathbf{x} \mapsto \mathbf{Z}_0\mathbf{x}$ is $(q^{-k} + \rho_{\sigma,q}^m)$-universal over $\{0,1\}^{nt}$. On the other hand, each column of $\mathbf{R}$ has min-entropy at least

$$-\log((1-p)^{nt}) \geq n + 2\kappa + 2\log r.$$

The strong leftover hash lemma therefore shows that each column of $\mathbf{Z}_0\mathbf{R}$ is $O(2^{-\kappa}/r)$-statistically close to uniform even given $\mathbf{A}, \mathbf{B}$ and $\mathbf{Z}$. A hybrid over the r columns proves the claim.

If $\mathcal{A}$ manages to find low-norm vectors $\mathbf{w}_0$ and $\mathbf{w}_1$ such that

$$\mathbf{Z} \cdot \mathbf{w}_0 + \mathbf{M} \cdot \mathbf{w}_1 = \mathbf{0},$$

we obtain that

$$\mathbf{Z} \cdot (\mathbf{w}_0 + (\mathbf{u}_0 \otimes \mathbf{R}) \cdot \mathbf{w}_1) = \mathbf{0},$$

where $\mathbf{u}_0$ denotes the first vector of the standard basis of $\mathbb{R}^t$. By Hoeffding's inequality and a union bound over the nt rows of $\mathbf{R}$, with probability $1 - O(2^{-\kappa})$, we have

$$\|\mathbf{w}_0 + (\mathbf{u}_0 \otimes \mathbf{R}) \cdot \mathbf{w}_1\|_\infty \leq \|\mathbf{w}_0\|_\infty + \left(p \cdot r + \sqrt{(\kappa+\log(nt))r}\right) \cdot \|\mathbf{w}_1\|_\infty.$$

Thus, if $\|\mathbf{w}_0, \mathbf{w}_1\|_\infty \leq B'$, the extracted vector has norm at most B.

It remains to observe that this vector is non-zero except with probability $O(2^{-\kappa})$. This is immediate when $\mathbf{w}_1 = \mathbf{0}$, since then $\mathbf{w}_0 \neq \mathbf{0}$. Otherwise, conditioned on $\mathbf{M}$, all but one of the columns of $\mathbf{R}$, and the randomness of $\mathcal{A}$, the equality $\mathbf{w}_0 + (\mathbf{u}_0 \otimes \mathbf{R})\mathbf{w}_1 = \mathbf{0}$ uniquely determines any column of $\mathbf{R}$ whose coefficient in $\mathbf{w}_1$ is non-zero. Averaging over its revealed image under $\mathbf{Z}_0$, the conditional guessing probability of that column is at most

$$q^k \cdot 2^{-(n+2\kappa+2\log r)} \leq 2^{-2\kappa}/r^2.$$

A union bound over the r columns completes the reduction to $(k, m, t, q, \sigma, B, s, \epsilon)$-DSIS. □

5 Defining Correlation Intractability

In this section, we provide definitions for correlation intractable hash functions. We start with the original definition of [CGH98].

Definition 5 (Correlation Intractability)**.** *Let $\mathcal{R} := (\mathcal{R}_\lambda \subseteq \mathcal{X}_\lambda \times \mathcal{Y}_\lambda)_{\lambda\in\mathbb{N}}$ be a family of relations and let $s : \mathbb{N} \to \mathbb{N}$ and $\epsilon : \mathbb{N} \to [0,1]$ be functions of the security parameter. We say that a hash*

function $(\mathsf{Setup}, \mathsf{Hash})$ *is* (s, ϵ)*-correlation intractable with respect to* $\mathcal{R}$ *if, for every PPT adversary* $\mathcal{A}$ *and every sufficiently large* $\lambda \in \mathbb{N}$*, we have*

$$\Pr\left[(x,y) \in \mathcal{R}_\lambda \middle| \begin{array}{l} \mathsf{hk} \xleftarrow{\$} \mathsf{Setup}(1^\lambda) \\ x \xleftarrow{\$} \mathcal{A}(1^\lambda, 1^s, \mathsf{hk}) \\ y \leftarrow \mathsf{Hash}(\mathsf{hk}, x) \end{array}\right] \leq \epsilon(\lambda).$$

If s *and* ϵ *are left unspecified, we intend that* $s(\lambda) = \lambda$ *and* $\epsilon(\lambda) = 2^{-\omega(\log \lambda)}$*.*

Observe that if a hash function is correlation intractable for a relation $\mathcal{R}$, it is also correlation intractable for any relation $\mathcal{R}' \subseteq \mathcal{R}$.

Sparse Relations. Building a correlation intractable hash function for every relation is impossible. For instance, what if there exists an $x \in \mathcal{X}_\lambda$ such that $(x, y) \in \mathcal{R}_\lambda$ for every $y \in \mathcal{Y}_\lambda$? That is why we usually restrict the problem to sparse relations, i.e., relations for which every $x \in \mathcal{X}_\lambda$ is in relations with at most a negligible fraction of all elements in $\mathcal{Y}_\lambda$.

Definition 6 (Sparse Relation)**.** *Let* $\mathcal{R} := (\mathcal{R}_\lambda \subseteq \mathcal{X}_\lambda \times \mathcal{Y}_\lambda)_{\lambda \in \mathbb{N}}$ *be a family of relations and let* $\delta : \mathbb{N} \to [0, 1]$ *be a function of the security parameter. We say that* $\mathcal{R}$ *is* δ*-sparse if, for every sufficiently large* $\lambda \in \mathbb{N}$ *and* $x \in \mathcal{X}_\lambda$*,*

$$\Pr\left[(x,y) \in \mathcal{R}_\lambda \middle| y \xleftarrow{\$} \mathcal{Y}_\lambda\right] \leq \delta(\lambda).$$

Searchable Relations. In this work, we will focus on particular classes of relations. One example are searchable relations, i.e., relations for which, for every $x \in \mathcal{X}_\lambda$ there exists at most one $y \in \mathcal{Y}_\lambda$ such that $(x, y) \in \mathcal{R}_\lambda$.

Definition 7 (Searchable Relation)**.** *We say that a relation* $\mathcal{R} := (\mathcal{R}_\lambda \subseteq \mathcal{X}_\lambda \times \mathcal{Y}_\lambda)_{\lambda \in \mathbb{N}}$ *is searchable if, for every* $\lambda \in \mathbb{N}$*, we have*

$$\forall x \in \mathcal{X}_\lambda,\ y, y' \in \mathcal{Y}_\lambda : \qquad (x,y), (x,y') \in \mathcal{R}_\lambda \implies y = y'.$$

Building (non-uniform) correlation intractable hash functions for a searchable relation where $|\mathcal{Y}_\lambda| > |\mathcal{X}_\lambda|$ is trivial. Indeed, there exists a $\overline{y} \in \mathcal{Y}_\lambda$ that is not in relation with any element of $\mathcal{X}_\lambda$. We can therefore consider the hash function that simply outputs $\overline{y}$.

Observe that any sequence of (not necessarily efficiently computable) functions $f := (f_\lambda : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda)_{\lambda \in \mathbb{N}}$ implicitly defines a searchable relation $\mathcal{R}_f := (\mathcal{R}_{f_\lambda})_{\lambda \in \mathbb{N}}$, where

$$\mathcal{R}_{f_\lambda} := \{(x,y) \in \mathcal{X}_\lambda \times \mathcal{Y}_\lambda | y = f_\lambda(x)\}.$$

This will be important in Section 6. Notice that if a hash function $(\mathsf{Setup}, \mathsf{Hash})$ is correlation intractable for some relation $\mathcal{R}_{f'}$, where $f'(x) := \mathsf{Trunc}_\ell(f(x))$ for any $\ell(\lambda)$, we can trivially build a correlation intractable hash function for $\mathcal{R}_f$ by simply padding $\mathsf{Hash}(\mathsf{hk}, x)$ to match the output size of f.

Batched Relations. In this work, we also focus on batched relations, i.e., relations $\mathcal{R}$ that are direct products of other relations. Specifically,

$$((x_0, \ldots, x_{c-1}), (y_0, \ldots, y_{c-1})) \in \mathcal{R}$$

if and only if $(x_i, y_i) \in \mathcal{R}^{(i)}$ for all $i \in [c]$. This type of relations is common to describe the *bad-challenge* function of Σ-protocols, where parallel repetitions are used for soundness amplification.

Definition 8 (Batched Relation)**.** *Let $c = c(\lambda)$ be a polynomially bounded positive integer. A c-batched relation consists of a relation*

$$\mathcal{R} := (\mathcal{R}_\lambda^{(0)} \times \cdots \times \mathcal{R}_\lambda^{(c(\lambda)-1)})_{\lambda \in \mathbb{N}},$$

where, for every $i \in \mathbb{N}$, $\mathcal{R}^{(i)} := (\mathcal{R}_\lambda^{(i)} \subseteq \mathcal{X}_\lambda \times \mathcal{Y}_\lambda)_{\lambda \in \mathbb{N}}$ is a relation.

For ease of notation, we often simply write

$$\mathcal{R} := \mathcal{R}^{(0)} \times \cdots \times \mathcal{R}^{(c-1)}.$$

Moreover, if $\mathcal{R}^{(0)} = \cdots = \mathcal{R}^{(c-1)} = \mathcal{R}'$, we simply write $\mathcal{R} := \mathcal{R}'^c$.

Notice that batched relations are product relations [HLR21], but not all product relations are batched relations. For example, all searchable relations are product relations, however, there exist searchable relations that are not c-batched for $c > 1$. We also observe that if $\mathcal{R}^{(i)}$ is searchable for every $i \in [c]$, then $\mathcal{R}$ is also searchable.

In the following lemma, we observe that any c-batched relation

$$\mathcal{R} = \mathcal{R}^{(0)} \times \cdots \times \mathcal{R}^{(c-1)}$$

is contained in the union of L^c-many c-batched searchable relations, where L depends on the maximum sparsity of $\mathcal{R}^{(0)}, \ldots, \mathcal{R}^{(c-1)}$. Looking ahead, we will use this result to upgrade a correlation intractable hash function for batched searchable relations to one for all batched relations.

Lemma 1. *Let $c = c(\lambda)$ be a polynomially bounded positive integer and let $L = L(\lambda)$ be a positive integer-valued function of the security parameter. For every $i \in \mathbb{N}$, let $\mathcal{R}^{(i)} := (\mathcal{R}_\lambda^{(i)} \subseteq \mathcal{X}_\lambda \times \mathcal{Y}_\lambda)_{\lambda \in \mathbb{N}}$ be a relation such that, for every $\lambda \in \mathbb{N}$,*

$$\max_{x \in \mathcal{X}_\lambda} \left| \left\{ y \in \mathcal{Y}_\lambda \middle| (x, y) \in \mathcal{R}_\lambda^{(i)} \right\} \right| \leq L(\lambda).$$

Let $\mathcal{R} = (\mathcal{R}_\lambda)_{\lambda \in \mathbb{N}}$ be the c-batched relation $\mathcal{R}^{(0)} \times \cdots \times \mathcal{R}^{(c-1)}$.

Then, there exist c-batched searchable *relations $\left(\mathcal{R}'^{(j)}\right)_{j \in \mathbb{N}}$ such that, for every $\lambda \in \mathbb{N}$, we have*

$$\mathcal{R}_\lambda \subseteq \bigcup_{j \in [L(\lambda)^{c(\lambda)}]} \mathcal{R}'^{(j)}_\lambda.$$

Proof. For every $i, \lambda \in \mathbb{N}$ and $x \in \mathcal{X}_\lambda$, define an ordering of the elements of

$$\mathcal{R}_\lambda^{(i)}[x] := \left\{ y \in \mathcal{Y}_\lambda \middle| (x, y) \in \mathcal{R}_\lambda^{(i)} \right\}.$$

Let them be $y_{i,x,\lambda}^{(j)}$ for $j \in [L_{i,x,\lambda}]$. Notice that, for every $i, \lambda \in \mathbb{N}$ and $x \in \mathcal{X}_\lambda$, we have that $L_{i,x,\lambda} \leq L(\lambda)$. Next, for every $i, j, \lambda \in \mathbb{N}$, define the set

$$\mathcal{R}_\lambda^{(i,j)} := \left\{ (x,y) \in \mathcal{X}_\lambda \times \mathcal{Y}_\lambda \middle| j < L_{i,x,\lambda} \ \wedge \ y = y_{i,x,\lambda}^{(j)} \right\}.$$

Notice that $(\mathcal{R}_\lambda^{(i,j)})_{\lambda \in \mathbb{N}}$ is a searchable relation.

For every $\lambda \in \mathbb{N}$ and every sequence $j_0, \ldots, j_{c(\lambda)-1} \in [L(\lambda)]$, define now the set $\mathcal{R}'^{(j)}_\lambda$ for $j := \sum_{i \in [c(\lambda)]} j_i \cdot L(\lambda)^i$ as follows

$$\mathcal{R}'^{(j)}_\lambda := \mathcal{R}_\lambda^{(0,j_0)} \times \cdots \times \mathcal{R}_\lambda^{(c(\lambda)-1, j_{c(\lambda)-1})}.$$

Clearly, for every $j \in \mathbb{N}$, we have that $\mathcal{R}'^{(j)} := (\mathcal{R}'^{(j)}_\lambda)_{\lambda \in \mathbb{N}}$ is c-batched and searchable. Moreover, we can easily verify that

$$\mathcal{R}_\lambda \subseteq \bigcup_{j \in [L(\lambda)^{c(\lambda)}]} \mathcal{R}'^{(j)}_\lambda.$$

Indeed, any element of $\mathcal{R}_\lambda$ is of the form

$$\left((x_0, \ldots, x_{c(\lambda)-1}), \left(y_{0,x_0,\lambda}^{(j_0)}, \ldots, y_{c(\lambda)-1, x_{c(\lambda)-1}, \lambda}^{(j_{c(\lambda)-1})} \right) \right)$$

for some integers $j_0, \ldots, j_{c(\lambda)-1} \in [L(\lambda)]$. Observe that such an element is contained in $\mathcal{R}'^{(j)}_\lambda$ where $j := \sum_{i \in [c(\lambda)]} j_i \cdot L(\lambda)^i < L(\lambda)^{c(\lambda)}$. □

6 Correlation Intractability for All Sparse Batched Relations

In this section, we explain how to build correlation intractable hash functions for all sparse batched relations from the hardness of decomposed SIS. We will start building correlation intractable hash functions for *all* batched searchable relations, even non-efficiently computable ones. Next, we generalise this construction to non-searchable batched relations.

6.1 The Construction for Batched Searchable Relations

Our construction relies on the tree encodings recently introduced by Abram, Malavolta and Roy [AMR26]. The main properties of these encodings are summarised by the following theorem. We refer to their work [AMR26] for additional details about their construction.

Theorem 5. *Let $\mathcal{F} = (\mathcal{F}_\lambda)_{\lambda \in \mathbb{N}}$ be a family of functions, where each $f \in \mathcal{F}_\lambda$ maps an $\ell = \ell(\lambda)$-bit input to an r-bit output. Assume $n \geq 2$. Let $\kappa := \kappa(\lambda)$ be a function of the security parameter. Suppose that each $f \in \mathcal{F}_\lambda$ is computable by a fan-in-two Boolean circuit of depth at most $d = d(\lambda) \geq 1$. Then, there exists a tuple of PPT algorithms* $(\mathsf{Setup}, \mathsf{Eval}, \mathsf{EvalH}, \mathsf{Expand})$ *with the following properties*

- Setup *takes as input the security parameter 1^λ and, for $t = 2mn$, outputs*

$$(\mathbf{A}, \mathbf{B}, \mathbf{Z}) \xleftarrow{\$} \mathsf{DSample}(1^k, 1^m, 1^t, q, \sigma).$$

- Eval *is deterministic and takes as input the output of* Setup*, and the circuit representing a function* $f \in \mathcal{F}_\lambda$*. The output is a matrix*

$$\mathbf{M}_f \in \mathbb{Z}_q^{k \times rmn^4}.$$

- EvalH *is deterministic and takes as input the output of* Setup*, the circuit representing a function* $f \in \mathcal{F}_\lambda$*, and a value* $\mathbf{x} \in \{0,1\}^\ell$*. The output is a matrix* $\mathbf{H}_{f,\mathbf{x}} \in \mathbb{Z}_q^{nt^2 \times rn}$*, where, simultaneously for all* $\mathbf{x} \in \{0,1\}^\ell$*,*

$$\|\mathbf{H}_{f,\mathbf{x}}\|_\infty \leq (mnt\sigma\sqrt{\kappa+\ell})^{O(\ell d)}$$

with probability at least $1 - 2^{-2\kappa}$ *over the randomness of* Setup*.*

- Expand *is deterministic and takes as input the output of* Setup*, and a value* $\mathbf{x} \in \{0,1\}^\ell$*. The output is a matrix* $\mathbf{P}_\mathbf{x} \in \mathbb{Z}_q^{rmn^4 \times rn}$*, where*

$$\|\mathbf{P}_\mathbf{x}^\intercal\|_{\infty,\infty} \leq n(nm)^\ell.$$

Moreover, for every $f \in \mathcal{F}_\lambda$ *and every* $\mathbf{x} \in \{0,1\}^\ell$*, it holds that*

$$\mathbf{M}_f \cdot \mathbf{P}_\mathbf{x} - f(\mathbf{x})^\intercal \otimes \mathbf{G}_q = \mathbf{Z} \cdot \mathbf{H}_{f,\mathbf{x}},$$

where $\mathbf{M}_f \leftarrow \mathsf{Eval}(\mathsf{pk}, f)$*,* $\mathbf{H}_{f,\mathbf{x}} \leftarrow \mathsf{EvalH}(\mathsf{pk}, f, \mathbf{x})$*,* $\mathbf{P}_\mathbf{x} \leftarrow \mathsf{Expand}(\mathsf{pk}, \mathbf{x})$*, and* $\mathsf{pk} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$*.*

Use the key-equation construction of [AMR26] with output arity r, defining

$$\begin{aligned}\mathbf{M}_f &= \mathbf{G}_q\mathsf{TreeEvalM}(\mathsf{pk}, f),\\ \mathbf{P}_\mathbf{x} &= \mathsf{TreeExpand}(\mathsf{pk}, 1^r, \mathbf{x}),\\ \mathbf{H}_{f,\mathbf{x}} &= \mathsf{TreeEvalH}(\mathsf{pk}, f, \mathbf{x}).\end{aligned}$$

Boolean circuits translate into bounded arithmetic circuits with constant-factor depth overhead. The construction and bounds also hold for prime q, using binary gadget preimages with $n = k\lceil \log q \rceil$ and $\mathbf{G}_q\mathbf{G}_q^{-1}(\mathbf{X}) = \mathbf{X}$. The simultaneous norm bound follows by applying the pointwise bound of [AMR26] with parameter $\kappa + \ell$ and taking a union bound over $\mathbf{x} \in \{0,1\}^\ell$. The main observation on which our correlation intractable hash function relies is that the running time of Expand is independent of the complexity of f, except for its depth d.

Before presenting our construction, we state the following simple fact.

Lemma 2. *There exists a deterministic polynomial time algorithm that, on input* 1^a*,* 1^c*,* p *with* $1 < p < q$ *such that* $ac \geq k \log p$ *and a matrix* $\mathbf{Q} \in \mathbb{Z}_q^{k \times k}$*, outputs a vector* $\mathbf{g}_{a,c,p}(\mathbf{Q}) \in \mathbb{Z}_q^{c^2 an}$ *such that* $\|\mathbf{g}_{a,c,p}(\mathbf{Q})\|_{\infty,\infty} \leq k \log q$ *and, for every vector* $\mathbf{u} \in \{0,1\}^{c^2 a}$*, we have that*

$$(\mathbf{u}^\intercal \otimes \mathbf{G}_q) \cdot \mathbf{g}_{a,c,p}(\mathbf{Q}) = \left\lceil \frac{q}{p} \right\rceil \cdot \mathbf{Q} \cdot \mathbf{G}_p \cdot \mathbf{v},$$

where

$$\forall i \in [k \log p]: \qquad \mathbf{v}[i] = \mathbf{u}\left[ac \cdot \left\lfloor \frac{i}{a} \right\rfloor + i\right]$$

Essentially, this states that there exists a low-norm vector $\mathbf{g}_{a,c,p}(\mathbf{Q})$ such that, given any other vector

$$\mathbf{u}^\intercal := \begin{bmatrix}\mathbf{u}_{0,0}^\intercal & \cdots & \mathbf{u}_{0,c-1}^\intercal & \cdots & \mathbf{u}_{c-1,0}^\intercal & \cdots & \mathbf{u}_{c-1,c-1}^\intercal\end{bmatrix}$$

where $\mathbf{u}_{i,j} \in \{0,1\}^a$ for every $i, j \in [c]$, it holds that

$$(\mathbf{u}^\intercal \otimes \mathbf{G}_q) \cdot \mathbf{g}_{a,c,p}(\mathbf{Q}) = \left\lceil \frac{q}{p} \right\rceil \cdot \mathbf{Q} \cdot \mathbf{G}_p \cdot \mathbf{v}$$

where $\mathbf{v}$ consists of the first $k \log p$ entries of

$$\begin{bmatrix}\mathbf{u}_{0,0}^\intercal & \mathbf{u}_{1,1}^\intercal & \cdots & \mathbf{u}_{c-1,c-1}^\intercal\end{bmatrix}.$$

The Construction. Let $1 < p < q$. Suppose that a and c are natural numbers such that $ac \geq k \log p$. We are finally ready to introduce our correlation intractable hash function for any batched searchable relation

$$\mathcal{R} := \mathcal{R}_{f^{(0)}} \times \cdots \times \mathcal{R}_{f^{(c-1)}}$$

where $f^{(i)} : \{0,1\}^\ell \to \{0,1\}^a$ for every $i \in [c]$. Notice that none of the functions $f^{(i)}$ need to be efficiently computable. Instantiate the encodings of Theorem 5 with $r = ac$.

$\mathsf{Setup}(1^\lambda)$**:** Compute $\mathbf{M} \xleftarrow{\$} \mathbb{Z}_q^{k \times camn^4}$, $\mathbf{Q} \xleftarrow{\$} \mathsf{GL}_k(\mathbb{Z}_q)$, $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^k$ and

$$(\mathbf{A}, \mathbf{B}, \mathbf{Z}) \xleftarrow{\$} \mathsf{DSample}(1^k, 1^m, 1^t, q, \sigma).$$

Then, output $\mathsf{hk} := (\mathbf{A}, \mathbf{B}, \mathbf{Z}, \mathbf{M}, \mathbf{Q}, \mathbf{r})$.

$\mathsf{Hash}(\mathsf{hk}, \mathbf{x})$**:** Rewrite $\mathbf{x} \in \{0,1\}^{c \cdot \ell}$ as

$$(\mathbf{x}_0, \ldots, \mathbf{x}_{c-1}), \qquad \forall i \in [c] : \quad \mathbf{x}_i \in \{0,1\}^\ell.$$

Then, for every $i \in [c]$, compute $\mathbf{P}_{\mathbf{x}_i} \leftarrow \mathsf{Expand}(\mathbf{A}, \mathbf{B}, \mathbf{Z}, \mathbf{x}_i)$. Finally, derive

$$\mathbf{y} \leftarrow \left\lfloor \mathbf{M} \cdot \begin{bmatrix}\mathbf{P}_{\mathbf{x}_0}, \ldots, \mathbf{P}_{\mathbf{x}_{c-1}}\end{bmatrix} \cdot \mathbf{g}_{a,c,p}(\mathbf{Q}) + \mathbf{r} \right\rfloor_p.$$

Output the bit representation of $\mathbf{y}$ padding it with zeros until it has bit-length ca.

The following theorem summarises the security property of our construction.

Theorem 6. *Let $k := k(\lambda)$, $m := m(\lambda)$, $\ell := \ell(\lambda), a := a(\lambda)$, $d := d(\lambda) \geq 1$, $\kappa := \kappa(\lambda)$ and $c := c(\lambda)$ be polynomial functions of the security parameter. Let $q := q(\lambda)$ be prime and let $p := p(\lambda)$, $\sigma := \sigma(\lambda)$ and $s := s(\lambda)$ be positive integers such that $1 < p < q$ and $k \log p \leq a \cdot c$. Let $\epsilon : \mathbb{N} \to [0,1]$ be a function of the security parameter. Define $n := k \log q$, $t := 2mn$, let $\phi_q(\kappa)$ be the smallest integer j such that*

$$\left(\Pr[\mathbf{Q} \notin \mathsf{GL}_k(\mathbb{Z}_q) \mid \mathbf{Q} \xleftarrow{\$} \mathbb{Z}_q^{k \times k}]\right)^j \leq 2^{-\kappa},$$

and set $r := acmn^4 + n + 2\kappa + k \cdot \phi_q(\kappa)$. Suppose that

$$m \log(1/\rho_{\sigma,q}) \geq n + 2\kappa + 2 \log r, \qquad n + 2\kappa + 2 \log r \leq nt/2.$$

Suppose that

$$B \geq \left(mnt\sigma\sqrt{\kappa+\ell}\right)^{\alpha\ell d}$$

for a sufficiently large constant $\alpha > 0$, and suppose that

$$\left\lceil \frac{q}{p} \right\rceil \leq \frac{2B}{r \cdot \frac{n+2\kappa+2\log r}{nt} + \sqrt{(\kappa+\log(nt))r}+1}.$$

Let $\mathcal{F} := (\mathcal{F}_\lambda)_{\lambda\in\mathbb{N}}$ be the function class where $\mathcal{F}_\lambda$ contains all functions

$$f : \{0,1\}^{\ell(\lambda)} \to \{0,1\}^{a(\lambda)}$$

that are computable by a fan-in-two Boolean circuit of depth at most $d(\lambda)$ and size at most $s(\lambda)$. Then, assuming the hardness of $(k, m, t, q, \sigma, B, s, \epsilon)$-DSIS, the above hash function is $(s, \epsilon+O(2^{-\kappa}))$-correlation intractable for any batched searchable relation

$$\mathcal{R} := \left(\mathcal{R}_{f_\lambda^{(0)}} \times \cdots \times \mathcal{R}_{f_\lambda^{(c-1)}}\right)_{\lambda\in\mathbb{N}}$$

where $f_\lambda^{(0)}, \ldots, f_\lambda^{(c(\lambda)-1)} \in \mathcal{F}_\lambda$ for every $\lambda \in \mathbb{N}$.

Proof. Suppose that there exist a sequence of functions $(f_\lambda^{(0)}, \ldots, f_\lambda^{(c-1)})_{\lambda\in\mathbb{N}}$ in $\mathcal{F}_\lambda$ and a PPT adversary $\mathcal{A}$ that, given 1^λ, 1^s and $\mathsf{hk} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$, manages to find $(\mathbf{x}_0, \ldots, \mathbf{x}_{c-1}) \in \{0,1\}^{c\ell}$ such that

$$\mathsf{Hash}(\mathsf{hk}, (\mathbf{x}_0, \ldots, \mathbf{x}_{c-1})) = (f_\lambda^{(0)}(\mathbf{x}_0), \ldots, f_\lambda^{(c-1)}(\mathbf{x}_{c-1}))$$

with probability greater than $\epsilon + O(2^{-\kappa})$.

We consider the following hybrid game.

Hybrid. Sample

$$\mathbf{Q} \xleftarrow{\$} \mathsf{GL}_k(\mathbb{Z}_q), \quad \mathbf{U} \xleftarrow{\$} \mathbb{Z}_q^{k\times(n+2\kappa)}, \quad \mathbf{h} \xleftarrow{\$} \mathbb{Z}_2^{n+2\kappa},$$
$$(\mathbf{A}, \mathbf{B}, \mathbf{Z}) \xleftarrow{\$} \mathsf{DSample}(1^k, 1^m, 1^t, q, \sigma).$$

Then, compute

$$\mathbf{M}_f \leftarrow \mathsf{Eval}(\mathbf{A}, \mathbf{B}, \mathbf{Z}, f),$$

where

$$\begin{aligned} f : \{0,1\}^\ell &\longrightarrow \{0,1\}^{ca} \\ \mathbf{x} &\longmapsto (f^{(0)}(\mathbf{x}), \ldots, f^{(c-1)}(\mathbf{x})) \end{aligned}$$

Finally, sample $\mathbf{M}' \xleftarrow{\$} \mathbb{Z}_q^{k\times acmn^4}$, and provide the adversary with the hash key

$$\mathsf{hk} = \left(\mathbf{A}, \mathbf{B}, \mathbf{Z}, \mathbf{Q}^{-1} \cdot (\mathbf{M}_f + \mathbf{M}'), \mathbf{Q}, \mathbf{Q}^{-1} \cdot \mathbf{U} \cdot \mathbf{h}\right).$$

Notice that the challenger now runs in time $\mathsf{poly}(s, \lambda)$.

Observe that this hybrid is $O(2^{-\kappa})$-statistically indistinguishable from the original game by the leftover hash lemma. So, the probability that the adversary manages to find $(\mathbf{x}_0, \ldots, \mathbf{x}_{c-1}) \in \{0,1\}^{c\ell}$ such that

$$\mathsf{Hash}(\mathsf{hk}, (\mathbf{x}_0, \ldots, \mathbf{x}_{c-1})) = (f_\lambda^{(0)}(\mathbf{x}_0), \ldots, f_\lambda^{(c-1)}(\mathbf{x}_{c-1}))$$

changes by at most $O(2^{-\kappa})$. We show that that would break the DSIS assumption.

Indeed, define $\mathbf{v} \in \mathbb{Z}_p^k$, whose bit decomposition is equal to

$$\mathsf{Trunc}_{k \log p} \begin{bmatrix} f_\lambda^{(0)}(\mathbf{x}_0) \\ \vdots \\ f_\lambda^{(c-1)}(\mathbf{x}_{c-1}) \end{bmatrix}$$

Now, define

$$\begin{aligned} \mathbf{w}_0 &:= \left[\mathbf{P}_{\mathbf{x}_0}, \ldots, \mathbf{P}_{\mathbf{x}_{c-1}}\right] \cdot \mathbf{g}_{a,c,p}(\mathbf{Q}) \\ \mathbf{w}_1 &:= \left[\mathbf{H}_{f,\mathbf{x}_0} \quad \ldots \quad \mathbf{H}_{f,\mathbf{x}_{c-1}}\right] \cdot \mathbf{g}_{a,c,p}(\mathbf{Q}) \end{aligned}$$

where $\mathbf{H}_{f,\mathbf{x}_i} \leftarrow \mathsf{EvalH}(\mathbf{A}, \mathbf{B}, \mathbf{Z}, f, \mathbf{x}_i)$. Observe that with probability at least $1 - 2^{-\kappa}$, we have

$$\begin{aligned} \|\mathbf{w}_0\|_\infty &\leq (mn)^{O(\ell)}, \\ \|\mathbf{w}_1\|_\infty &\leq \left(mnt\sigma\sqrt{\kappa + \ell}\right)^{O(\ell d)}. \end{aligned}$$

Observe that with the new notation

$$\mathsf{Hash}(\mathsf{hk}, (\mathbf{x}_0, \ldots, \mathbf{x}_{c-1})) = \mathsf{Pad}_{ac}(\lceil \mathbf{Q}^{-1} \cdot (\mathbf{M}' + \mathbf{M}_f) \cdot \mathbf{w}_0 + \mathbf{Q}^{-1} \cdot \mathbf{U} \cdot \mathbf{h} \rfloor_p).$$

We infer that if $\mathsf{Hash}(\mathsf{hk}, (\mathbf{x}_0, \ldots, \mathbf{x}_{c-1})) = (f_\lambda^{(0)}(\mathbf{x}_0), \ldots, f_\lambda^{(c-1)}(\mathbf{x}_{c-1}))$, it holds that

$$\mathbf{Q}^{-1} \cdot (\mathbf{M}' + \mathbf{M}_f) \cdot \mathbf{w}_0 + \mathbf{Q}^{-1} \cdot \mathbf{U} \cdot \mathbf{h} = \left\lceil \frac{q}{p} \right\rfloor \cdot \mathbf{v} + \mathbf{e},$$

where $\|\mathbf{e}\|_\infty \leq \frac{1}{2} \cdot \left\lceil \frac{q}{p} \right\rfloor$, and so

$$\mathbf{M}_f \cdot \mathbf{w}_0 - \left\lceil \frac{q}{p} \right\rfloor \cdot \mathbf{Q} \cdot \mathbf{v} = -\mathbf{M}' \cdot \mathbf{w}_0 + \mathbf{Q} \cdot \mathbf{e} - \mathbf{U} \cdot \mathbf{h}.$$

We observe that, by Theorem 5,

$$\begin{aligned} \mathbf{Z} \cdot \mathbf{w}_1 &= \mathbf{M}_f \cdot \mathbf{w}_0 - \left(\begin{bmatrix} f(\mathbf{x}_0) \\ \ldots \\ f(\mathbf{x}_{c-1}) \end{bmatrix}^\intercal \otimes \mathbf{G}_q \right) \cdot \mathbf{g}_{a,c,p}(\mathbf{Q}) \\ &= \mathbf{M}_f \cdot \mathbf{w}_0 - \left\lceil \frac{q}{p} \right\rfloor \cdot \mathbf{Q} \cdot \mathbf{v} \\ &= -\mathbf{M}' \cdot \mathbf{w}_0 + \mathbf{Q} \cdot \mathbf{e} - \mathbf{U} \cdot \mathbf{h}. \end{aligned}$$

This violates the hardness of Extended DSIS (see Theorem 4). We use the first $acmn^4$ columns of the matrix we are provided by the challenger in addition to $\mathbf{A}, \mathbf{B}$ and $\mathbf{Z}$, to define $\mathbf{M}'$. Next, we

use the following $n + 2\kappa$ columns to define $\mathbf{U}$. We use the rest to define $\mathbf{Q}$. Specifically, we split the remaining columns into $\phi_q(\kappa)$ blocks in $\mathbb{Z}_q^{k\times k}$. Notice that $\phi_q(\kappa) = O(\kappa)$. With probability at least $1 - O(2^{-\kappa})$, at least one of these will be an invertible matrix, we pick that to be $\mathbf{Q}$ (if there are multiple ones, we pick the first). Now, suppose that $\mathbf{Q}$ consists of the i-th block. Then, the vector

$$\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_0 \\ \mathbf{h} \\ -\mathbf{u}_i \otimes \mathbf{e} \end{bmatrix}$$

where $\mathbf{u}_i$ denotes the i-th element of the standard basis of $\mathbb{R}^{\phi_q(\kappa)}$, consists of a solution for the provided DSIS challenge. By the choice of α and the displayed bound on $\lceil q/p \rceil$, its norm is within the bound required by Theorem 4. Moreover, this solution is different from $\mathbf{0}$ with probability at least $1 - 2^{-(n+2\kappa)}$ as $\mathbf{h}$ is sampled at random over $\mathbb{Z}_2^{n+2\kappa}$. □

Observe also that all functions $f : \{0,1\}^\ell \to \{0,1\}^a$ can be computed by a circuit of size $a \cdot 2^\ell \cdot \mathsf{poly}(\lambda)$ and multiplicative depth at most $O(\log(2\ell))$. To see why, just consider the circuit that has the truth table of f hardcoded, and on input $x \in \{0,1\}^\ell$, it derives the 2^ℓ-dimensional unit vector having 1 at position x in multiplicative depth $\log \ell$ (to compute each entry of the unit vector, we need ℓ multiplications, which we can arrange in a tree of depth $\log \ell$). Then, the circuit outputs the inner product between this unit vector and the truth table of f. Let $K = mnt\sigma\sqrt{\kappa + \ell}$. Each indicator has total arithmetic depth $O(\log(2\ell))$, so its correction norm is at most $E = K^{O(\ell \log(2\ell))}$. Addition adds the two correction matrices, and flattening contributes at most $F = K^{O(\ell)}$ per gate. Thus summing at most 2^ℓ indicators gives correction norm at most

$$2^\ell E + (2^\ell - 1)F = K^{O(\ell \log(2\ell))}.$$

The indicators are disjoint, so their partial sums remain Boolean. This proves the norm estimate needed in the proof of Theorem 6, despite the final additions increasing total depth. The following corollary follows with $s = 2^\ell$, $\kappa = \Theta(n)$ chosen small enough for the entropy conditions, and the smallest p satisfying the rounding condition.

Corollary 1. *Let $\sigma \geq 3$ and $B \geq 2^{\alpha \cdot \ell \cdot \log(2\ell) \cdot (\log\lambda + \log\sigma)}$ for a sufficiently large constant $\alpha > 0$. Choose suitable parameters where $q > B$ is prime and $\rho_{\sigma,q} \leq 1/(\sigma - 1)$. Let $n := k \log q$ and suppose $m \geq 2k \log q$. Set $t = 2mn$ and $s = 2^\ell$.*

Assuming the hardness of $(k, m, t, q, \sigma, B, s, \epsilon)$-DSIS, there exists a hash function that is $(s, \epsilon + 2^{-\Omega(n)})$-correlation intractable for any c-batched searchable relation $\mathcal{R} \subseteq (\{0,1\}^\ell)^c \times (\{0,1\}^a)^c$ provided that $ac \geq k \cdot (\log q - \log B + O(\log(mnt)))$ and $a = \mathsf{poly}(\lambda)$.

Moreover, the hash key is unstructured and has size $a \cdot \mathsf{poly}(\ell, \lambda)$.

6.2 The Construction for All Sparse Batched Relations

In this section, we show how to build correlation intractable hash functions for any sparse batched relation.

We start by proving the following lemma, which states that if any hash function is correlation intractable for a family of relations, it is also correlation intractable for the union of all these relations. There is, however, a security loss proportional to the size of the family. Looking ahead, we will use lemma to upgrade our correlation intractable hash function for batched searchable relation to one for any sparse batched relation.

Lemma 3. *Let $(\mathcal{R}^{(i)})_{i\in\mathbb{N}}$ be a family of relations and let $L := L(\lambda)$ be a (not-necessarily polynomial) positive integer-valued function of the security parameter. Suppose that, for every sequence $i(\lambda) \in [L(\lambda)]$, $(\mathsf{Setup}, \mathsf{Hash})$ is a (s,ϵ)-correlation intractable hash function for $(\mathcal{R}_\lambda^{(i(\lambda))})_{\lambda\in\mathbb{N}}$. Then, $(\mathsf{Setup}, \mathsf{Hash})$ is $(s, L\cdot\epsilon)$-correlation intractable for the relation*

$$\mathcal{R} := \left(\bigcup_{i\in[L(\lambda)]} \mathcal{R}_\lambda^{(i)} \right)_{\lambda\in\mathbb{N}}.$$

Proof. Consider any PPT adversary $\mathcal{A}$ trying to win the correlation intractability game for $\mathcal{R}$. For this fixed adversary, the component success probabilities are at most $\epsilon(\lambda)$ simultaneously for all $i < L(\lambda)$ once λ is sufficiently large: otherwise a sequence of violating indices would contradict the hypothesis. We consider a series of $L(\lambda)+1$ hybrids, defined as follows:

Hybrid j: In this hybrid, the adversary wins if, on input $1^\lambda, 1^s$ and $\mathsf{hk} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$, it outputs $x \in \mathcal{X}_\lambda$ such that

$$(x, \mathsf{Hash}(\mathsf{hk}, x)) \in \bigcup_{i\in[j]} \mathcal{R}_\lambda^{(i)}.$$

It is easy to see that the advantage of $\mathcal{A}$ in Hybrid 0 is $\mathsf{Adv}^0_{\mathcal{A}}(\lambda) = 0$, as $\bigcup_{i\in[0]} \mathcal{R}_\lambda^{(i)} = \emptyset$. Moreover, by the (s,ϵ)-correlation intractability of $(\mathsf{Setup}, \mathsf{Hash})$ for $\mathcal{R}^{(i)}$, we have that

$$\mathsf{Adv}^{i+1}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{i}_{\mathcal{A}}(\lambda) \le \epsilon(\lambda)$$

Finally, notice that $\mathsf{Adv}^L_{\mathcal{A}}(\lambda)$ corresponds to the advantage of $\mathcal{A}$ against the correlation-intractability for $\mathcal{R}$. From all of these observations, we infer the claim. □

The Construction. We are ready to introduce our final construction. The idea is quite simple: we combine our correlation intractable hash function for searchable batched relations with a PRF. The goal of the latter is to reduce the problem from a batched relation $\mathcal{R}^{(0)} \times \cdots \times \mathcal{R}^{(c-1)}$ where every x is in relation with a large number of y's, to another batched relation $\mathcal{R}'^{(0)} \times \cdots \times \mathcal{R}'^{(c_0-1)}$ where every x is in relation with a much smaller number of elements. At that point, we use Lemma 1 and Lemma 3 to argue the security of the whole construction.

Formally, the construction below tackles a c-batched relation

$$\mathcal{R} := \mathcal{R}^{(0)} \times \cdots \times \mathcal{R}^{(c-1)},$$

where each $\mathcal{R}^{(i)} \subseteq \{0,1\}^\ell \times \{0,1\}^r$, and $c(\lambda)$ can be decomposed as the product $c_0(\lambda)\cdot c_1(\lambda)$. We rely on a hash function $\mathsf{CIBSR} = (\mathsf{Setup}, \mathsf{Hash})$ that is correlation intractable for every c_0-batched searchable relation in $\left(\{0,1\}^{\ell c_1}\right)^{c_0} \times (\{0,1\}^a)^{c_0}$. Moreover, we rely on a pseudorandom function $\mathsf{PRF} = (\mathsf{Setup}, \mathsf{Eval})$ with domain $\{0,1\}^a$ and range $\{0,1\}^{rc_1}$.

$\mathsf{Setup}(1^\lambda)$**:** Compute $\mathsf{hk}' \xleftarrow{\$} \mathsf{CIBSR.Setup}(1^\lambda)$ and $K \xleftarrow{\$} \mathsf{PRF.Setup}(1^\lambda)$. Then, output $\mathsf{hk} := (\mathsf{hk}', K)$.

$\mathsf{Hash}(\mathsf{hk}, x)$**:** Compute $y \leftarrow \mathsf{CIBSR.Hash}(\mathsf{hk}', x)$. Rewrite $y \in \{0,1\}^{c_0\cdot a}$ as

$$(y_0, \ldots, y_{c_0-1}), \qquad \forall i \in [c_0] : \quad y_i \in \{0,1\}^a.$$

Then, for every $i \in [c_0]$, compute $z_i \leftarrow \mathsf{PRF.Eval}(K, y_i)$. Finally, output the concatenation of $z_0, \ldots, z_{c_0-1}$.

We summarise the security properties of the above construction in the following theorem.

Theorem 7. *Assume that, for all sufficiently large λ, every $i < c(\lambda)$ and every $x \in \{0,1\}^\ell$ satisfy $|\{y : (x,y) \in \mathcal{R}_\lambda^{(i)}\}| \leq \delta(\lambda)2^r$. Suppose that* CIBSR *is (s,ϵ)-correlation intractable for every c_0-batched searchable relation, and* PRF *is $(s_{\mathsf{PRF}}, \epsilon_{\mathsf{PRF}})$-secure for*

$$s_{\mathsf{PRF}} \geq 2^a \cdot (\delta \cdot 2^r)^{c_1}.$$

Then, for any function $\gamma : \mathbb{N} \to \mathbb{R}$ such that $\gamma > 0$, the above hash function $(\mathsf{Setup}, \mathsf{Hash})$ *is (s,ϵ')-correlation intractable for $\mathcal{R} := \mathcal{R}^{(0)} \times \cdots \times \mathcal{R}^{(c-1)}$, where*

$$\epsilon'(\lambda) := ((1+\gamma) \cdot \delta^{c_1} \cdot 2^a)^{c_0} \cdot \epsilon(\lambda) + c_0 \cdot 2^{\ell c_1} \cdot \left(\epsilon_{\mathsf{PRF}} + e^{-\frac{\gamma^2 \cdot \delta^{c_1} \cdot 2^a}{2+\gamma}}\right).$$

Proof. Consider any elements $x_0, \ldots, x_{c_1-1} \in \{0,1\}^\ell$ and any $j \in [c_0]$. Since the component sparsity bound holds uniformly for $i < c$, the probability that random elements $z_0, \ldots, z_{c_1-1} \xleftarrow{\$} \{0,1\}^r$ satisfy $(x_i, z_i) \in \mathcal{R}_\lambda^{(i+j\cdot c_1)}$ for every $i \in [c_1]$ is eventually dominated by $\delta(\lambda)^{c_1}$.

Now, consider the truth table of $\mathsf{PRF}(K,\cdot)$. By the $(s_{\mathsf{PRF}}, \epsilon_{\mathsf{PRF}})$-security of PRF, we have that the truth table is ϵ_{PRF}-indistinguishable from a random table of 2^a elements of $\{0,1\}^{c_1 r}$ even for adversaries that run in time $\mathsf{poly}(s_{\mathsf{PRF}}, \lambda)$. Notice that these adversaries are able to check the whole truth table of PRF and they are able to check whether any candidate $z \in \{0,1\}^r$ is such that $(x_i, z) \in \mathcal{R}_\lambda^{(i+jc_1)}$ for any $i \in [c_1]$.

By the Chernoff bound, with probability $1 - e^{-\frac{\gamma^2 \cdot \delta^{c_1} \cdot 2^a}{2+\gamma}}$, the number of entries $(z_0, \ldots, z_{c_1-1})$ in a random truth table of size 2^a such that $(x_i, z_i) \in \mathcal{R}_\lambda^{(i+jc_1)}$ for every $i \in [c_1]$ is at most $(1+\gamma) \cdot \delta^{c_1} \cdot 2^a$. In other words, with probability at least $1 - c_0 \cdot 2^{\ell c_1} \cdot (\epsilon_{\mathsf{PRF}} + e^{-\frac{\gamma^2 \cdot \delta^{c_1} \cdot 2^a}{2+\gamma}})$, there exist no elements $x_0, \ldots, x_{c_1-1} \in \{0,1\}^\ell$, $j \in [c_0]$ such that

$$\left| \left\{ y \in \{0,1\}^a \middle| \exists z_0, \ldots, z_{c_1-1} \in \{0,1\}^r : \quad \begin{pmatrix} \mathsf{PRF.Eval}(K,y) = (z_0, \ldots, z_{c_1-1}) \\ \forall i \in [c_1] : (x_i, z_i) \in \mathcal{R}_\lambda^{(i+jc_1)} \end{pmatrix} \right\} \right| > (1+\gamma) \cdot \delta^{c_1} \cdot 2^a.$$

For every $j \in \mathbb{N}$, consider now the relation $\mathcal{R}'^{(j)} = (\mathcal{R}'^{(j)}_\lambda)_{\lambda \in \mathbb{N}}$ defined as

$$\mathcal{R}'^{(j)}_\lambda := \left\{ ((x_0, \ldots, x_{c_1-1}), y) \middle| \begin{array}{l} \forall i \in [c_1] : x_i \in \{0,1\}^\ell \\ y \in \{0,1\}^a \\ \\ \exists z_0, \ldots, z_{c_1-1} \in \{0,1\}^r : \quad \begin{pmatrix} \mathsf{PRF.Eval}(K,y) = (z_0, \ldots, z_{c_1-1}) \\ \forall i \in [c_1] : (x_i, z_i) \in \mathcal{R}_\lambda^{(i+jc_1)} \end{pmatrix} \end{array} \right\}$$

We conclude that with probability at least $1 - c_0 \cdot 2^{\ell c_1} \cdot (\epsilon_{\mathsf{PRF}} + e^{-\frac{\gamma^2 \cdot \delta^{c_1} \cdot 2^a}{2+\gamma}})$ over the randomness of $K \xleftarrow{\$} \mathsf{PRF.Setup}(1^\lambda)$, the relation $\mathcal{R}'^{(j)}$ is $(1+\gamma) \cdot \delta^{c_1}$-sparse for every $j \in [c_0]$.

Conditioned on this event, we infer that, if we consider the c_0-batched relation

$$\mathcal{R}' := \mathcal{R}'^{(0)} \times \cdots \times \mathcal{R}'^{(c_0-1)},$$

for any $x \in \{0,1\}^{\ell \cdot c_0 \cdot c_1}$, it holds that

$$\left|\left\{ y \in \{0,1\}^{a \cdot c_0} \middle| (x,y) \in \mathcal{R}'_\lambda \right\}\right| \leq ((1+\gamma) \cdot \delta^{c_1} \cdot 2^a)^{c_0}.$$

By Lemma 1, this means that the c_0-batched relation $\mathcal{R}'$ is contained in the union of at most $((1+\gamma)\cdot\delta^{c_1}\cdot 2^a)^{c_0}$-many c_0-batched searchable relations.

For any fixed adversary, security for every sequence of batched searchable relations gives a bound uniform over the searchable slices and the good PRF keys: if a violating slice and key existed at infinitely many parameters, choose them at those parameters and hardcode the polynomial-length key as non-uniform advice. This would contradict the assumed security for that sequence of searchable relations. Thus the preceding union lemma applies even after conditioning on the public PRF key.

We terminate the proof by applying Lemma 3: with probability

$$1 - c_0 \cdot 2^{\ell c_1} \cdot (\epsilon_{\mathsf{PRF}} + e^{-\frac{\gamma^2 \cdot \delta^{c_1} \cdot 2^a}{2+\gamma}}),$$

we have that CIBSR is $(s, ((1+\gamma)\cdot\delta^{c_1}\cdot 2^a)^{c_0}\cdot\epsilon)$-correlation intractable for $\mathcal{R}'$. □

The following corollary summarises our results. Essentially, this states that under the subexponential hardness of DSIS, there exists a correlation intractable hash function for every relation $\mathcal{R}^c$ provided that $c(\lambda)$ is a sufficiently large polynomial and that the sparsity of $\mathcal{R}$ is bounded away from 1 by a non-negligible amount.

Corollary 2. *Let $\mathcal{R} = (\mathcal{R}_\lambda \subseteq \{0,1\}^\ell \times \{0,1\}^r)_{\lambda\in\mathbb{N}}$ be a δ-sparse relation with $\ell, r = \mathsf{poly}(\lambda)$ where $\delta \le 1 - 1/\mathsf{poly}(\lambda)$. Assume the hardness of $(k, m, t, q, \sigma, B, s, \epsilon)$-DSIS, with k as its security parameter, all remaining parameters functions of k, and $q > B$ prime, where there exist constants $\alpha, \beta, \mu, \nu > 0$ such that $\min\{\mu,\beta\} + \nu > 1 + \alpha$ and*

$$2k\log q \le m \le k^{O(1)}, \qquad t = 2mk\log q, \qquad 3 \le \sigma \le k^{O(1)}, \qquad \rho_{\sigma,q} \le 1/(\sigma-1)$$
$$\log q = \Theta(k^\alpha + k^\beta), \qquad \log B = \Theta(k^\beta), \qquad \log q - \log B = \Theta(k^\alpha),$$
$$\log s = \Theta(k^\mu), \qquad \log \epsilon^{-1} = \Theta(k^\nu).$$

Then, for every sufficiently large $c(\lambda) = \mathsf{poly}(\lambda)$, there exists a correlation intractable hash function for the relation $\mathcal{R}^c$.

Proof. With these parameters, Theorem 4 implies subexponential hardness of SIS and hence the existence of subexponentially secure PRFs. Their security parameter may be chosen independently as a polynomial in λ.

Increasing the sparsity bound if necessary, take $\delta = 1 - \lambda^{-D}$ for a fixed positive integer D. By slightly weakening the security exponent, we may assume that ν is rational, $\nu < 1 + \alpha$, and $\epsilon = 2^{-\lfloor k^\nu \rfloor}$, while retaining $\min\{\mu,\beta\} + \nu > 1 + \alpha$. Fix a rational constant $\zeta < \min\{\mu,\beta\}$ such that $\zeta + \nu > 1 + \alpha$.

Instantiate the assumed parameter family at $k = \lceil \lambda^C \rceil$, for a sufficiently large integer C, and set

$$c_1 := \left\lceil \frac{k^\zeta}{\ell} \right\rceil, \qquad h := \ell c_1, \qquad b := -\log\delta, \qquad E := -\log\epsilon.$$

Then $h = \Theta(k^\zeta)$, so $h = o(\log s)$ and

$$h\log(2+h)\log(mnt\sigma\sqrt{n}) = O\big(k^\zeta(\log k)^2\big) = o(\log B).$$

Thus Corollary 1 applies to h-bit inputs with searchable-relation error $\eta := \epsilon + 2^{-\Omega(n)} \le 2\epsilon$, since $\nu < 1 + \alpha$.

Let $N := \lceil k(\log q - \log B + C_0 \log(mnt)) \rceil$, with C_0 sufficiently large for Corollary 1; thus $N = \Theta(k^{1+\alpha})$. Fix $\gamma > 7.4$ and set

$$A := h + E + \lambda, \qquad a := \lceil bc_1 + \log A \rceil, \qquad c_0 := \left\lceil \frac{N}{a} \right\rceil, \qquad f := (1+\gamma)\delta^{c_1} 2^a.$$

These choices are computable: equivalently, a is the least integer with $2^a \delta^{c_1} \geq A$. Moreover, $(1+\gamma)A \leq f < 2(1+\gamma)A$ and $N \leq ac_0 < N + a$. Since $b \leq 1$ eventually and $\zeta < \beta$, we have $a = O(k^\zeta + \log k)$.

Because $b \geq \lambda^{-D}$ and ℓ is polynomially bounded, choosing C sufficiently large ensures

$$\frac{bc_1 E}{N} = \Omega\left(\frac{\lambda^{-D}}{\ell}\, k^{\zeta+\nu-1-\alpha}\right) = \omega(\log \lambda).$$

It follows that

$$c_0 \log f \leq \left(\frac{N}{bc_1} + 1\right) O(\log \lambda) = o(E), \qquad f^{c_0}\eta \leq 2^{1-E+o(E)}.$$

Instantiate the PRF so that $s_{\mathsf{PRF}} \geq 2^a(\delta 2^r)^{c_1}$ and $\epsilon_{\mathsf{PRF}} \leq 2^{-2h-\lambda}$. These requirements have polynomial exponents in λ, so they follow by polynomially increasing its security parameter. By Theorem 7 and the choice of γ, the correlation-intractability error for $\mathcal{R}^{c_0 c_1}$ is at most

$$f^{c_0}\eta + c_0 2^h(\epsilon_{\mathsf{PRF}} + 2^{-f}) \leq 2^{1-E+o(E)} + c_0 2^{-h-\lambda} + c_0 2^{h-(1+\gamma)A} = \mathsf{negl}(\lambda).$$

Finally, for any $c \geq c_0 c_1$, hash the first $c_0 c_1$ input blocks with this construction and append arbitrary fixed output blocks. Membership in $\mathcal{R}^c$ implies membership of the retained blocks in $\mathcal{R}^{c_0 c_1}$, which proves the claim for every sufficiently large polynomial number of repetitions. □

7 The Parallel Repetition of 3-Round Proofs is Not Zero-Knowledge

As a consequence of our main result, the parallel repetition (for a sufficiently large number of repetitions) of any perfectly complete public-coin three-round proof with a non-trivial soundness gap for a language outside BPP is not zero-knowledge. This follows by the well-known connection between correlation intractability and zero-knowledge [DNRS99, HLR21]. For completeness, we recall the argument here, but this implication can also be found explicitly in [HLR21].

A three-round proof Π between a prover and a verifier for a language $\mathcal{L}$ satisfies c-completeness if for all $x \in \mathcal{L}$ the probability that the transcript (α, β, γ) is accepting is at least $c = c(\lambda)$. For the special case of three-round proofs we can define the s-soundness as follows: For all $x \notin \mathcal{L}$ and all first messages α, it holds that

$$\Pr_{\beta}\left[\exists\ \gamma : (\alpha, \beta, \gamma) \in \text{Accept}\right] \leq s$$

for some function $s = s(\lambda)$. Then we say that the protocol is non-trivially sound if $c - s \geq 1/\mathsf{poly}(\lambda)$. The following claim is a restatement from [DNRS99, HLR21]. The original proof considers the case of a completeness-soundness gap negligibly close to 1, but the proof works almost verbatim for the case of a non-trivial gap. We recall the proof for completeness.

Lemma 4 ([DNRS99, HLR21])**.** *Assume that there exists a function family* $(\mathsf{Setup}, \mathsf{Hash})$ *and a three-round protocol* Π *for a language* $\mathcal{L}$ *such that the protocol with* $\beta = \mathsf{Hash}(\mathsf{hk}, (x, \alpha))$ *has completeness at least* c*, computational soundness at most* s*, and* $c - s \geq 1/\mathsf{poly}(\lambda)$*. If the three-round protocol is auxiliary-input zero-knowledge, then* $\mathcal{L} \in \mathrm{BPP}$*.*

Proof Sketch. Consider a verifier that is given as auxiliary input a function $\mathsf{hk} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ and that computes $\beta = \mathsf{Hash}(\mathsf{hk}, (x, \alpha))$. Then, by assumption, there exists a simulator whose output for every $x \in \mathcal{L}$ is computationally indistinguishable from the honest interaction. Consider the algorithm that samples hk, runs this simulator on (x, hk), and accepts if β was computed correctly and the transcript is accepting. We have two cases:

- $x \in \mathcal{L}$: In this case the algorithm accepts with probability at least $c - \mathsf{negl}(\lambda)$, since the simulator produces a transcript that is computationally indistinguishable from the original one.
- $x \notin \mathcal{L}$: In this case the algorithm accepts with probability at most $s + \mathsf{negl}(\lambda)$, since the simulator itself is an efficient prover against the computational soundness of the protocol with β computed as the output of the hash.

The gap is inverse-polynomial, and standard amplification therefore gives a BPP decision algorithm for $\mathcal{L}$. □

Thus, we obtain the following consequence. Let Π be a perfectly complete public-coin three-round protocol with a non-trivial soundness gap for a language not in BPP. Its c-fold parallel repetition, for a sufficiently large $c = \mathsf{poly}(\lambda)$, defines a batched relation whose component sparsity is bounded away from 1 by a non-negligible amount. Combining Lemma 4 with Corollary 2, we obtain that the parallel repeated protocol is not auxiliary-input zero-knowledge.

8 NIZKs from Decomposed SIS

We conclude by recording a direct application of our correlation-intractable hash functions to non-interactive zero knowledge. We follow the Fiat–Shamir framework of [CCH+19], using the fact that our construction supports batched searchable relations even when they are not efficiently searchable.

Theorem 8. *Assuming the sub-exponential hardness of DSIS in the parameter regime of Corollary 1, every language in* NP *has an adaptively sound multi-theorem computational zero-knowledge non-interactive argument in the common reference string model.*

Proof Sketch. Under the parameters of Corollary 1, DSIS implies the existence of one-way functions. Indeed, consider the function family keyed by $(\mathbf{A}, \mathbf{B}, \mathbf{Z})$ that maps a binary vector $\mathbf{x} \in \{0,1\}^{t^2 n}$ to $\mathbf{Zx}$. Since $t = 2mn$, the binary domain is larger than $\mathbb{Z}_q^k$, so this family is compressing. Any collision $\mathbf{x} \neq \mathbf{x}'$ yields a non-zero vector $\mathbf{x} - \mathbf{x}'$ in the kernel of $\mathbf{Z}$ with infinity norm at most $1 \leq B$, contradicting DSIS. Thus, this family is collision resistant and, by compression, yields a one-way function. One-way functions imply statistically binding and computationally hiding commitments [HILL99, Nao91].

We use the delayed-input adaptive honest-verifier zero-knowledge protocol for graph Hamiltonicity of [FLS90]. This is a three-message proof, which we instantiate with this commitment scheme and repeat c times in parallel. The receiver message pk of the commitment scheme is included in the common reference string. In every repetition:

- The first prover message α_i consists of commitments to the adjacency matrix of a randomly permuted canonical cycle.
- The verifier sends a random bit β_i.
- The prover sends a response γ_i. If $\beta_i = 0$, the prover opens the entire committed graph and the verifier checks that it is a cycle. If $\beta_i = 1$, the prover opens the entries corresponding to the non-edges of a permutation of the statement and the verifier checks that all of them are zero.

Define $f(\mathsf{pk}, \alpha_i) = 0$ if α_i has a valid opening to a cycle, and $f(\mathsf{pk}, \alpha_i) = 1$ otherwise. This function need not be efficiently computable. Fix a receiver message for which the commitment scheme is binding. Every accepting transcript for a false statement satisfies $\beta_i = f(\mathsf{pk}, \alpha_i)$. Indeed, challenge 0 requires an opening to a cycle, whereas an accepting response to challenge 1 for commitments that open to a cycle would exhibit a Hamiltonian cycle in the statement. Consequently, the bad-challenge relation of the parallel repeated protocol is contained in the c-batched searchable relation $\mathcal{R}_f^c$.

Let $(\mathsf{Setup}, \mathsf{Hash})$ be the hash function from Corollary 1, instantiated with $a = 1$ and a sufficiently large c, for the relation $\mathcal{R}_f^c$. We first apply a standard mask to make this hash function programmable. Define $(\mathsf{Setup}', \mathsf{Hash}')$ as follows:

$\mathsf{Setup}'(1^\lambda)$**:** Compute $\mathsf{hk} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ and $\mathbf{u} \xleftarrow{\$} \{0,1\}^c$. Output $\mathsf{hk}' := (\mathsf{hk}, \mathbf{u})$.

$\mathsf{Hash}'(\mathsf{hk}', x)$**:** Output $\mathsf{Hash}(\mathsf{hk}, x) \oplus \mathbf{u}$.

The mask preserves correlation intractability for every c-batched searchable relation. Indeed, for every fixed $\mathbf{u}$, finding an input whose output under Hash' satisfies $\mathcal{R}_f^c$ is equivalent to breaking correlation intractability of $(\mathsf{Setup}, \mathsf{Hash})$ for the batched searchable relation obtained by replacing the output of the search function in coordinate i with $f(\mathsf{pk}, \alpha_i) \oplus \mathbf{u}[i]$. A standard non-uniform averaging argument over $\mathbf{u}$ therefore gives the claim.

Moreover, $(\mathsf{Setup}', \mathsf{Hash}')$ is one-point programmable. Given any input x and target output $y \in \{0,1\}^c$, sample $\mathsf{hk} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$ and set

$$\mathbf{u} := \mathsf{Hash}(\mathsf{hk}, x) \oplus y.$$

The resulting $\mathsf{hk}' = (\mathsf{hk}, \mathbf{u})$ is distributed exactly as a key sampled by Setup' conditioned on $\mathsf{Hash}'(\mathsf{hk}', x) = y$.

The common reference string of the NIZK consists of pk and $\mathsf{hk}' \xleftarrow{\$} \mathsf{Setup}'(1^\lambda)$. Given a statement and a witness, the prover computes the first messages $\alpha_0, \ldots, \alpha_{c-1}$ and sets

$$(\beta_0, \ldots, \beta_{c-1}) \leftarrow \mathsf{Hash}'(\mathsf{hk}', ((\mathsf{pk}, \alpha_0), \ldots, (\mathsf{pk}, \alpha_{c-1}))).$$

It then computes the responses $\gamma_0, \ldots, \gamma_{c-1}$ and outputs the resulting transcript. The verifier recomputes the challenges and checks every repetition. Completeness follows immediately from the completeness of the interactive protocol.

For soundness, suppose that an adversary produces an accepting proof for a false statement. Except with negligible probability, pk is binding. It follows that $\beta_i = f(\mathsf{pk}, \alpha_i)$ for every $i \in [c]$, and therefore

$$\big(((\mathsf{pk}, \alpha_0), \ldots, (\mathsf{pk}, \alpha_{c-1})), \mathsf{Hash}'(\mathsf{hk}', ((\mathsf{pk}, \alpha_0), \ldots, (\mathsf{pk}, \alpha_{c-1})))\big) \in \mathcal{R}_f^c,$$

contradicting correlation intractability. Since f is defined independently of the statement, this argument also holds when the statement is chosen after seeing the common reference string, and soundness is adaptive.

Finally, consider computational zero knowledge. We use adaptive-input honest-verifier zero knowledge together with one-point programmability. The simulator first samples pk honestly. Using the adaptive-input honest-verifier simulator, it then generates the first messages and uniform challenges

$$(\alpha_0, \ldots, \alpha_{c-1}) \qquad \text{and} \qquad (\beta_0, \ldots, \beta_{c-1}).$$

It samples hk honestly and programs $\mathbf{u}$ so that the displayed hash equation holds. Once the statement is chosen, the adaptive-input simulator completes the responses $\gamma_0, \ldots, \gamma_{c-1}$. One-point programmability and adaptive-input honest-verifier zero knowledge imply that the simulated common reference string and transcript are computationally indistinguishable from the real distribution. This gives single-theorem zero knowledge. The standard OR trick of [FLS90] gives multi-theorem zero knowledge without any additional assumption. □

Acknowledgments

We thank Alex Lombardi for discussions at an early stage of this project, for pointing us to the use of pseudorandom functions in [HLR21], and for encouraging us to complete this manuscript. We thank Abhishek Jain for suggesting the implication of NIZKs from decomposed SIS. The constructions and proofs were produced by humans. ChatGPT was used solely for writing assistance. The authors take full responsibility for all claims and proofs made in this paper.

D.A. was supported by a fellowship with the Simons Institute for the Theory of Computing. This work was supported by Input Output (iog.io) through their funding of the Edinburgh Blockchain Technology Lab. G.M. was supported by the European Research Council through an ERC Starting Grant (Grant agreement No. 101077455, ObfusQation) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA – 390781972. L.R. was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement number 101124977 (DECRYPSIS).

Part of this work was conducted while the authors were visiting the Simons Institute for the Theory of Computing.

References

[AMR25a] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Key-homomorphic computations for ram: Fully succinct randomised encodings and more. In *Advances in Cryptology – CRYPTO 2025: 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17–21, 2025, Proceedings, Part III*, page 236–268, Berlin, Heidelberg, 2025. Springer-Verlag. 2, 4, 5, 10, 11

[AMR25b] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Succinct oblivious tensor evaluation and applications: Adaptively-secure laconic function evaluation and trapdoor hashing for all circuits. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, page 1875–1886, New York, NY, USA, 2025. Association for Computing Machinery. 4, 8

[AMR26] Damiano Abram, Giulio Malavolta, and Lawrence Roy. Tree encodings I: How to authenticate a truth table. Cryptology ePrint Archive, Paper 2026/741, 2026. 7, 15, 16

[BBK+23] Zvika Brakerski, Maya Farber Brodsky, Yael Tauman Kalai, Alex Lombardi, and Omer Paneth. SNARGs for monotone policy batch NP. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part II*, volume 14082 of *LNCS*, pages 252–283. Springer, Cham, August 2023. 2

[BGG+14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556. Springer, Berlin, Heidelberg, May 2014. 4

[BKM20] Zvika Brakerski, Venkata Koppula, and Tamer Mour. NIZK from LPN and trapdoor hash via correlation intractability for approximable relations. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 738–767. Springer, Cham, August 2020. 7

[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. 2, 3, 25

[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, Berlin, Heidelberg, January 2016. 2

[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, Cham, April / May 2018. 2, 3

[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. 1, 12

[CJJ21] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. Non-interactive batch arguments for NP from standard assumptions. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 394–423, Virtual Event, August 2021. Springer, Cham. 2

[CJJ22] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for $\mathcal{P}$ from LWE. In *62nd FOCS*, pages 68–79. IEEE Computer Society Press, February 2022. 2

[DGKV22] Lalita Devadas, Rishab Goyal, Yael Kalai, and Vinod Vaikuntanathan. Rate-1 non-interactive arguments for batch-NP and applications. In *63rd FOCS*, pages 1057–1068. IEEE Computer Society Press, October / November 2022. 2

[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. 3, 24, 25

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st FOCS*, pages 308–317. IEEE Computer Society Press, October 1990. 25, 27

[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, Berlin, Heidelberg, August 1987. 1

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th FOCS*, pages 102–115. IEEE Computer Society Press, October 2003. 1

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In *27th FOCS*, pages 174–187. IEEE Computer Society Press, October 1986. 3

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. 25

[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. 2

[HLR21] Justin Holmgren, Alex Lombardi, and Ron D. Rothblum. Fiat-Shamir via list-recoverable codes (or: parallel repetition of GMW is not zero-knowledge). In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd ACM STOC*, pages 750–760. ACM Press, June 2021. 2, 3, 8, 14, 24, 25, 27

[JJ21] Abhishek Jain and Zhengzhong Jin. Non-interactive zero knowledge from sub-exponential DDH. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 3–32. Springer, Cham, October 2021. 2

[KLVW23] Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. Boosting batch arguments and RAM delegation. In Barna Saha and Rocco A. Servedio, editors, *55th ACM STOC*, pages 1545–1552. ACM Press, June 2023. 2

[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, Cham, August 2017. 2

[KRS25] Dmitry Khovratovich, Ron D. Rothblum, and Lev Soukhanov. How to prove false statements: Practical attacks on fiat-shamir. In *Advances in Cryptology – CRYPTO*

2025: 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17–21, 2025, Proceedings, Part VI, page 3–26, Berlin, Heidelberg, 2025. Springer-Verlag. 1

[Lom25] Alex Lombardi. Open Problems at the Simons Institute for the Theory of Computing, 2025. 2

[LV22] Alex Lombardi and Vinod Vaikuntanathan. Correlation-intractable hash functions via shift-hiding. In Mark Braverman, editor, *ITCS 2022*, volume 215, pages 102:1–102:16. LIPIcs, January / February 2022. 2, 7

[Mer88] Ralph C. Merkle. A digital signature based on a conventional encryption function. In Carl Pomerance, editor, *CRYPTO'87*, volume 293 of *LNCS*, pages 369–378. Springer, Berlin, Heidelberg, August 1988. 5

[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, Berlin, Heidelberg, April 2012. 4

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, January 1991. 25

[PP22] Omer Paneth and Rafael Pass. Incrementally verifiable computation via rate-1 batch arguments. In *63rd FOCS*, pages 1045–1056. IEEE Computer Society Press, October / November 2022. 2

[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, Cham, August 2019. 2, 7

[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. 3, 4

[Wee24] Hoeteck Wee. Circuit ABE with $\mathsf{poly}(\mathsf{depth}, \lambda)$-sized ciphertexts and keys from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 178–209. Springer, Cham, August 2024. 2