Binary Codes for Computationally Bounded Errors Under Standard Crypto Assumptions

George Lu
UT Austin
gclu@cs.utexas.edu

Jad Silbak
Northeastern
jadsilbak@gmail.com

Daniel Wichs
Northeastern University and NTT Research
wichs@ccs.neu.edu

Abstract

We study error-detection and error-correction codes for computationally bounded adversarial channels. We consider *seeded* codes where the polynomial-time encoding and decoding procedures share a public random seed, but are otherwise deterministic. An adversarial channel gets this seed and can perform arbitrary polynomial-time computation to adaptively select both the message to be encoded and a bounded number of errors to be added to the resulting codeword. The goal is to detect or correct such errors with overwhelming probability, while achieving better trade-offs between rate and error tolerance than those possible for computationally unbounded channels.

For large alphabets, prior work (ITCS '25) achieves essentially optimal parameters under minimal cryptographic assumptions. However, for the binary alphabet, prior works (TCC '20, EUROCRYPT '25) either only achieved a weaker notion of selective security under the learning with errors (LWE) assumption, or relied on non-standard cryptographic assumptions to get the full notion of adaptive security.

In this work, we construct binary codes that achieve the full notion of adaptive security assuming trapdoor hashing, which can in turn be instantiated under a variety of standard cryptographic assumptions such as LWE, or Decisional Diffie-Hellman (DDH), or Quadratic Residuosity (QR), or Decisional Composite Residuosity (DCR). For error detection, our codes get essentially optimal rate $R \approx 1$ and relative error tolerance $p \approx \frac{1}{2}$. For error correction, they can *uniquely* correct $p < 1/4$ fraction of errors with a rate R matching that of the best known *list-decodable* codes for this error tolerance.

As a central technical tool of potentially independent interest, we construct *multi-input correlation intractable hashing* for "shifted output relations" under the standard cryptographic assumptions above.

Contents

1 Introduction **3**
- 1.1 Our Techniques . . . 4

2 Preliminaries **7**
- 2.1 Relations and Correlation Intractable Hashing . . . 8
- 2.2 Standard Error Correcting Codes . . . 9
- 2.3 Codes For Computationally Bounded Adversaries: Seeded Codes . . . 10

3 Our Result for Seeded Codes: Theorem Statement **11**

4 Seeded Codes from 2-Input Correlation Intractable Hashing **12**
- 4.1 Decomposable Binary Linear Codes . . . 12
- 4.2 Construction and Proof for Seeded Codes . . . 13
  - 4.2.1 A Generic Approach to Error Correction . . . 18
  - 4.2.2 Proof of Main Theorems for Seeded Codes: Proof of Theorems 4.1 and 4.2. . . . 19

5 Multi-Input Correlation Intractability from Standard Assumptions **19**
- 5.1 Trapdoor Hashing . . . 20
  - 5.1.1 Generic Properties of Trapdoor Hash Functions . . . 21
- 5.2 Constructing Multi-Input Correlation Intractability from Trapdoor Hashing . . . 23
  - 5.2.1 Proof of Theorem 5.10 . . . 24

A Proofs for Decomposable Binary Codes **29**
- A.1 Proof of Theorem 4.5 . . . 29
- A.2 Proof of Theorem 4.6 . . . 29
  - A.2.1 Dual-Distance of a Code Concatenation . . . 30
  - A.2.2 List-Decodable Codes with Column-Robustness Approaching the Blokh-Zyablov Bound. . . . 31

B Trapdoor Hashing for Circuits **32**

C Public Coin Multi-Input Correlation Intractability **35**
- C.1 Public Coin Multi-Input CI From Explainability . . . 36
- C.2 Explainability of [DGI+19a] Trapdoor Hashing . . . 38
  - C.2.1 Learning with Errors . . . 38
  - C.2.2 Decisional Diffie Hellman . . . 38

1 Introduction

We study communication over a noisy adversarial channel that may introduce a bounded number of errors. The goal is to design codes that achieve *error-detection* (i.e., recover the original message when no errors have occurred and detect the presence of errors otherwise) or *error-correction* (i.e., always uniquely recover the original message). The main parameters of interest are the *rate* R, defined as the ratio of the message length to the codeword length, and the *relative error tolerance* p, defined as the ratio of the number of errors that can be detected or corrected to the codeword length. Codes for worst-case errors have been extensively studied since the seminal work of Hamming [Ham50]. In this case, the error-tolerance for both detection and correction is fully determined by the *distance* of the code, for which we have several well-established upper and lower bounds.

Seeded Codes for Computationally Bounded Channels. More recently, a long line of works [Lip94, OPS07, MPSW10, HOSW11, GS16, SS21a, GHY20, BGGZ21, SS21b, SS22, SS24, SW25b, SW25a] studies error detection and correction under the additional restriction that the channel is *computationally bounded*. Concretely, we consider the setting where the channel can perform arbitrary polynomial time (or even sub-exponential time) computation.[1] Interestingly, this restriction often allows us to surpass the information-theoretic limits.

Following [GHY20, SW25b, SW25a], we consider *seeded codes*, where the encoding and decoding procedures $(\mathsf{Enc}_{\mathsf{pp}}, \mathsf{Dec}_{\mathsf{pp}})$ are polynomial-time algorithms initialized with a public random seed pp, but are deterministic otherwise.[2] An arbitrary polynomial-time adversary gets the seed pp of the code and adaptively chooses a message m together with an erroneous codeword c' at a relative Hamming distance $\leq p$ from the correct codeword $c = \mathsf{Enc}_{\mathsf{pp}}(m)$. For error-correction we require that $\mathsf{Dec}_{\mathsf{pp}}(c') = m$ with overwhelming probability, and for error-detection we require that with overwhelming probability $\mathsf{Dec}_{\mathsf{pp}}(c') = m$ when $c' = c$ (no error) and $\mathsf{Dec}_{\mathsf{pp}}(c') = \perp$ otherwise (error).

Prior Work. The work of Grossman, Holmgren and Yogev [GHY20] introduced the notion of *seeded codes* and gave a construction based on a strong and non-standard cryptographic hardness assumption – namely a strong form of *two-input correlation-intractable hash functions*, which are currently only known to exist in the random-oracle model, but have no known provably secure instantiations under any standard cryptographic hardness assumptions.

The work of Silbak and Wichs [SW25b] considered a stronger notion of *self-seeded* codes, where the seed pp only needs to be known to the encoding algorithm but not the decoding algorithm; such codes imply seeded codes as a special case. They construct these codes for large constant sized alphabets with essentially optimal parameters under minimal assumptions: one-way functions for selective security and collision-resistant hash functions for adaptive security. The codes achieve essentially optimal error-detection with error tolerance $p \approx 1$ and rate $R \approx 1$ as well as essentially optimal error-correction with error tolerance $p < 1/2$ and rate $R \approx 1 - p$ in the large alphabet setting.[3] Unfortunately, their results crucially rely on the use of a large alphabet and do not extend to the more challenging case of a binary alphabet.

The work of [SW25a] constructed seeded codes for a binary alphabet under the *learning-with-errors (LWE)* assumption, but only satisfying a weak notion of selective security where the message to be encoded must be chosen by the adversary before seeing the public seed of the code.[4] They also constructed codes that satisfy our default notion of adaptive security, albeit only under a new non-standard assumption based on the "crypto dark matter" [BIP+18] methodology of mixing linear functions over different moduli. For error detection, their codes achieve essentially optimal rate $R \approx 1$ and relative error tolerance $p \approx \frac{1}{2}$. For error correction, their selectively secure codes under LWE can uniquely correct $p < 1/4$ relative errors with essentially the same rate R as the best currently known information-theoretic binary list-decodable codes [GR08] with error tolerance p, which is given by the Blokh-Zyablov bound $R_{BZ}(p)$ [BZ76]; see equation 2.1. Their adaptively secure codes under a non-standard assumption (as well as the random-oracle based construction of [GHY20]) achieve similar parameters, albeit only for some sufficiently smaller values of $p < .1$.

[1] For the rest of the introduction, we focus on the case of polynomial-time channels, but all results extend to sub-exponential time channels at the cost of needing stronger sub-exponential version of the underlying cryptographic assumptions.

[2] As discussed in prior works, some form of randomness is inherently needed to achieve security against non-uniform attacks.

[3] Throughout the introduction, we use "$\approx$" to hide additive constant factors that can be made arbitrarily small.

[4] The standard method to amplify selective security to adaptive security via complexity leveraging does not apply here since it blows up the parameters to the point where the resulting codes would be uninteresting.

We also briefly mention many other prior related works [Lip94, OPS07, MPSW10, HOSW11, GS16, SS21a, BGGZ21, SS21b, SS22, SS24] that do not quite fit our setting, either because they require more complex setup between the encoder and decoder (e.g., shared secret keys or a public-key infrastructure) or because they cannot handle arbitrary polynomial-time adversarial channels.

At a technical level, the prior results of [GHY20, SW25a] both relied on different forms of *multi-input correlation intractable hashing (MICIH)* [CGH04] as a building block. A 2-input MICIH (2-MICIH) for a class of relations $\mathcal{R}$ ensures that for any $R \in \mathcal{R}$, given a randomly chosen hash function H from the hash family $\mathcal{H}$, it is hard to find inputs $x_1 \neq x_2$ such that $R(x_1, x_2, H(x_1), H(x_2)) = 1$. The work of [GHY20] relied on 2-MICIH for general *efficiently computable sparse relations*, which we only know how to construct in the random oracle model. The work of [SW25a] relied on 2-MICIH for relations of the form $R_f(x_1, x_2, H(x_1), H(x_2)) = 1 \Leftrightarrow H(x_2) \in f(x_1, x_2, H(x_1))$ for some efficiently computable f that outputs a list. They were able to construct these under LWE, albeit only achieving a weak form of selective security where the adversary has to choose x_1 before seeding the seeded hash function H. This in turn only allowed them to achieve a weak form of selective security for their seeded codes.

Our Results. We revisit the problem of seeded codes for the binary alphabet. We achieve the strong default notion of adaptive security assuming the existence of *trapdoor hash functions* (TDH) for linear predicates [DGI+19b], which can in turn be instantiated under a variety of standard cryptographic assumptions including *learning-with-errors (LWE)* or *decisional Diffie-Hellman (DDH)* or *quadratic residuosity (QR)* or *decisional composite residuosity (DCR)*. This improves on the prior work of [SW25a] by giving the first constructions of seeded binary codes that: (1) achieves the strong default notion adaptive security under standard assumptions, (2) can be instantiated under standard assumptions other than LWE. For our constructions based on LWE and DDH, the seed can just be a uniformly chosen random string, while for our constructions based on QR and DCR it needs to be chosen from some specified distribution using secret random coins.

The parameters of our adaptively secure codes match those achieved by [SW25a] in the selective setting: for error detection, we get essentially optimal rate $R \approx 1$ and relative error tolerance $p \approx \frac{1}{2}$. For error correction, we can uniquely correct $p < 1/4$ relative errors with essentially the same rate R as the best currently known information-theoretic binary list-decodable codes [GR08] with error tolerance p, which is given by the Blokh-Zyablov bound $R_{BZ}(p)$ [BZ76]. The latter slightly improves on the prior adaptively secure constructions of [GHY20, SW25a] under non-standard assumptions, which only achieved such parameters for sufficiently small $p < .1$. See [SW25a] for an in-depth discussion of these parameters, how they compare to information-theoretic codes, and their optimality. In brief, our achieved parameters for error detection are optimal. For error correction, requiring error tolerance $p < 1/4$ is also optimal. However, one may hope for higher rate $R \approx 1 - H(p)$ matching the Shannon bound, which exceed $R \approx R_{BZ}(p)$. While we do not currently know how to achieve this, our construction template shows how to do so given plausible future advances in list-decodable binary codes.

Technically, our result consists of two main components. As our first contribution, we show that 2-MICIH for the class $\mathcal{R}$ of *linearly shifted output relations* suffices to construct seeded binary codes. The relations $R \in \mathcal{R}$ consist of some arbitrary efficiently computable sparse relation R_0 and a linear function f such that

$$R(x_1, x_2, H(x_1), H(x_2)) = 1 \Leftrightarrow R_0(H(x_1) + f(x_1), H(x_2) + f(x_2)) = 1.$$

Prior work [LV22] constructed such 2-MICIH from indistinguishability obfuscation. As our second contribution, which may be of independent interest, we construct such 2-MICIH generically from trapdoor hash functions for linear predicates [DGI+19b], which can in turn be instantiated under a host of standard cryptographic assumptions listed above. In fact, our construction generalizes to MICIH for an arbitrary number of inputs. Moreover, under the LWE assumption, it generalizes to shifted output relations where the shift f can be any bounded-depth circuit, and under an additional circular security assumption we can even allow for arbitrary bounded-size circuits f.

1.1 Our Techniques

We give a brief overview of our techniques. We first show how to achieve seeded error-detection via 2-input correlation-intractable hashing (2-MICIH) for linearly shifted output relations. We then show how to extend the construction to also achieve error-correction. Finally, we show how to construct MICIH from Trapdoor Hashing.

Error-Detection Code Construction. Our construction of error-detection codes closely follow that of [SW25a]. We rely on a hash function family $\mathcal{H}$ consisting of functions $H : \{0,1\}^k \to \{0,1\}^v$ with small output length $v \ll k$. Let $\mathbf{G} = \left[\begin{smallmatrix}\mathbf{A}\\ \mathbf{B}\end{smallmatrix}\right] \in \mathbb{F}_2^{n\times n}$ be a full-rank matrix with $n := k + v$. The seed of the code consists of a random $H \leftarrow \mathcal{H}$ and the encoding algorithm $\mathsf{Enc}_H(m)$ gets a message $m \in \{0,1\}^k$, computes $H(m)$, and outputs the codeword $c \in \{0,1\}^n$ via:

$$c := (m, H(m))\mathbf{G} = m\mathbf{A} + H(m)\mathbf{B}.$$

The decoding algorithm $\mathsf{Dec}_H(c')$ computes $(m', y') = c'\mathbf{G}^{-1}$ and outputs the message m' if $y' = H(m')$, and outputs the special error symbol $\perp$ otherwise. The rate of this code is $k/n = k/(k+v) \approx 1$. The intuition behind error-detection is that, given H, it should be hard to find $m \neq m'$ such that $(m, H(m))\mathbf{G}$ and $(m', H(m'))\mathbf{G}$ are "too close" – having relative Hamming distance $(1/2 - \varepsilon)$ for any constant $\varepsilon > 0$. But unfortunately, this property does not appear to follow from standard collision resistance. The work of [SW25a] only proved a weaker selective security version of this property (where m had to be chosen selectively before seeing H) for a specific hash family constructed using LWE, in the case where $\mathbf{B}$ is an efficiently list-decodable code.

Analysis via 2-MICIH. We give a new analysis showing that adaptive security of the above construction is guaranteed if the hash family is 2-MICIH for linearly shifted output relations. We need the following additional "column robustness" property of the matrix $\mathbf{B} \in \mathbb{F}_2^{v\times n}$: if we sample a random subset T of v out of n columns of $\mathbf{B}$ then the corresponding matrix $\mathbf{B}_T \in \mathbb{F}_2^{v\times v}$, denoting the restriction of $\mathbf{B}$ to the columns selected by the set T, is invertible with noticable probability. This holds (e.g.,) for a random $\mathbf{B}$. Now, suppose that an adversary breaks the error-detection of our seeded code with non-negligible probability. This means that, given $H \leftarrow \mathcal{H}$ the adversary has a non-negligible probability of being able to come up with $m \neq m'$ such that:

$$\begin{aligned} & (m, H(m))\mathbf{G} = (m', H(m'))\mathbf{G} + \mathbf{e} \\ \Leftrightarrow \quad & m\mathbf{A} + H(m)\mathbf{B} = m'\mathbf{A} + H(m')\mathbf{B} + \mathbf{e} \end{aligned}$$

for some error $\mathbf{e} \in \mathbb{F}_2^n$ with relative Hamming weight at most $1/2 - \varepsilon$. Now consider choosing a random subset T of v out of n columns as a thought experiment, but not revealing it to the adversary. Note that $\mathbf{B}_T$ is invertible with noticeable probability and, by a simple Chernoff bound, the relative Hamming weight of $\mathbf{e}_T$ is approximately the same as that of $\mathbf{e}$ with overwhelming probability. Therefor, with non-negligible probability, $\mathbf{B}_T$ is invertible *and*:

$$m\mathbf{A}_T + H(m)\mathbf{B}_T = m'\mathbf{A}_T + H(m')\mathbf{B}_T + \mathbf{e}_T$$

for some $\mathbf{e}_T$ with relative Hamming weight at most $1/2 - \varepsilon/2$. This is equivalent to:

$$\begin{aligned} & m\mathbf{A}_T\mathbf{B}_T^{-1} + H(m) = m'\mathbf{A}_T\mathbf{B}_T^{-1} + H(m') + \mathbf{e}_T\mathbf{B}_T^{-1} \\ \Leftrightarrow \quad & R_0(H(m) + f(m), H(m') + f(m')) = 1 \end{aligned}$$

Where $f(x) = x\mathbf{A}_T\mathbf{B}_T^{-1}$ is a linear function and $R_0(y, y') = 1$ iff $y = y' + \mathbf{e}^*\mathbf{B}_T^{-1}$ for some $\mathbf{e}^* \in \mathbb{F}_2^v$ with relative Hamming weight at most $1/2 - \varepsilon/2$. Furthermore, R_0 is efficiently computable, and it is an exponentially sparse relation, meaning that $\mathbf{Pr}_{y,y' \leftarrow \{0,1\}^v}[R_0(y, y') = 1] \leq 2^{-\Omega(v)}$. Therefore the relation

$$R(m, m', H(m), H(m')) = R_0(H(m) + f(m), H(m') + f(m'))$$

is a linearly shifted output relation. Putting it all together, an adversary that breaks seeded error-detection of our code also breaks 2-MICIH security of the hash function family $\mathcal{H}$ for the class of $\mathcal{R}$ of linearly shifted output relations R.

Error-Correction via List Decoding. We combine the above ideas for error detection with list-decodable codes to get unique error correction. First, following [GHY20, SW25a], we show that any code that is simultaneously a seeded error-detection code with error-tolerance p_d and efficiently list-decodable for error rate p_l, also allows us to uniquely correct errors with tolerance $p = \min\{p_l, p_d/2\}$. Assume that an adversary gets the seed s of the code and comes up with a message m and some value c^* within relative distance p of $c = \mathsf{Enc}_s(m)$. By list-decoding c^* we get a polynomial list of candidate messages that includes m. We claim that there cannot be any other $m' \neq m$ in the list

such that $c' = \mathsf{Enc}_s(m')$ is within relative distance p of c^*: otherwise by triangle inequality c and c' would be within distance $\leq 2p = p_d$ of each other, which would allow an adversary who modifies c to c' to break error detection.

Secondly, we construct codes that are simultaneously seeded error-detection codes and list-decodable by modifying our construction above. Instead of a square invertible matrix $\mathbf{G}$, we now use a matrix $\mathbf{G} = \left[\begin{smallmatrix}\mathbf{A}\\ \mathbf{B}\end{smallmatrix}\right] \in \mathbb{F}_2^{(k+v)\times n}$ which is the generator matrix of an efficient list-decodable code having error-tolerance p_l and rate $R_l = (k+v)/n$; the best known such codes [GR08] achieve $R_l \approx R_{BZ}(p_l)$. The encoding algorithm is otherwise the same and outputs $\mathsf{Enc}_H(m) = (m, H(m))\mathbf{G} = m\mathbf{A} + H(m)\mathbf{B}$. The overall code then has rate $k/n \approx R_l$ and is clearly list decodable for up to p_l errors. We argue that it is also an error-detection code with tolerance $p_d \approx 1/2$ by essentially the same argument used for the earlier construction. There is one subtlety in that, for the known constructions of list-decodable codes, we can only show a weaker column-robustness property of the matrix $\mathbf{B} \in \mathbb{F}_2^{v\times n}$: for some $w = v^{\Omega(1)}$, if we sample a random subset T of w out of n columns of $\mathbf{B}$ then the corresponding matrix $\mathbf{B}_T \in \mathbb{F}_2^{v\times w}$ is full-rank with noticeable probability. As in the previous analysis, if an adversary breaks error-detection then, with non-negligible probability, $\mathbf{B}_T \in \mathbb{F}_2^{v\times w}$ is full-rank and the adversary finds $m \neq m'$ such that:

$$m\mathbf{A}_T + H(m)\mathbf{B}_T = m'\mathbf{A}_T + H(m')\mathbf{B}_T + \mathbf{e}_T$$

for some $\mathbf{e}_T$ with relative Hamming weight at most $1/2 - \varepsilon/2$. The previous analysis continued by relying on the square matrix $\mathbf{B}_T$ being invertible, and therefore we need to adapt it. Let ${\mathbf{B}_T}^{-1_L}$ be any *left inverse* of $\mathbf{B}_T$ such that ${\mathbf{B}_T}^{-1_L}\mathbf{B}_T = \mathbf{I}_w$ and let $\mathcal{N} = \{\mathbf{z} \in \mathbb{F}_2^v : \mathbf{z}\mathbf{B}_T = 0\}$ be the null space of $\mathbf{B}_T$. Then:

$$\begin{aligned}
& m\mathbf{A}_T + H(m)\mathbf{B}_T = m'\mathbf{A}_T + H(m')\mathbf{B}_T + \mathbf{e}_T \\
\Leftrightarrow\quad & m\mathbf{A}_T{\mathbf{B}_T}^{-1_L} + H(m) = m'\mathbf{A}_T{\mathbf{B}_T}^{-1_L} + H(m') + \mathbf{e}_T{\mathbf{B}_T}^{-1_L} + \mathbf{z} \;:\; \mathbf{z} \in \mathcal{N} \\
\Leftrightarrow\quad & R_0(H(m) + f(m), H(m') + f(m')) = 1,
\end{aligned}$$

where $f(x) = x\mathbf{A}_T{\mathbf{B}_T}^{-1_L}$ is a linear function and

$$R_0(y, y') = 1 \quad\Leftrightarrow\quad y = y' + \mathbf{e}_T^*{\mathbf{B}_T}^{-1_L} + \mathbf{z} \quad\Leftrightarrow\quad y\mathbf{B}_T = y'\mathbf{B}_T + \mathbf{e}^*$$

for some $\mathbf{e}^* \in F_2^w$ with relative Hamming weight at most $1/2 - \varepsilon/2$ and $\mathbf{z} \in \mathcal{N}$. Furthermore R_0 is efficiently computable and it is a sparse relation with $\mathbf{Pr}_{y,y' \leftarrow \{0,1\}^v}[R_0(y, y') = 1] \leq 2^{-\Omega(w)}$. Therefore an adversary that breaks seeded error-detection of this code also breaks 2-MICIH security for the class of $\mathcal{R}$ of linearly shifted output relations.

We note that, if there is a plausible future advance in constructing binary linear list-decodable codes achieving the Shannon bound with $R_l \approx 1 - H(p_l)$, we could plug it in to our construction to get seeded error-correction with unique decoding against computationally bounded channels with corresponding parameters, as long as the code satisfied the additional "column robustness" property.

MICIH from Trapdoor Hashing. We construct 2-input correlation-intractable hashing (2-MICIH) for shifted output relations from *trapdoor hash functions* [DGI+19b]. Trapdoor hash functions were originally conceived to construct high-rate oblivious transfer, private information retrieval, and related primitives. Prior works of [BKM20, BJSS25] also used them to construct a certain form of 1-input correlation intractability, but as far as we know, we are the first to use them to construct any form of MICIH.

Let $\mathcal{F}$ be a function family consisting of functions $f : \{0,1\}^k \to \{0,1\}^v$. A trapdoor hash for $\mathcal{F}$ consists of:

- $\mathsf{hk} \leftarrow \mathsf{Setup}()$: a setup algorithm that generates a hash key hk,
- $\sigma = h_{\mathsf{hk}}(x)$: a compressing hash that takes as input $x \in \{0,1\}^k$ and outputs $\sigma \in \{0,1\}^\ell$,
- $(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}(\mathsf{hk}, f)$: a key generation algorithm that takes as input a function $f \in \mathcal{F}$ and outputs an evaluation key ek and a trapdoor td,
- $y := \mathsf{E}(\mathsf{ek}, x)$: an encoding algorithm that outputs $y \in \{0,1\}^v$,
- $y' := \mathsf{D}(\mathsf{td}, \sigma)$ a decoding algorithm that outputs $y' \in \{0,1\}^v$.

The scheme has to satisfy *function privacy*, meaning that for any $f_0, f_1 \in \mathcal{F}$, one cannot distinguish whether an evaluation key ek was generated using f_0 or f_1. The scheme also has to satisfy a *correctness* property, which say that when ek is generated using some function $f \in \mathcal{F}$ then $\mathsf{E}(\mathsf{ek}, x) = \mathsf{D}(\mathsf{td}, h_{\mathsf{hk}}(x)) - f(x)$.[5] We focus on the setting where $\ell \ll v \ll k$; in fact we can set ℓ to some fixed polynomial in the security parameter independent of the choice of v, k. In that case, when ek is generated using f, then the function $\mathsf{E}(\mathsf{ek}, x) + f(x) \in \{0,1\}^v$ is *very lossy* since it only depends on a short hash $h_{\mathsf{hk}}(x) \in \{0,1\}^\ell$; concretely, the image of $\mathsf{E}(\mathsf{ek}, x) + f(x) = \mathsf{D}(\mathsf{td}, h_{\mathsf{hk}}(x))$ is of size $2^\ell \ll 2^v$. We additionally require that h_{hk} is collision-resistant, which we show to be automatically implied for any trapdoor hash for a sufficiently expressive class $\mathcal{F}$ (e.g., one that includes linear functions) by the properties above. Lastly, we require that $\mathsf{D}(\mathsf{td}, \cdot)$ is pairwise independent over the choice of td, and show that this property can generically be added to any trapdoor hash.

Assume we have a trapdoor hash as above. Consider the hash function family $H_{\mathsf{ek}}(x) = \mathsf{E}(\mathsf{ek}, x)$ where $\mathsf{hk} \leftarrow \mathsf{Setup}()$ and $(\mathsf{ek}, \cdot) \leftarrow \mathsf{Gen}(\mathsf{hk}, f_0)$ for some arbitrary fixed choice of $f_0 \in \mathcal{F}$. We claim that this hash family is 2-MICIH for $\mathcal{F}$-shifted output relations. In particular, no PPT adversary can win the following game except with negligible probability: the adversary chooses $f \in \mathcal{F}$ along with some "sufficiently sparse" and efficiently computable relation R_0, gets ek, selects $x \neq x'$ and wins if

$$R_0(\mathsf{E}_{\mathsf{ek}}(x) + f(x),\ \mathsf{E}_{\mathsf{ek}}(x') + f(x')) = 1.$$

We define sufficiently sparse to mean that $\mathbf{Pr}_{y,y'}[R_0(y, y')] \leq 2^{-2\ell}\mu$ for some negligible μ. To show the above, first observe that the adversary's winning probability can only change negligibly if we generate the evaluation key $(\mathsf{ek}, \cdot) \leftarrow \mathsf{Gen}(\mathsf{hk}, f)$ using the function f that the adversary selected instead of using the default function f_0. Thus, by correctness, the adversary wins if

$$R_0(\mathsf{D}(\mathsf{td}, h_{\mathsf{hk}}(x)), \mathsf{D}(\mathsf{td}, h_{\mathsf{hk}}(x'))) = 1,$$

This idea, namely, the use of function privacy to (undetectably) sample the evaluation key so as to cancel out the shift induced by the relation, thereby reducing a shifted relation on E to an output relation on D was first used by [BKM20] for constructing 1-input correlation intractable hash functions.

To handle 2-input (and more generically multi-input) relations we now make use of the two additional properties stated above (the collision resistance of h_{hk} and the pairwise independence of $\mathsf{D}(\mathsf{td}, \cdot)$). Observe that the adversary's winning probability can only change negligibly if we redefine the winning condition as:

$$R_0(\mathsf{D}(\mathsf{td}, h_{\mathsf{hk}}(x)), \mathsf{D}(\mathsf{td}, h_{\mathsf{hk}}(x'))) = 1 \quad \text{and} \quad h_{\mathsf{hk}}(x) \neq h_{\mathsf{hk}(x')},$$

which follows by the collision resistance. Finally, we rely on a statistical argument to show that the probability of winning in this modified game is negligible. For each $\sigma \neq \sigma' \in \{0,1\}^\ell$, by pairwise independence we have $\mathbf{Pr}[R_0(\mathsf{D}(\mathsf{td}, \sigma), \mathsf{D}(\mathsf{td}, \sigma')) = 1] \leq 2^{-2\ell}\mu$, and by union bound over all values of σ and σ' it follows that

$$\mathbf{Pr}[\exists \sigma \neq \sigma' \ :\ R_0(\mathsf{D}(\mathsf{td}, \sigma), \mathsf{D}(\mathsf{td}, \sigma')) = 1] \leq \mu.$$

Thus it is statistically unlikely that *any* pair of outputs of $\mathsf{D}(\mathsf{td}, \cdot)$ falls in R_0, bounding the probability that the adversary wins.

2 Preliminaries

Notation We write λ to denote the security parameter. For a positive integer $n \in \mathbb{N}$, we write $[n]$ to denote the set $\{1, \ldots, n\}$. For prime powers p, we use $\mathbb{F}_p$ to refer to the finite field of order p. We denote matrices with bold letters $(\mathbf{A}, \mathbf{B})$ and bold lowercase letters to denote vectors (e.g., $\mathbf{u}, \mathbf{v}$). We index into vectors via the non-boldfaced subscripts $\mathbf{v} = (v_1, \ldots, v_n)$. We use $\mathsf{poly}(x_1, \ldots, x_t)$ to denote polynomials in variables $x_1, \ldots, x_t$, and $\mathsf{negl}(x)$ to denote negligible functions in x (i.e. a function which is $< \frac{1}{\mathsf{poly}(x)}$ for any polynomial poly for sufficiently large x).

[5] In the body of the paper, we relax this correctness property to only hold with $1 - 1/\mathsf{poly}$ probability for each output bit and show that this suffices. This relaxed notion is also the one achieved under standard assumptions. However, for the sake of simplicity, we assume full correctness for the technical overview, as this already conveys the main intuition.

When performing arithmetic operations over binary strings $\in \{0,1\}^n$, we will implicitly assume arithmetic is performed over coordinate-wise over $\mathbb{F}_2^n$ unless otherwise specified. For pairs of binary strings $x_0, x_1 \in \{0,1\}^n$, we use $\Delta(x_0, x_1)$ to refer to the Hamming distance between x_0 and x_1 and $\delta(x_0, x_1) = \Delta(x_0, x_1)/n$ to be the relative hamming distance. We will use matrix $\mathbf{M} \in \mathbb{F}_2^{n \times m}$ to refer to linear function $f(x) : (\mathbb{F}_2^n \to \mathbb{F}_2^m) = x \cdot \mathbf{M}$. We use I_n to refer to the $n \times n$ identity matrix. For a matrix $\mathbf{M} \in \mathbb{F}^{n \times m}$ and a subset $T \subseteq [m]$, we use $[\mathbf{M}]_T$ to denote the $\mathbb{F}^{n \times |T|}$ submatrix composed of the columns of $\mathbf{M}$ indexed by T. When $n \geq m$ and $\mathbf{M}$ is of full rank m, we use $\mathbf{M}^{-1_L} \in \mathbb{F}^{m \times n}$ to refer to the (not necessarily unique) left inverse where $\mathbf{M}^{-1_L} \cdot \mathbf{M} = I_m$. Similarly, when $m \geq n$ and $\mathbf{M}$ is of full rank n, we will use $\mathbf{M}^{-1_R}$ to denote the right inverse where $\mathbf{M} \cdot \mathbf{M}^{-1_R} = I_n$. Note that these are efficiently computable, generally equivalent to solving overdetermined linear systems.

The Blokh-Zyablov Bound. Given a constant $0 \leq p \leq 1/2$, the Blokh-Zyablov bound is defined to be:

$$R_{BZ}(p) \coloneqq 1 - H(p) - p \cdot \int_0^{1-H(p)} \frac{dx}{H^{-1}(1-x)} \tag{2.1}$$

The best known explicit codes with efficient list decoding algorithms are by Guruswami and Rudra [GR08]. These codes are constructed by concatenating Folded Reed-Solomon codes with suitable inner codes so that they approach rate R_{BZ}. The Blokh-Zyablov bound due to Blokh and Zyablov generalizes the well known Zyablov Bound that provides an achievable trade-off between the rate and the relative distance of concatenated codes. A more detailed discussion on the Blokh-Zyablov bound and the explicit construction in [GR08], appears in the appendix of [SW25a].

2.1 Relations and Correlation Intractable Hashing

Relations. A t-ary relation R is a subset of $(\{0,1\}^k \times \{0,1\}^v)^t$. We will say $R((x^{(1)}, y^{(1)}), \ldots (x^{(t)}, y^{(t)})) = 1$ if $(x^{(1)}, y^{(1)}), \ldots, (x^{(t)}, y^{(t)})$ is contained in the set defined by R (and $R(\cdot) = 0$ otherwise). A relation ensemble $\mathcal{R} = \{R_\lambda\}$ is a set of relations indexed by security parameters $\lambda \in \mathbb{N}$, where $R_\lambda : (\{0,1\}^{k(\lambda)} \times \{0,1\}^{v(\lambda)})^{t(\lambda)}$.

Correlation intractable hashing. The notion of correlation intractable hashing was first introduced by Canetti, Goldreich, and Halevi [CGH04], as a way to model the behaviors and properties of a random oracle. Loosely speaking, a hash family $\mathcal{H} = \{h_K\}_{K \in \mathcal{K}}$ is correlation intractable for a (sparse) t-ary relationship $R((x^{(1)}, y^{(1)}), \ldots, (x^{(t)}, y^{(t)}))$, if given a random key $k \leftarrow \mathcal{K}$ it is computationally hard to find an $x^{(1)}, \cdots, x^{(t)}$, such that $((x^{(1)}, h_k(x^{(1)}), \ldots, (x^{(t)}, h_k(x^{(t)})) \in R$.

There has been much work on correlation intractability for single-input relations [CCR16, KRR17, CCRR18, HL18, CCH+19, PS19, BKM20, LV20]. The work of [LV22] instantiated certain forms of multi-input CI under the LWE assumption. However, in contrast to the single-input setting, multi-input correlation intractability (for any $t \geq 2$) remains significantly less understood.

Definition 2.1 (Correlation Intractable Hashing)**.** Consider a keyed hash function $\Pi_{\mathsf{Hash}} = (\mathsf{Gen} : 1^\lambda \times 1^k \times 1^v \to \mathcal{K}, \mathsf{Hash} : \mathcal{K} \times \{0,1\}^k \to \{0,1\}^v)$ and a t-ary relation ensemble $\mathcal{R} = \{\mathcal{R}_\lambda \subseteq (\{0,1\}^k \times \{0,1\}^v)^t\}$. We define the correlation intractability for $\mathcal{R} = \{R_\lambda\}$ via an experiment $\mathsf{ExpCI}^{(\mathcal{R})}(\mathcal{A}, \lambda)$ between an adversary $\mathcal{A}$ and challenger as follows:

- On input the security parameter 1^λ, the challenger samples a key $K \leftarrow \mathsf{Gen}(1^\lambda, 1^k, 1^v)$ and sends K to the adversary $\mathcal{A}$.
- The adversary returns inputs $x^{(1)}, \ldots, x^{(t)} \in \{0,1\}^n$ to the challenger.
- For $i \in [t]$, the challenger sets $y^{(i)} = \mathsf{Hash}(K, x^{(i)})$. The output of the experiment is $R_\lambda((x^{(1)}, y^{(1)}), \ldots (x^{(t)}, y^{(t)}))$.

We say that Π_{Hash} is t-input correlation intractable (t-MICIH) against $\mathcal{R}$ if for all polynomial time adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\mathbf{Pr}[\mathsf{ExpCI}^{(\mathcal{R})}(\mathcal{A}, \lambda) = 1] = \mathsf{negl}(\lambda).$$

Moreover, we say that Π_{Hash} is **public coin** if the function Gen samples key K as uniformly random bits.

For our particular applications, we will be concerned with correlation intractability with respect to a particular class of relations called *shifted output* relations.

Definition 2.2 (Shifted Output Relation [LV22])**.** A shifted output relation R is a relation defined by a tuple (R_{out}, f), where R_{out} is a subset of $(\{0,1\}^v)^t$ (an 'output relation') and $f : \{0,1\}^k \to \{0,1\}^v$ is a function. We define the relation $R = (R_{\text{out}}, f)$ as:

$$R((x^{(1)}, y^{(1)}), \ldots, (x^{(t)}, y^{(t)})) \Leftrightarrow (y^{(1)} - f(x^{(1)}), \ldots, y^{(t)} - f(x^{(t)})) \in R_{\text{out}} \wedge \forall i \neq j \in [t] : x^{(i)} \neq x^{(j)}.$$[6]

In other words, whether an input satisfies R only depends on the value of the shifted output $y - f(x)$. We say a shifted output relation $R = (R_{\text{out}}, f)$ is s-**sparse** if $\frac{|R_{\text{out}}|}{2^{t \cdot v}} \leq 2^{-s}$. Note sparsity is defined only with respect to the output relation R_{out} and not the function f.

Definition 2.3 (Efficiently Checkable Relation Ensemble)**.** For a relation *ensemble* $\mathcal{R} = \{R_\lambda = (R_{\text{out},\lambda}, f_\lambda)\}_{\lambda \in \mathbb{N}}$ of shifted output relations, we say $\mathcal{R}$ is **efficiently checkable** if there exists a polynomial poly where for all $\lambda \in \mathbb{N}$, function f_λ and relation $R_{\text{out},\lambda}$ are computable in $\text{poly}(\lambda)$ time.

Definition 2.4 (Multi-Input Correlation Intractability for Shifted Output Relations)**.** For functions $t = t(\lambda)$ and $s = s(\lambda, k, v)$ and function family ensemble $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda \in \mathbb{N}}$, we say that a hash function Π_{Hash} with input length $k(\lambda)$ and output length $v = v(\lambda)$ is t-**multi-input correlation intractability (t-MICIH) for $\mathcal{F}$-shifted s-sparse output relations** if it is correlation intractability for *any* t-ary shifted output relation ensemble $\mathcal{R} = \{R_\lambda = (R_{\text{out},\lambda}, f_\lambda)\}$ which satisfy the following properties:

- For all $\lambda \in \mathbb{N}$, $f_\lambda \in \mathcal{F}_\lambda$.
- $\mathcal{R}$ is efficiently checkable.
- For all $\lambda \in \mathbb{N}$, R_λ is $s(\lambda, k, v)$-sparse.

The above definition is stated for arbitrary t and generic family $\mathcal{F}$ (which we show how to construct, see Section 5). However, for our coding result we will require the simple case where $t = 2$ and $\mathcal{F}$ is the family of all linear functions over $\mathbb{F}_2$. We will refer to such as hash as a **linearly shifted MICIH**.

2.2 Standard Error Correcting Codes

We now define error-correcting codes for information theoretic adversaries, also called Hamming channels. In this section, a code is a pair (Enc, Dec) of deterministic encoding and decoding algorithms, and different notions are obtained by considering the requirements of the decoding algorithm.

Definition 2.5 (Standard codes for Hamming channels)**.** Let k, n, q be parameters and let $\text{Enc} : \{0,1\}^k \to [q]^n$ be an invertible function. We say that Enc is an encoding function for a code that:

- p-***corrects***, if there exists a function $\text{Dec} : [q]^n \to \{0,1\}^k$ such that for every $m \in \{0,1\}^k$ and every $v \in [q]^n$ with $\delta(\text{Enc}(m), v) \leq p$, we have $\text{Dec}(v) = m$.
- p-***detects***, if there exists a function $\text{Dec} : [q]^n \to \{0,1\}^k \cup \{\perp\}$ such that for every $m \in \{0,1\}^k$, $\text{Dec}(\text{Enc}(m)) = m$ and for every $v \in [q]^n$ such that $v \neq \text{Enc}(m)$ and $\delta(\text{Enc}(m), v) \leq p$, we have $\text{Dec}(v) = \perp$.
- (p, L)-***list-decodes***, if the function Dec is allowed to output a list of size at most L, and for every $m \in \{0,1\}^k$ and every $v \in [q]^n$ with $\delta(\text{Enc}(m), v) \leq p$, we have $m \in \text{Dec}(v)$.

The rate of the code is the ratio of the message length and output length of Enc, where both lengths are measured in bits. That is the rate $R = \frac{k}{n \cdot \log q}$.

[6] Multi-input relations with this additional restriction $x^{(i)} \neq x^{(j)}$ are often referred to as *distinct-input* relations. For the purpose of this work, all relations we consider will be distinct-input.

A binary code $(\mathsf{Enc}, \cdot)$ is said to be ***linear*** if $\mathsf{Enc}\colon \{0,1\}^k \to \{0,1\}^n$ is a linear map. We say that $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ is the generating matrix of $(\mathsf{Enc}, \cdot)$ if $\mathsf{Enc}(m) = m \cdot \mathbf{G}$ for every $m \in \{0,1\}^k$.

A code (Enc, Dec), is ***explicit*** if Enc and Dec run in polynomial time. (Naturally, this makes sense only for a family of encoding and decoding/detecting functions with varying message length k, block length n, and alphabet sizes q).

The best known construction for list-decoding is given by Guruswami and Rudra [GR08]. This construction achieves the Blokh-Zyablov bound stated in Eq. (2.1). We state their result in the following theorem.

Theorem 2.6 (List decodable code approaching the Blokh-Zyablov bound, [GR08])**.** *For every constant $0 \le p < 1/4$ and $0 < \varepsilon$, and every sufficiently large $k \in \mathbb{N}$ there exists an explicit linear code that is (p, L)-list decodable with rate $R > R_{BZ}(p) - \varepsilon$ and $L = \mathsf{poly}(k)$.*[7]

2.3 Codes For Computationally Bounded Adversaries: Seeded Codes

Seeded codes for computationally bounded channels were first introduced in [GHY20]. Below, we define the syntax for seeded code. Loosely speaking, a seeded code is a triplet of efficient algorithms (Enc, Dec, Setup). Where on input message length k and security parameter λ, $\mathsf{Setup}(1^k, 1^\lambda)$ produces a public "seed" pp that is known to the sender (the encoding function Enc) and receiver (the decoding function Dec) and to any possible adversary. Note that seeded codes do not require any additional assumptions (like secret-coins or public-key infrastructure). All seeded codes we consider in this paper are binary.

As shown in [GHY20, SW25b, SW25a], seeded codes are useful when dealing with computationally bounded adversaries because they enable us to achieve better error-correcting codes than what is possible information theoretically. We note that seeded codes (that is, the presence of a public seed) do not yield any improvements in the rate when considering information-theoretic adversaries. Throughout this paper, when mentioning seeded codes, we implicitly assume that the adversary is computationally bounded, and detection and correction for seeded codes are modeled using the experiments in the definition below.

Definition 2.7 (Seeded codes)**.** A ***Seeded code*** is a triplet of efficient algorithms $\Pi_{\mathsf{scode}} = (\mathsf{Enc}, \mathsf{Dec}, \mathsf{Setup})$ with the following syntax:

- $\mathsf{Setup}(1^\lambda, 1^k) \to \mathsf{pp}$: On input security parameter λ, and message length k, the setup outputs a public parameter (seed) pp.
- Enc is a deterministic function that on input $m \in \{0,1\}^k$ and a seed pp outputs a string $c \in \{0,1\}^{n(\lambda,k)}$. We use $\mathsf{Enc}_{\mathsf{pp}}(m)$ to denote $\mathsf{Enc}(m, \mathsf{pp})$.
- Dec is a deterministic function that on input $z \in \{0,1\}^{n(\lambda,k)}$ and seed pp outputs a string $\widehat{m} \in \{0,1\}^k \cup \{\perp\}$. We use $\mathsf{Dec}_{\mathsf{pp}}(z)$ to denote $\mathsf{Dec}(z, \mathsf{pp})$.

We will look at codes satisfying the following properties:

- **Perfect Reliability:** For all $\lambda, k \in \mathbb{N}$, pp in the support of $\mathsf{Setup}(1^\lambda, 1^k)$ and $m \in \{0,1\}^k$, it holds that $\mathsf{Dec}_{\mathsf{pp}}(\mathsf{Enc}_{\mathsf{pp}}(m)) = m$.
- **Computational p-Detection/Correction:** For a security parameter $\lambda \in \mathbb{N}$ and a $p \in [0,1]$, the (computational) error *detection* experiment $\mathsf{ExpDetect}^{(p)}(\mathcal{A}, \lambda)$ between an adversary and challenger is as follows:
  1. On input security parameter 1^λ, the adversary $\mathcal{A}$ sends an input length 1^k to the challenger.
  2. The challenger samples public parameters $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$ which it returns to the adversary.
  3. The adversary $\mathcal{A}$ sends a message $m \in \{0,1\}^k$ and string $z \in \{0,1\}^n$ to the challenger.
  4. The output of the experiment is 1 if $\delta(\mathsf{Enc}_{\mathsf{pp}}(m), z) \le p \wedge \mathsf{Dec}_{\mathsf{pp}}(z) \notin \{m, \perp\}$.

[7] The result in [GR08] is stated for a family of codes. However, a careful examination of the construction and proof of [GR08], shows that it is possible to achieve this rate for every sufficiently large $k \in \mathbb{N}$.

We say that Π_{scode} p-detects if for all polynomial time $\mathcal{A}$, there exists $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$

$$\mathbf{Pr}[\mathsf{ExpDetect}^{(p)}(\mathcal{A}, \lambda)] = \mathsf{negl}(\lambda).$$

We define the *p-correction* experiment $\mathsf{ExpCorrect}^{(p)}(\mathcal{A}, \lambda)$ identically to above, except

4. The output of the experiment is 1 if $\delta(\mathsf{Enc}_{\mathsf{pp}}(m), z) \leq p \wedge \mathsf{Dec}_{\mathsf{pp}}(z) \neq m$.

and similarly, we say Π_{scode} p-corrects if for all polynomial time $\mathcal{A}$, there exists $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$

$$\mathbf{Pr}[\mathsf{ExpCorrect}^{(p)}(\mathcal{A}, \lambda)] = \mathsf{negl}(\lambda).$$

- **Rate:** We define the rate of the code to be the quantity $\frac{k}{n(\lambda,k)}$.
- **Public Coin:** We will say a seeded code is *public coin* if the output $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$ is a uniformly random string $\in \{0,1\}^{l(\lambda,k)}$.

3 Our Result for Seeded Codes: Theorem Statement

We now state our main result for seeded codes. As discussed in the introduction, these codes are the first construction that achieves *adaptive* security from standard cryptographic assumptions.

Our first result, is a rate 1 seeded code that *detects* from $1/2 - \varepsilon$-relative errors for any constant $0 < \varepsilon$.

Theorem 3.1. *Assuming any one of the following assumptions DDH, DCR, LWE or QR: There exists a polynomial q such that for every constant $p \in [0, 1/2)$, there exists a seeded code with p-detection and rate $R(k, \lambda) = 1 - q(\lambda)/k$. In particular for a sufficiently large polynomial $k(\lambda)$, the rate is $1 - o(1)$. Furthermore, if assuming either DDH or LWE, this seeded code is public coin.*

Our second result is a seeded code that achieves *unique* decoding against computationally bounded adversaries that induce $p \in [0, 1/4)$ relative errors, with a rate that matches the Blokh-Zyablov bound $R_{BZ}(p)$ (see, Eq. (2.1) for the exact bound) which is the rate of the best known explicit construction for list decodable codes.

Theorem 3.2. *Assuming any one of the following assumptions DDH, DCR, LWE or QR: There exists a polynomial q such that for every constant $p \in [0, 1/4)$ and $0 < \varepsilon$, there exists a seeded code with p-correction and rate $R(k, \lambda) = R_{BZ}(p) - \varepsilon - q(\lambda)/k$. In particular for a sufficiently large polynomial $k(\lambda)$, the rate is $R_{BZ}(p) - \varepsilon$. Furthermore, if assuming either DDH or LWE, this seeded code is public coin.*

Remark 3.3 (Sub-exponential Security)**.** Our constructions in Theorems 3.1 and 3.2 can be based on any one of the following cryptographic assumptions: DDH, DCR, LWE, or QR. If the chosen assumption holds with sub-exponential security, then the resulting codes inherit sub-exponential security—i.e., they have sub-exponentially small failure probability against sub-exponential-size channels. This follows because all our reductions run in polynomial time and incur only a polynomial loss in advantage. Nevertheless, for simplicity, we present our formal statements and proofs under the standard polynomial-time (negligible) security assumption.

Proof of Theorems 3.1 and 3.2. The proof follows directly from our results in the following sections. Specifically, in Theorems 4.1 and 4.2 from Section 4, we show that assuming correlation intractable hash for any sublinearly output sparse relation family over linear shifts $\mathcal{R}$, there exists seeded codes that exactly match the stated results in Theorems 3.1 and 3.2 respectively. In Section 5 we give a generic construction of correlation intractable hashing for sublinearly output sparse relation families from trapdoor hashing (see Theorem 5.10, for the formal statement). This can then be instantiated from the existing construction trapdoor hashing for linear predicates of [DGI+19a], which can be realized with any of the following standard assumptions: DDH, DCR, LWE, or QR Theorem 5.2. The proof of Theorems 3.1 and 3.2 is immediate from these results.

4 Seeded Codes from 2-Input Correlation Intractable Hashing

In this section, we show how to construct seeded codes using two input correlation intractability

Theorem 4.1. *Assume there exists a 2-MICIH Π_{Hash} for linearly shifted s-sparse output relations for some polynomial s. Then there exists a polynomial q such that for every constant $p \in [0, 1/2)$, there exists a seeded code with p-detection and rate $R(k, \lambda) = 1 - q(\lambda)/k$. Furthermore, if Π_{Hash} is public coin, then the seeded code is as well.*

Our second result is a seeded code that achieve *unique* decoding against computationally bounded adversaries that induce $p \in [0, 1/4)$ with a rate that matches the Blokh-Zyablov bound $R_{BZ}(p)$ (see, Eq. (2.1) for the exact bound) which is the rate of the best known explicit construction for list decodable codes.

Theorem 4.2. *Assume there exists a 2-MICIH Π_{Hash} for linearly shifted s-sparse output relations for some polynomial s. There exists a $q \in \mathsf{poly}$ such that for every constant $p \in [0, 1/4)$ and $\varepsilon' > 0$, there exists a seeded code with rate $R(k, \lambda) = R_{BZ}(p) - \varepsilon' - q(\lambda)/k$ that p-corrects. Furthermore, if Π_{Hash} is public coin, then our seeded code is as well.*

Remark 4.3 (Required sparsity for the reduction)**.** For simplicity, in our theorem statements above, we assume a sparsity parameter of $s = \mathsf{poly}(\lambda)$, notably independent of the output length v (which is what our constructions of MICIH in Theorem 5.10 achieves). However, in our actual construction (see Lemma 4.7), it suffices for Π_{Hash} to be correlation intractable for sparse relations with sublinear dependence on v (specifically, $s = v^{\beta} \cdot \mathsf{poly}(\lambda)$ with $\beta < 1$).

In Section 4.1 we define a property of linear binary codes, which we call decomposable codes with Column-Robustness. We then show in Section 4.2, that any linear code that satisfies this property implies seeded codes with essentially 1/2 error detection. Finally, in Section 4.2.1 we show a generic way to achieve error correction if these codes also have efficient list-decoding. The proof of these theorems then follows by appropriately instantiating our main technical lemma Lemma 4.7, which we show in Section 4.2.2.

4.1 Decomposable Binary Linear Codes

For our construction presented in Section 4.2, we require as an ingredient a linear codes that satisfies the following: There exists an efficient (possibly randomized) algorithm GenM, than on input k and v, returns a generating matrix $\mathbf{G}$ which we view as being composed of two part $\mathbf{A} \in \mathbb{F}_2^{k \times n}$ and $\mathbf{B} \in \mathbb{F}_2^{v \times n}$ such that $\mathbf{G} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$. (Note that any linear code with message length $k + v$, can be viewed in this manner. However, for the sake simplicity and to make the our choice of parameters more explicit, we index the matrix $\mathbf{G}$ with both k and v.) We will demand the following property which we call *column-robustness*. Loosely speaking, the property states that if we chose a (sufficiently small) random subset of the column of matrix $\mathbf{B}$, these columns will be independent with noticeable probability. As we will show in the rest of this section, this property (together with an appropriate 2-input CI hash function) is sufficient for showing error detection for almost $\frac{1}{2}$-relative errors.

Definition 4.4 (Decomposable Binary Linear Codes With Column-Robustness)**.** We say that an efficient (possibly randomized) function $\mathsf{GenM}(\cdot, \cdot)$ is a decomposable binary linear code if it satisfies the following: On input message length $k \in \mathbb{N}$ and nested message length $v \in \mathbb{N}$ such that $\log^5(k) < v < k^{0.49}$:[8]

- $\mathsf{GenM}(1^k, 1^v) \to \mathbf{G}$: The GenM algorithm outputs a generating matrix $\mathbf{G} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$, where $\mathbf{A} \in \mathbb{F}_2^{k \times n}$, $\mathbf{B} \in \mathbb{F}_2^{v \times n}$, and $\mathbf{G}$ has full-rank.

We use $\mathsf{Enc}_{\mathbf{G}}$ to denote the encoding function $\mathsf{Enc}_{\mathbf{G}}((m, y) \in \{0,1\}^{k+v}) = (m, y) \cdot \mathbf{G}$, and consider the following properties:

- **Rate:** We take the sum of the message and nested message lengths when computing the rate $R = \frac{k+v}{n}$.

[8] The restriction that $\log^5(k) < v < k^{0.49}$, is put in place to avoid edge cases. For our coding application, the function GenM would not be invoked on v that does not satisfy this condition.

- **Column-Robustness:** Let $q : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a polynomial and $\alpha > 0$ be a constant. We say a matrix $\mathbf{B}$ is (q, α)-well formed if for all $v, k \in \mathbb{N}$, sampling a random $T \subset [n]$ where $|T| = w := v^{\alpha}$.

$$\Pr_{T}[\mathsf{rank}([\mathbf{B}]_T) = w] \geq \frac{1}{q(k, v)}.$$

  We say our decomposable code is α-column-robust if there exists q such that

$$\Pr_{\mathbf{G}=\left[\begin{smallmatrix}\mathbf{A}\\ \mathbf{B}\end{smallmatrix}\right] \leftarrow \mathsf{GenM}(1^k, 1^v)} [\mathbf{B} \text{ is } (q, \alpha)\text{-well formed}] = 1 - \mathsf{negl}(v).$$

- **List Decodability:** For a constant $p \in [0, 1/2)$ and polynomial L, we say that the code is (p, L)-list decodable if there exists an efficient function Dec such that for every $k, v \in \mathbb{N}$ and every $\mathbf{G}$ in the support of $\mathsf{GenM}(1^k, 1^v)$ the standard code $(\mathsf{Enc}_{\mathbf{G}}, \mathsf{Dec}_{\mathbf{G}})$ is (p, L)-list decodable (see Definition 2.5).

The following theorem states that there are decomposable codes with rate $R = 1$ that are 1-column-robust. The proof of Theorem 4.5 stated below is given in Appendix A.1. Loosely speaking, we show that a random code (that is, sampling the generating matrix $\mathbf{G}$ uniformly) yields a decomposable code. Note that in term of rate and column-robust, this code is optimal. The only property missing from the theorem below is efficient list-decoding.

Theorem 4.5 (Random Linear Codes are Decomposable)**.** *There exists a decomposable binary linear code that is* 1*-column-robust with rate* $R = 1$.

The following theorem, states that there is a decomposable binary linear code that is column-robustness, list-decodable, and has a rate that approaches the Blokh-Zyablov bound. The proof for the theorem is given in Appendix A.2. Loosely speaking, we show that the list-decodable codes of Guruswami Rudra [GR08], which is the best know explicit list-decodable code construction, (with a rate that approaches the Blokh-Zyablov bound) also satisfies column-robustness.[9]

Theorem 4.6 (List-Decodable Codes)**.** *For any constant* $\alpha < 1$ *and every* $p \in [0, 1/2)$ *and* $0 < \varepsilon$*, there exists a decomposable binary linear code that is* α*-column-robust, with rate* $R > R_{BZ}(p) - \varepsilon$ *and efficient* (p, L)*-list decoding for some fixed* $L \in \mathsf{poly}(k)$.

In Section 4.2, we show that for any $p \in [0, \frac{1}{4})$ given a linear code with column-robustness that can list-decode from p-fraction of errors and has rate $R(p)$, we can construct a seeded code that can *uniquely* p-correct, with essentially that same rate $R(p)$.

We stress that if in the future, better linear efficiently list-decodable codes that have (column-robustness) are constructed, then these codes would immediately yield an improvement in our results. In particular, a random linear code would achieve optimal parameters and will have the above column-robustness property (as demonstrated in Theorem 4.5), but it does not have an efficient list-decoding algorithm. However, as of now, the best explicit construction for list-decodable codes remains the construction of Guruswami and Rudra [GR08] (that achieves the Blokh-Zyablov bound), which we show that it has column-robustness.

4.2 Construction and Proof for Seeded Codes

Using the decomposable codes of Section 4.1 and correlation intractable hashing Definition 2.1, we are now ready to present a generic construction in Fig. 1 constructs seeded codes. We will then show in our main technical lemma, Lemma 4.7, the properties our seeded code achieves.

Lemma 4.7. *Let* $\alpha > \beta \geq 0$ *be constants and* $q \in \mathsf{poly}(\lambda)$*. Suppose we had the following ingredients:*

- *A decomposable binary linear code* GenM *with rate* $R' = R'(k, v)$ *and* α*-column-robustness.*

[9] In fact, we show something more general. Informally, we prove that any linear code that is produced via code concatenation satisfies column-robustness as long as the outer code has sufficient dual distance. When viewed in a particular order, the codes of [GR08] satisfy this property.

Figure 1: Construction For Seeded Codes With Detection.

Ingredients and parameters: Let $\lambda, k \in \mathbb{N}$ denote the security parameter and input length respectively.

- A decomposable binary linear code $\Pi_{\mathsf{code}} = \mathsf{GenM}$ with α-column-robustness for some constant $\alpha > 0$.
- A 2-MICIH Π_{Hash} for the family of $v^\beta \cdot q(\lambda)$-sparse linearly shifted relations for some constant $\beta \in (0, \alpha)$ and $q \in \mathsf{poly}(\lambda)$.

We set $v = (\lambda \cdot q(\lambda))^{\frac{1}{\alpha-\beta}}$. Since $\alpha > \beta \Leftrightarrow \frac{1}{\alpha-\beta} > 0$, this is a well defined fixed polynomial in λ.

Setup algorithm: We define a setup function $\mathsf{Setup} : \{0,1\}^\lambda \times \{0,1\}^k \to \{0,1\}^{\ell(\lambda,k)}$:

Input: Security parameter and message length 1^λ and 1^k.

Output: A public seed $\mathsf{pp} \in \{0,1\}^{\ell(\lambda,k)}$.

Operation :

- Generate a hash key: $\mathsf{hk} \leftarrow \mathsf{Gen}(1^\lambda, 1^k, 1^v)$.
- Compute generating matrix: $\mathbf{G} \leftarrow \mathsf{GenM}(1^k, 1^v)$.
- Output $\mathsf{pp} = (\mathbf{G}, \mathsf{hk})$.

Encoding algorithm: We define an encoding function $\mathsf{Enc} : \{0,1\}^k \times \{0,1\}^{\ell(\lambda,k)} \to \{0,1\}^{n(\lambda,k)}$:

Input: A message $m \in \{0,1\}^k$ and public parameters $\mathsf{pp} = (\mathbf{G}, \mathsf{hk})$.

Output: A codeword $c \in \{0,1\}^n$.

Operation :

- Hash the message: Compute $y = \mathsf{Hash}(\mathsf{hk}, m)$.
- Encode the message and the hash: Output $c = (m, y) \cdot \mathbf{G}$.

Decoding algorithm: We define a decoding function $\mathsf{Dec} : \{0,1\}^{n(\lambda,k)} \times \{0,1\}^{\ell(\lambda,k)} \to \{0,1\}^k \cup \{\perp\}$:

Input: A received a string $z \in \{0,1\}^n$ and public parameters $\mathsf{pp} = (\mathbf{G}, \mathsf{hk})$.

Output: A message $\widehat{m} \in \{0,1\}^\lambda \cup \{\perp\}$.

Operation :

- Recall that since G is a matrix with full row rank $(k+v)$. Compute right inverse $\mathbf{G}^{-1_R}$ and set $(\widehat{m}, \widehat{y}) = z \cdot \mathbf{G}^{-1_R}$ where $(\widehat{m}, \widehat{y}) \in \{0,1\}^k \times \{0,1\}^v$.
- Check consistency: If $\mathsf{Hash}(\mathsf{hk}, \widehat{m}) = \widehat{y}$ output $\widehat{m}$. Otherwise output $\perp$.

- *A hash function Π_{Hash} which is correlation intractable against a two-input $v^\beta \cdot q(\lambda)$-sparse linearly shifted relation family.*

Then the seeded code $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Setup})$ *described in Fig. 1 satisfies the following properties:*

- *is perfectly reliable.*
- *has rate $R(k,\lambda) \geq R'(k,v) - \frac{v}{k}$ for some $v(\lambda) \in \mathsf{poly}(\lambda)$ (independent of k).*
- *if Π_{Hash} is public coin, then the seeded code is public coin as well.*

- *for any constant* $p \in [0, \frac{1}{2})$*, if* GenM *is* (L, p)*-list decodable, then for every* $k, \lambda \in \mathbb{N}$ *and every* pp *in the support of* $\mathsf{Setup}(1^k, \lambda)$*,* $(\mathsf{Enc}_{\mathsf{pp}}, \mathsf{Dec}_{\mathsf{pp}})$ *is* (L, p)*-list decodable.*
- *for any constant* $p \in [0, \frac{1}{2})$*,* p*-detects.*

Proof of Lemma 4.7. We argue each of these properties individually.

Reliability. Note that GenM always outputs a full rank matrix $\mathbf{G}$. As such, there exists a right inverse $\mathbf{G}^{-1_R}$ which is computed by the decode algorithm. Then observe decode compute $(\widehat{m}, \widehat{y}) = (m, y) \cdot \mathbf{G} \cdot \mathbf{G}^{-1_R} = (m, y) \cdot I_{k+v} = (m, y)$. Since $y = \mathsf{Hash}(\mathsf{hk}, m)$, the consistency check passes, which implies that the seeded code in Fig. 1 is perfectly reliable.

Rate. The proof for the rate follows directly from inspection. By construction, the codeword is simply a codeword of the generating matrix $\mathbf{G}$ on message m and nested message $\mathsf{Hash}(\mathsf{hk}, m)$, where the output length of Hash is set to $v = q(\lambda)^{\frac{1}{\alpha-\beta}}$, which by definition is a fixed polynomial in λ. Thus, we can compute the rate of our new code as $R'(k, v) \cdot \frac{k}{k+v} \geq R'(k, v) - v/k$.

Public Coin. To prove the last item, recall that our seed pp simply consists of the hash key hk and a generating matrix $\mathbf{G}$. Since we only require information-theoretic properties on our decomposable code, we can always represent $\mathbf{G}$ via the randomness of GenM. Thus, our code is public coin exactly when our hash key hk is.

List Decodability. Recall for every fixed $k, \lambda \in \mathbb{N}$, every $\mathsf{pp} = (\mathbf{G}, \mathsf{hk})$ in the support of $\mathsf{Setup}(1^\lambda, 1^k)$, any codeword of $(\mathsf{Enc}_{\mathsf{pp}}, \mathsf{Dec}_{\mathsf{pp}})$ is simply a codeword of the generating matrix $\mathbf{G}$ on (m, y) where $y = \mathsf{Hash}(\mathsf{hk}, m)$ is only a function of m when hk is fixed. This implies that $(\mathsf{Enc}_{\mathsf{pp}}, \mathsf{Dec}_{\mathsf{pp}})$ is (p, L)-list decodable if $(\mathsf{Enc}_{\mathbf{G}}, \mathsf{Dec}_{\mathbf{G}})$ is (p, L)-list decodable.

Error Detection. The bulk of our technical argument will now follow in showing that our code p-detects for any constant $p < \frac{1}{2}$. To argue this, we will define a sequence of experiments as follows:

- $\mathsf{Exp}_{\mathsf{real}}$: This is the real error detection experiment instantiated with our scheme
  1. On input security parameter λ, the adversary $\mathcal{A}$ sends an input length 1^k to the challenger.
  2. The challenger samples $\mathsf{hk} \leftarrow \mathsf{Gen}(1^\lambda, 1^k)$, $\mathbf{G} \leftarrow \mathsf{GenM}(1^k, 1^v)$ which it sends to $\mathcal{A}$.
  3. The adversary $\mathcal{A}$ sends a message $m \in \{0,1\}^k$ and string $z \in \{0,1\}^n$ to the challenger.
  4. The experiment checks the following, outputting 1 iff all are true:
     (a) Compute $y = \mathsf{Hash}(\mathsf{hk}, m)$, $c = (m, y) \cdot \mathbf{G}$ and check $\delta(c, z) \leq p$.
     (b) Recover $(m', y') = z \cdot \mathbf{G}^{-1_R}$ check $\mathsf{Hash}(\mathsf{hk}, m') = y'$.
     (c) Check $m' \neq m$.
- Exp_1: In this experiment, rather than directly checking $\delta(\mathsf{Enc}_{\mathsf{pp}}(m), z) \leq p$, our challenger instead 'subsamples' subsets of indices of the codewords output by the adversary, and checks relative hamming distance δ on these subsets instead.
  4. Set $p' = \frac{2p+1}{4}$. The challenger samples a random $T \subseteq [n] : |T| = w = v^\alpha$. The experiment checks the following are true, outputting 1 iff all are true:
     (a) Compute $y = \mathsf{Hash}(\mathsf{hk}, m)$, $c = (m, y) \cdot \mathbf{G}$, and check $\delta([c]_T, [z]_T) \leq p'$.
     ...
- Exp_2: In this experiment, the experiment aborts with output 0 if the rank of $[\mathbf{B}]_T$ on all subsampled index sets is not full.
  4. Set $p' = \frac{2p+1}{4}$. The challenger samples a random $T \subseteq [n] : |T| = w = v^\alpha$. The experiment checks the following are true, outputting 1 iff all are true:
     ...

(d) $\mathsf{rank}([\mathbf{B}]_T) = w$.

Lemma 4.8. *For all adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that*

$$\mathbf{Pr}[\mathsf{Exp}_1(\mathcal{A}, \lambda) = 1] \geq \mathbf{Pr}[\mathsf{Exp}_{\mathsf{real}}(\mathcal{A}, \lambda) = 1] - \mathsf{negl}(\lambda)$$

Proof. To prove this lemma, we will make use of a simple Chernoff inequality:

Claim 4.9. *Let p, p' be constants $\in (0, 1)$ where $p < p'$. Let $S = \{s_0, s_1, \ldots s_m\}$ be a set of $m = m(\lambda)$ negatively associated Bernoulli random variables with $E[s_i] = p$ for $i \in [m]$. Then*

$$\mathbf{Pr}\left[\sum S \geq mp'\right] = \exp(-O(m)).$$

Proof. This follows from a straightforward application of the (multiplicative) Chernoff bound. As $E[\sum S] = m \cdot p$ by linearity, we can bound

$$\mathbf{Pr}\left[S \geq mp' = mp \cdot \frac{p'}{p}\right] \leq \exp\left(-\frac{\left(\frac{p'-p}{p}\right)^2 \cdot m \cdot p}{3}\right).$$

Since $\frac{(p'-p)^2}{3p}$ is simply a (positive) constant, this completes the proof of our claim. □

Observe that these experiments are identical up until the output of the experiment is computed. Thus, the output distribution (m, z) of the adversary is identical. Now suppose $\mathcal{A}$ outputs some (m, z) which succeeds in $\mathsf{Exp}_{\mathsf{real}}$. By check (a), this means $\delta(\mathsf{Enc}(\mathsf{pp}, m), z) \leq p$. Applying Claim 4.9 on our set T, we can see that $\mathbf{Pr}[\delta([\mathsf{Enc}(\mathsf{pp}, m)]_T, [z]_T) \leq p'] = \mathsf{negl}(\lambda)$, giving us that the probability (m, z) does not succeed in Exp_1 is negligible. □

Lemma 4.10. *Assuming Π_{code} is α-column-robust, then for all adversaries $\mathcal{A}$, there exists a polynomial $q'(\lambda)$ and a negligible function $\mathsf{negl}(\cdot)$ such that*

$$\mathbf{Pr}[\mathsf{Exp}_2(\mathcal{A}, \lambda) = 1] \geq \frac{1}{q'} \cdot \mathbf{Pr}[\mathsf{Exp}_1(\mathcal{A}, \lambda) = 1] - \mathsf{negl}(\lambda).$$

Proof. By column-robustness of Π_{code}, with probability $1 - \mathsf{negl}(\lambda)$, $\mathbf{B}$ is a (q_0, α)-well formed matrix. Thus, by definition, the probability $[\mathbf{B}]_T$ is full rank is at least $\frac{1}{q_0(k,v)}$. Since $\mathcal{A}$ is a poly-time adversary, k, v are bound by some polynomial in λ, so this can be bound by some $q_0(k, v)$ and can be bound by some $q'_0(\lambda)$. Since T is sampled uniformly and independent of the view of the adversary, our lemma statement follows immediately. □

Lemma 4.11. *Assuming Π_{Hash} is 2-input correlation intractable for $(v^\beta \cdot q(\lambda))$-sparse linearly shifted output relations, then for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that*

$$\mathbf{Pr}[\mathsf{Exp}_2(\mathcal{A}, \lambda) = 1] = \mathsf{negl}(\lambda)$$

Proof. Suppose there exists adversary $\mathcal{A}$ which can succeed in Exp_2 with nonneglible probability. Then the following is a reduction $\mathcal{B}$ which succeeds with nonneglible probability in the correlation intractability game on Π_{Hash}:

1. Reduction $\mathcal{B}$ receives security parameter, and begins simulating $\mathcal{A}$, receiving challenge input length 1^k, setting parameters n, v as in Fig. 1 and computing $\mathbf{G} = \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \leftarrow \mathsf{GenM}(1^k, 1^v)$.
2. Setting $p' = \frac{2p+1}{4}$, $\mathcal{B}$ samples a random $T \subseteq [n] : |T| = w = v^\alpha$. $\mathcal{B}$ checks that $\mathsf{rank}([\mathbf{B}]_T) = w$, and aborts with output 0 otherwise.
3. $\mathcal{B}$ defines the two-input shifted output relation (R, f) as follows:

- On inputs $y^{(0)}, y^{(1)}$, output relation $R(y^{(0)}, y^{(1)})$ computes $y'^{(0)} = y^{(0)} \cdot [\mathbf{B}]_T$ and $y'^{(1)} = y^{(1)} \cdot [\mathbf{B}]_T$. Then, it checks $\delta(y'^{(0)}, y'^{(1)}) \leq p'$ (i.e. the relative hamming weight of the first w entries of y') as the output of the relation.
- Define linear function $f(m) = m \cdot [\mathbf{A}]_T \cdot {[\mathbf{B}]_T}^{-1_L}$.

Recall that since we chose $[\mathbf{B}]_T$ to have full column rank w, the left inverse ${[\mathbf{B}]_T}^{-1_L}$ is guaranteed to exist. $\mathcal{B}$ sends $1^k, 1^v$, and (R, f) to the correlation intractability challenger, who responds with hash key hk.

4. $\mathcal{B}$ sends public parameters $\mathsf{pp} = (\mathbf{G}, \mathsf{hk})$ to $\mathcal{A}$, who responds with (m, z).
5. $\mathcal{B}$ recovers $\Pi_{\mathsf{code}}.\mathsf{Dec}(z) = (m', y')$ and sends (m, m') to the challenger.

First, we observe that if $\mathcal{B}$ accurately simulates the view of Exp_2 for $\mathcal{A}$, only aborting when $\mathsf{rank}([\mathbf{B}]_T) \neq w$, in which case Exp_2 always outputs 0 as well.

Claim 4.12. *The relation ensemble $\mathcal{R} = \{(R_{\mathsf{out},\lambda}, f_\lambda)\}$ defined by $\mathcal{B}$ is a $(v^\beta \cdot q(\lambda))$-sparse linearly shifted relation ensemble.*

Proof. Recall from Definition 2.2, for (R_{out}, f) to fall into this relation ensemble, we require the following:

- f **is a linear function:** Recall we define $f(m) = m \cdot [\mathbf{A}]_T \cdot {[\mathbf{B}]_T}^{-1_L}$. This is linear by inspection.
- R_{out} **is efficiently checkable:** By definition, R_{out} simply performs a matrix multiplication and followed by bit counting, both of which are efficient.
- **When R_{out} is viewed as a set $\in (\{0,1\}^v)^2$, $\frac{R}{2^{2v}} \leq 2^{-v^\beta \cdot q(\lambda)}$:** This is equivalent to showing that

$$\Pr_{y, y' \xleftarrow{R} \{0,1\}^v}[R_{\mathsf{out}}(y, y') = 1] \leq 2^{-v^\beta q(\lambda)}$$

To see this, recall R_{out} begins by multiplying by matrix $[\mathbf{B}]_T$. Since $[\mathbf{B}]_T$ is full rank and thus surjective, the distribution of $y \cdot [\mathbf{B}]_T$ for $y \xleftarrow{R} \{0,1\}^v$ is uniform. So it suffices to show the relation $\delta(y'^{(0)}, y'^{(1)}) \leq p'$ is sufficiently sparse on random y'. We can show this is true for any $p' < \frac{1}{2}$ by a simple Chernoff bound analogous to Claim 4.9. Fixing an arbitrary $y'^{(0)}$, the probability a uniform $y'^{(1)}$ agrees with $y'^{(0)}$ on any particular bit is independently $\frac{1}{2}$. Thus, we can bound the probability it agrees on less than a p' fraction of w bits as:

$$Pr\left[\delta(y'^{(0)}, y'^{(1)}) \leq p' = 2p' \cdot \frac{1}{2}\right] \leq \exp\left(-\frac{(1-2p')^2 \cdot w \cdot \frac{1}{2}}{2}\right).$$

Since $\frac{(1-2p')^2}{4}$ is simply a (positive) constant, we get that our relation is $2^{-\Omega(w)}$ sparse. Substituting

$$w = v^\alpha = ((\lambda \cdot q(\lambda))^{\frac{1}{\alpha-\beta}})^\alpha = (\lambda \cdot q(\lambda))^{\frac{\alpha}{\alpha-\beta}} = (\lambda \cdot q(\lambda))^{\frac{\beta}{\alpha-\beta}} \cdot \lambda \cdot q(\lambda) = v^\beta \cdot \lambda \cdot q(\lambda)$$

gives sparsity $2^{-\Omega(v^\beta q(\lambda) \cdot \lambda)} \ll 2^{-v^\beta \cdot q(\lambda)}$, which completes the proof of our claim.

□

Claim 4.13. *On any output (m, z) for which $\mathcal{A}$ succeeds in Exp_2, $\mathcal{B}$ succeeds in the correlation-intractability game.*

Proof. Our shifted output relation checks $\delta(y'^{(0)}, y'^{(1)}) \leq p'$ where (applying our shift) $y' = (y - f(m)) \cdot [\mathbf{B}]_T$. Thus we can simplify

$$\begin{aligned}
y' &= (y - f(m)) \cdot [\mathbf{B}]_T \\
&= y \cdot [\mathbf{B}]_T - f(m) \cdot [\mathbf{B}]_T \\
&= y \cdot [\mathbf{B}]_T - m \cdot [\mathbf{A}]_T \cdot {[\mathbf{B}]_T}^{-1_L} \cdot [\mathbf{B}]_T \\
&= y \cdot [\mathbf{B}]_T - m \cdot [\mathbf{A}]_T \cdot I_n \\
&= y \cdot [\mathbf{B}]_T - m \cdot [\mathbf{A}]_T \\
&= (m, y) \cdot [\mathbf{G}]_T
\end{aligned}$$

Recall for an adversary to succeed in Exp_2, it must be the case that $(m', y') = z \cdot \mathbf{G}^{-1_R}$ for some m' where $\mathsf{Hash}(\mathsf{hk}, m') = y'$ and $m \neq m'$, meaning $(m', y') \cdot \mathbf{G}$ must exactly be codeword $\mathsf{Enc}_{\mathsf{pp}}(m')$. From above, we can see any such m, m' which produce $\delta([\mathsf{Enc}_{\mathsf{pp}}(m)]_T, [\mathsf{Enc}_{\mathsf{pp}}(m')]_T) \leq p'$ breaks our correlation-intractable hash. □

Taking together these above two observations, we can see that $\mathcal{B}$ is a valid reduction. □

Taking together Lemmas 4.8, 4.10 and 4.11 gives us the proof of p-detection. □

This concludes the proof of Lemma 4.7.

4.2.1 A Generic Approach to Error Correction

The following lemma says that for every $p \in [0, 1/4)$ if we have a seeded code $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Setup})$ that detects from $2p$-relative errors, and at the same time for every fixed seed pp, $\mathsf{Enc}_{\mathsf{pp}}$ has an efficient list-decoding algorithm, then there is a generic way to *uniquely* decode from p-relative errors.

While the following lemma does not appear explicitly in prior results [GHY20, SW25b, SW25a]. It is the guiding idea and intuition in all prior constructions. We make this intuition formal in the lemma below. We mention that [GHY20] gives a similar result in spirit, where instead of error detection, [GHY20] use the notion of pseudodistance, which for seeded codes can be shown to be equivalent to detection.

Lemma 4.14. *[Detection + List-decoding → Unique-decoding]. Let $p \in [0, 1/4)$ be a constant and let $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Setup})$ be a seeded code. If the following holds:*

- $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Setup})$ *$2p$-detects.*
- *There exists a polytime function $\mathsf{Dec}^{\mathsf{List}}$ and a polynomial $L = L(\lambda, k)$, such that for every $\lambda, k \in \mathbb{N}$ and every pp in the support of $\mathsf{Setup}(1^\lambda, 1^k)$, $(\mathsf{Enc}_{\mathsf{pp}}, \mathsf{Dec}^{\mathsf{List}}_{\mathsf{pp}})$ is a (ρ, L)-list-decodable code (see Definition 2.5).*

then, there exists a function Dec' such that $(\mathsf{Enc}, \mathsf{Dec}', \mathsf{Setup})$ p-corrects.

Proof of Lemma 4.14. Given $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Setup})$ and $\mathsf{Dec}^{\mathsf{List}}$, we define Dec' as follows: On input $z \in \{0,1\}^{n(1^\lambda, 1^k)}$ and pp, compute $\mathsf{List} = \mathsf{Dec}^{\mathsf{List}}_{\mathsf{pp}}(z)$, and output the first $\widehat{m} \in \mathsf{List}$ for which $\delta(\mathsf{Enc}_{\mathsf{pp}}(\widehat{m}), z) \leq p$ and $\mathsf{Dec}_{\mathsf{pp}}(\mathsf{Enc}_{\mathsf{pp}}(\widehat{m})) = \widehat{m}$ (if no such $\widehat{m}$ exists output $\perp$).

Assume towards contradiction that $(\mathsf{Enc}, \mathsf{Dec}', \mathsf{Setup})$ does not achieve p-correction. That is, there exists a ppt adversary $\mathcal{A}$ and a polynomial $q \in \mathsf{poly}$, and for infinitely many $\lambda \in \mathbb{N}$: $1^k \leftarrow \mathcal{A}(\lambda)$ and

$$\Pr_{\substack{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^k) \\ (m,z) \leftarrow \mathcal{A}(\mathsf{pp})}} \left[\delta(\mathsf{Enc}_{\mathsf{pp}}(m), z) \leq p \wedge \mathsf{Dec}'_{\mathsf{pp}}(z) \neq m\right] \geq q \tag{4.1}$$

We define a ppt adversary $\mathcal{A}'$ (using $\mathcal{A}$) that contradicts that $2p$-detection property of $(\mathsf{Enc}, \mathsf{Dec}, \mathsf{Setup})$ as follows: On input pp, $\mathcal{A}'$ runs $(m, z) \leftarrow \mathcal{A}(\mathsf{pp})$, computes $\widehat{m} = \mathsf{Dec}'_{\mathsf{pp}}(z)$, $\widehat{c} = \mathsf{Enc}_{\mathsf{pp}}(\widehat{m})$ and outputs $(m, \widehat{c})$. To complete the proof, we need to show that:

$$\Pr_{\substack{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda, 1^k) \\ (m,\widehat{c}) \leftarrow \mathcal{A}'(\mathsf{pp})}} \left[\delta(\mathsf{Enc}_{\mathsf{pp}}(m), \widehat{c}) \leq 2p \wedge \mathsf{Dec}_{\mathsf{pp}}(\widehat{c}) \notin \{m, \perp\}\right] \geq q \tag{4.2}$$

Consider the above experiment, and let $\mathsf{List} = \mathsf{Dec}^{\mathsf{List}}_{\mathsf{pp}}(z)$, $\widehat{m} = \mathsf{Dec}'_{\mathsf{pp}}(z)$ and $\widehat{c} = \mathsf{Enc}_{\mathsf{pp}}(\widehat{m})$. If $\delta(\mathsf{Enc}_{\mathsf{pp}}(m), z) \leq p$ then by definition it follows that $m \in \mathsf{List}$, which implies that $\widehat{m} \neq \perp$ and that $\delta(\mathsf{Enc}_{\mathsf{pp}}(m), \widehat{c}) \leq 2p$ (this holds by the triangular inequality, since by assumption $\delta(\mathsf{Enc}_{\mathsf{pp}}(m), z) \leq p$ and by construction $\delta(\mathsf{Enc}_{\mathsf{pp}}(\widehat{m}), z) \leq p$)). Put together, we get that if $\delta(\mathsf{Enc}_{\mathsf{pp}}(m), z) \leq p$ and $\mathsf{Dec}'_{\mathsf{pp}}(z) = \widehat{m} \neq m$ then it follows that $\delta(\mathsf{Enc}_{\mathsf{pp}}(m), \widehat{c}) \leq 2p$ and that $\mathsf{Dec}_{\mathsf{pp}}(\widehat{c}) \notin \{m, \perp\}$. This concludes the proof, since this means that *Eq.* (4.1) implies *Eq.* (4.2). □

4.2.2 Proof of Main Theorems for Seeded Codes: Proof of Theorems 4.1 and 4.2.

Proof of Theorem 4.1. We instantiate our code in Fig. 1 with the following ingredients:

- 2-MICIH Π_{Hash} which is correlation intractable for a sparse linearly shifted relation family $\mathcal{R}$ per the assumption.
- By Theorem 4.5, there exists Π_{code} which is 1-column-robust code with rate $R' = 1$.

Recall by definition since $\mathcal{R}$ is sparse, there exists $\epsilon > 0$ and $q \in \mathrm{poly}(\lambda)$ for which it is $v^{1-\epsilon} \cdot q(\lambda)$-sparse. Thus we can apply Lemma 4.7 on constant $\alpha = 1, \beta = 1 - \epsilon$ and the same q (since $\epsilon > 0 \Leftrightarrow \beta = 1 - \epsilon < 1 = \alpha$) to get a seeded code which is:

- perfectly reliable,
- has rate $R(k, \lambda) = 1 - \frac{v(\lambda)}{k}$,
- for any constant in $p \in [0, \frac{1}{2})$, p-detects.

Furthermore, if Π_{Hash} is public coin, our seeded coin is public coin as well. □

Proof of Theorem 4.2. We instantiate our code in Fig. 1 with the following ingredients:

- 2-MICIH Π_{Hash} which is correlation intractable for a sparse linearly shifted relation family $\mathcal{R}$ per the assumption.
- By Theorem 4.6, for any constant $p_c \in [0, \frac{1}{4})$ and $\epsilon' > 0$, there exists Π_{code} which is $1 - \epsilon'$-column-robust code with rate $R' = R_{BZ}(p_c)$ *and is* (p_c, L)*-list decodable for some* $L \in \mathrm{poly}(k)$.

Recall by definition since $\mathcal{R}$ is sparse, there exists $\epsilon > 0$ and $q \in \mathrm{poly}(\lambda)$ for which it is $v^{1-\epsilon} \cdot q(\lambda)$-sparse. Taking $\epsilon' = \epsilon/2$ in our decomposable code, we can apply Lemma 4.7 on constant $\alpha = 1 - \epsilon/2, \beta = 1 - \epsilon$ and the same q to get a seeded code which is:

- perfectly reliable,
- has rate $R(k, \lambda) = R_{BZ}(p_c) - \frac{v(\lambda)}{k}$,
- is (L, p_c)-list decodable,
- for any constant in $p \in [0, \frac{1}{2})$, p-detects.

Furthermore, if Π_{Hash} is public coin, our seeded coin is public coin as well. Applying Lemma 4.14, and taking p-detection for $p = 2 \cdot p_c$, we also get that our code p_c-corrects. □

5 Multi-Input Correlation Intractability from Standard Assumptions

In this section, we present a generic construction of multi-input correlation intractability for shifted output relations from trapdoor hash functions. In Section 5.2, we formally define trapdoor hashing, and show that the definition implies additional properties that will be important for our construction. In Section 5.2, we prove that trapdoor hashing for a function family $\mathcal{F}$, implies multi-input correlation intractability for shifted sparse relations in this family.

5.1 Trapdoor Hashing

We recall the notion of (rate-1) trapdoor hashing as introduced in [DGI+19a]. Our definition follows the modifications of [BKM20], which introduce *enhanced* correctness and does not require input privacy.

Definition 5.1 (Rate-1 Trapdoor Hash Functions)**.** A rate-1 trapdoor hash for predicate family $\mathcal{F} = \mathcal{F}_\lambda$ consists of a tuple of efficient algorithms $\Pi_{\mathsf{TDH}} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{H}, \mathsf{E}, \mathsf{D})$ with the following syntax:

- $\mathsf{Setup}(1^\lambda, 1^k) \to \mathsf{hk}$: On input a security parameter λ, an input length n, the setup algorithm samples a hash key hk.
- $\mathsf{Gen}(\mathsf{hk}, f) \to \mathsf{ek}, \mathsf{td}$: On input hash key hk and predicate function $f : \{0,1\}^k \to \{0,1\} \in \mathcal{F}$, the setup algorithm samples an evaluation key ek_f and trapdoor td.
- $\mathsf{H}(\mathsf{hk}, x) \to \sigma$: On hash key hk and input $x \in \{0,1\}^k$, the hash algorithm *deterministically* outputs a short hash $\sigma \in \{0,1\}^\ell$.
- $\mathsf{E}(\mathsf{ek}, x) \to y$: On input evaluation key ek and input $x \in \{0,1\}^k$, the encode algorithm *deterministically* outputs a bit $y \in \{0,1\}$.
- $\mathsf{D}(\mathsf{td}, \sigma) \to y'$: On input trapdoor td and hash $\sigma \in \{0,1\}^\ell$, the decode algorithm *deterministically* outputs a bit $y' \in \{0,1\}$.

In addition, Π_{TDH} should satisfy the following properties:

- **Enhanced $(1-\tau)$-Correctness:** For all $\lambda \in \mathbb{N}$, $k \in \mathsf{poly}(\lambda)$ and $f : \{0,1\}^k \to \{0,1\} \in \mathcal{F}$, any hk in the support of $\mathsf{Setup}(1^\lambda, 1^k)$, and $\sigma \in \{0,1\}^\ell$:

$$\Pr_{(\mathsf{ek}_f, \mathsf{td}) \leftarrow \mathsf{Gen}(\mathsf{hk}, f)} \left[\forall x \in \{0,1\}^k : \mathsf{H}(\mathsf{hk}, x) = \sigma \Rightarrow \mathsf{E}(\mathsf{ek}_f, x) + \mathsf{D}(\mathsf{td}, \sigma) = f(x) \right] = 1 - \tau.$$

- **Function Privacy:** For a bit $b \in \{0,1\}$ and a security parameter λ, the function privacy game between an adversary $\mathcal{A}$ and a challenger $\mathsf{ExpFP}^{(b)}(\mathcal{A}, \lambda)$ is as follows:
  1. On input security parameter λ, the adversary outputs an input length 1^n and a functions $f_0, f_1 \in \mathcal{F}$.
  2. The challenger samples hash key $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$ and evaluation key $\mathsf{ek}_{f_b}, \mathsf{td}_{f_b} \leftarrow \mathsf{Gen}(\mathsf{hk}, f_b)$. The challenger then sends $\mathsf{hk}, \mathsf{ek}_{f_b}$ to $\mathcal{A}$.
  3. The adversary outputs a guess $b' \in \{0,1\}$, which is the output of the experiment.

  We say that Π_{TDH} satisfies function privacy if for all polynomial time adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\left| \Pr[\mathsf{ExpFP}^{(0)}(\mathcal{A}, \lambda) = 1] - \Pr[\mathsf{ExpFP}^{(1)}(\mathcal{A}, \lambda) = 1] \right| = \mathsf{negl}(\lambda)$$

- **Compactness:** The size of the image of H is bound by 2^ℓ for some fixed polynomial $\ell = \ell(\lambda)$ (and independent of n).

Trapdoor Hashing for Multi-bit Functions. For the purpose of our construction, it will often be convenient to specify a trapdoor hash over *multi-bit* functions $\mathbf{f} = (f_1, \ldots f_v) \in \{0,1\}^k \to \{0,1\}^v$. In these cases, we will define $\mathsf{Gen}, \mathsf{E}, \mathsf{D}$ to refer to the bitwise output of $\mathbf{f}$ over independently generated evaluation keys as follows:

- $\mathsf{Gen}(\mathsf{hk}, \mathbf{f}) \to \mathsf{ek}_\mathbf{f}, \mathsf{td}_\mathbf{f}$: For each $i \in [m]$, samples $\mathsf{ek}_i, \mathsf{td}_i \leftarrow \mathsf{Gen}(\mathsf{hk}, f_i)$. Output $\mathsf{ek}_\mathbf{f} = (\mathsf{ek}_1, \ldots, \mathsf{ek}_v), \mathsf{td}_\mathbf{f} = (\mathsf{td}_1, \ldots, \mathsf{td}_v)$.
- $\mathsf{E}(\mathsf{ek}_\mathbf{f}, x) \to y$: Taking $\mathsf{ek}_\mathbf{f} = (\mathsf{ek}_1, \ldots, \mathsf{ek}_v)$, for each $i \in [v]$, compute $y_i \leftarrow \mathsf{E}(\mathsf{ek}_i, x)$. Output $y = y_1, \ldots, y_v$.

- $\mathsf{D}(\mathsf{td}_{\mathbf{f}}, \sigma) \to y'$: Taking $\mathsf{td}_{\mathbf{f}} = (\mathsf{td}_1, \ldots, \mathsf{td}_v)$, for each $i \in [v]$, compute $y'_i \leftarrow \mathsf{D}(\mathsf{td}_i, \sigma)$. Output $y' = y'_1, \ldots, y'_v$.

It is easy to see that function privacy still holds over multi-bit $\mathbf{f}$ by a straightforward hybrid argument.

We will make use of the following theorems and result on trapdoor hashing.

Theorem 5.2 ([DGI+19a, BKM20]). *For any constant $c \in \mathbb{N}$ and arbitrarily small $\tau = \tau(\lambda) \in \frac{1}{\mathsf{poly}(\lambda)}$, there exists rate-1 trapdoor hashes for the predicate family of degree c-polynomials over $\mathbb{F}_2$ satisfying enhanced $(1-\tau)$ correctness, function privacy, and compactness assuming DDH/DCR/LWE/QR.*

Theorem 5.3 ([AMR25, BJSS25]). *There exists rate-1 trapdoor hashes for the predicate family of bounded depth circuits satisfying enhanced $(1 - \mathsf{negl})$ correctness, function privacy, and compactness assuming LWE.*

Theorem 5.4 ([AMR25, BJSS25]). *There exists rate-1 trapdoor hashes for the predicate family of poly-size circuits satisfying enhanced $(1 - \mathsf{negl})$ correctness, function privacy, and compactness assuming LWE and a circular security assumption (refer to [HLL23] for the exact assumption).*

Remark 5.5. The known constructions of trapdoor hashing for circuit predicates [AMR25, BJSS25] go through mildly different abstractions and do not show the enhanced correctness property we need. In Appendix B, we present a direct construction of trapdoor hashing for circuits similar to [AMR25] which fulfills the properties we use.

5.1.1 Generic Properties of Trapdoor Hash Functions

Our constructions will also require Π_{TDH} to satisfy a couple of additional properties that are not explicitly stated in the definition of trapdoor hash functions. However, we show these are all either implied by or can be generically constructed from existing constructions of trapdoor hashes.

The Hashing Algorithm H is Collision Resistance. We now show that if $\Pi_{\mathsf{TDH}} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{H}, \mathsf{E}, \mathsf{D})$, then H is collision resistance function. More formally, we define collision resistance as correlation intractability against the relation family $\mathcal{R}$ consisting of the single relation $R(x, x', y, y') \Leftrightarrow (x \neq x') \wedge (y = y')$.

Lemma 5.6 (Collision Resistance). *Let Π_{TDH} be a trapdoor hash for a predicate family $\mathcal{F}$ which includes linear functions[10] and $(1-\tau)$-correctness where $\tau \leq \frac{1}{2} - \frac{1}{\mathsf{poly}(\lambda)}$ for any polynomial. Then algorithm H on $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$ is a collision resistant hash function.*

Proof of Lemma 5.6. Suppose $\mathcal{A}$ is an adversary that wins the collision-resistance game with advantage ϵ. Then, consider the following reduction $\mathcal{B}$ against function privacy of Π_{TDH}.

1. Reduction $\mathcal{B}$ samples $i \in [n]$, and submits input length 1^k, f_0 as the constant 0 function and f_1 as the ith standard basis vector e_i to the function privacy challenger.
2. The challenger responds with $(\mathsf{hk}, \mathsf{ek})$. $\mathcal{B}$ forwards hk to $\mathcal{A}$, who responds with x, x'.
3. Reduction $\mathcal{B}$ outputs 1 iff:

$$\mathsf{H}(\mathsf{hk}, x) = \mathsf{H}(\mathsf{hk}, x') \wedge x'_i \neq x'_i \wedge \mathsf{E}(\mathsf{ek}, x) \neq \mathsf{E}(\mathsf{ek}, x')$$

First, note that $\mathcal{B}$ accurately simulates the view of $\mathcal{A}$ in the collision resistance game (by simply forwarding hk). To see that $\mathcal{B}$ has noticeable advantage in the function privacy game, consider the following events:

1. The challenger samples $i \in [n] : x_i \neq x'_i$.
2. The key ek_f sampled by the challengers satisfies enhanced correctness.

[10] more specifically $\mathcal{F}$ only needs to support index functions

If event 2 is true, observe that since $\mathsf{H}(\mathsf{hk}, x) = \mathsf{H}(\mathsf{hk}, x') = \sigma$, we can simplify our correctness equation as $\mathsf{E}(\mathsf{ek}_f, x) - f(x) = \mathsf{E}(\mathsf{ek}_f, x') - f(x')$. Furthermore, since $f_1(x) \neq f_1(x')$ when $x_i \neq x'_i$ in event 1, we can conclude when $b = 0$, $\mathsf{E}(\mathsf{ek}, x) = \mathsf{E}(\mathsf{ek}, x')$ and conversely, $\mathsf{E}(\mathsf{ek}, x) \neq \mathsf{E}(\mathsf{ek}, x')$ when $b = 1$. Thus, conditional on event 1, we can upper bound $\mathbf{Pr}[\mathsf{E}(\mathsf{ek}, x) \neq \mathsf{E}(\mathsf{ek}, x') \mid b = 0] \leq \tau$, which gives us a lower bound on the advantage of $\mathcal{B}$ as $1 - 2\tau$, which is noticeable by assumption.

Since event 1 is *independent* of the view of the adversary, and occurs with polynomial probability, we can bound the overall advantage of $\mathcal{B}$ with some $\epsilon/\mathsf{poly}(\lambda)$. □

Closeness for All Inputs. We also show that when generalized to multi-bit inputs, enhanced correctness implies a notion of 'closeness', where the bits recovered by trapdoor hashing $\mathsf{E}(\mathsf{ek}_\mathbf{f}, x) \oplus \mathsf{D}(\mathsf{td}_\mathbf{f}, \sigma)$ have bounded hamming distance to the true output of the function $\mathbf{f}(x)$ for all inputs x. This property is somewhat implicitly used in [BKM20], but we formalize it below.

Lemma 5.7 (Semi-perfect ε-Closeness)**.** *Suppose Π_{TDH} satisfies $(1-\tau)$ enhanced correctness for any arbitrarily small $\tau \in \frac{1}{\mathsf{poly}(\lambda)}$. Then for $\lambda \in \mathbb{N}$, $n, v \in \mathsf{poly}(\lambda)$, and $\mathbf{f} : \{0,1\}^k \to \{0,1\}^v \in \mathcal{F}_\lambda$,*

$$\Pr_{\substack{\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k) \\ \forall i \in [v] \ (\mathsf{ek}_\mathbf{f}, \mathsf{td}_\mathbf{f}) \leftarrow \mathsf{Gen}(\mathsf{hk}, \mathbf{f})}} \left[\forall x \in \{0,1\}^k \ \ \Delta(\mathsf{E}(\mathsf{ek}_\mathbf{f}, x), \mathsf{D}(\mathsf{td}_\mathbf{f}, \sigma) + f(x)) \leq \varepsilon \mid \sigma = \mathsf{H}(\mathsf{hk}, x)\right] = 1 - \mathsf{negl}(\lambda)$$

Where $\varepsilon = \varepsilon(\lambda)$ is a fixed polynomial (independent of n, v).

Proof. Fix an arbitrary $\sigma \in \{0,1\}^\ell$. Let X_i be the binary random variable that is 1 iff the ith bit of $\mathbf{f}$ satisfies correctness - i.e. $\mathsf{E}(\mathsf{ek}_i, x) + \mathsf{D}(\mathsf{td}_i, \sigma) \neq f_i(x)$. Since these are independent with expectation τ, we can Chernoff bound

$$\mathbf{Pr}\left[\sum_{i \in [v]} X_i \geq \frac{\varepsilon}{\tau \cdot v} \cdot (\tau \cdot v)\right] \leq \exp\left(-\left(\frac{\varepsilon}{\tau \cdot v} - 1\right)^2 \cdot (\tau \cdot v) / \left(2 + \left(\frac{\varepsilon}{\tau \cdot v} - 1\right)\right)\right)$$

Setting $\tau = v^{-1}$ and $\varepsilon = 2\ell\lambda + 1$, we can bound this probability with $\exp(-\ell \cdot \lambda)$. Now union bounding over all 2^ℓ values of σ, we can union bound the probability any σ is 'incorrect' on $> \varepsilon$ many indices with $2^\ell \cdot \exp(-\ell \cdot \lambda) = \mathsf{negl}(\lambda)$. □

t-wise Independence of the Decoder's Output. We require that $\mathsf{D}(\mathsf{td}, \cdot)$ is a t-wise independent function over the choice of td. We now show that this property can be generically added to any trapdoor hash.

Lemma 5.8 (t-Universality)**.** *Let Π_{TDH} be a trapdoor hash. Then there exists $\Pi'_{\mathsf{TDH}} = (\mathsf{Setup}', \mathsf{Gen}', \mathsf{H}', \mathsf{E}', \mathsf{D}')$ such that for any constant $t \in \mathbb{N}$[11] For all $\lambda, n \in \mathbb{N}$, unique $\{\sigma^{(k)}\}_{k \in [t]} \in \{0,1\}^\ell$, all $y^{(1)}, \ldots, y^{(t)} \in \{0,1\}$, and all $f \in \mathcal{F}$*

$$\Pr_{\substack{\mathsf{hk} \leftarrow \mathsf{Setup}'(1^\lambda, 1^k) \\ (\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}'(\mathsf{hk}, f)}} \left[\forall k \in [t] \ \ \mathsf{D}'(\mathsf{td}, \sigma^{(k)}) = y^{(k)}\right] \leq 2^{-t}$$

Proof. We define our new trapdoor hash as follows:

- $\mathsf{Setup}'(1^\lambda, 1^k)$: Same as $\mathsf{Setup}(1^\lambda, 1^k)$.
- $\mathsf{Gen}'(\mathsf{hk}, f)$: Sample K as the hash key from a t-universal hash family $\{0,1\}^\ell \to \{0,1\}$. Compute $\mathsf{ek}, \mathsf{td} \leftarrow \mathsf{Gen}(\mathsf{hk}, f)$, and output $(\mathsf{ek}' = (\mathsf{hk}, \mathsf{ek}, K), \mathsf{td}' = (\mathsf{td}, K))$.
- $\mathsf{H}'(\mathsf{hk}, x)$: Same as $\mathsf{H}(\mathsf{hk}, x)$.
- $\mathsf{E}'(\mathsf{ek}' = (\mathsf{hk}, \mathsf{ek}, K), x)$: Output $\mathsf{E}(\mathsf{ek}, x) + \mathsf{Hash}(K, \mathsf{H}(\mathsf{hk}, x))$.

[11] Our constructions will support any $t \in \mathsf{poly}(\lambda)$. However, since we only need $t = 2$ our construction of binary codes, we treat t as a constant for simplicity.

- $\mathsf{D}'(\mathsf{td}', \sigma)$: Output $\mathsf{D}(\mathsf{td}, \sigma) + \mathsf{Hash}(K, \sigma)$.

Since our new hash simply shifts all inputs by $\mathsf{Hash}(K, \sigma)$, it is easy to verify that Π'_{TDH} satisfies the same correctness/compactness parameters of Π_{TDH}. Similarly, function privacy of Π'_{TDH} follows directly from function privacy of Π_{TDH}, as $(\mathsf{hk}', \mathsf{ek}')$ is simply a function of hk, ek and an independently sampled K.

Finally, t-universality of Π_{TDH} follows from t-universality of K after fixing randomness of Setup, Gen. □

Corollary 5.9. *As an immediate corollary of the previous Lemma 5.8, we can see that over v-bit output functions $\mathbf{f}$, for all unique $\{\sigma^{(k)}\}_{k\in[t]} \in \{0,1\}^{\ell}$ and all $y^{(1)}, \ldots, y^{(t)} \in \{0,1\}^{v}$,*

$$\Pr_{\substack{\mathsf{hk}\leftarrow\mathsf{Setup}'(1^\lambda,1^k)\\(\mathsf{ek},\mathsf{td})\leftarrow\mathsf{Gen}'(\mathsf{hk},\mathbf{f})}}\left[\forall k \in [t]\ \ \mathsf{D}'(\mathsf{td}, \sigma^{(k)}) = y^{(k)}\right] \leq 2^{-tv}$$

since each of the v output bits are sampled independently.

5.2 Constructing Multi-Input Correlation Intractability from Trapdoor Hashing

Now, we are ready to construct multi-input correlation intractability from trapdoor hashing. While the construction we show in this section is not necessarily public coin, we show in Appendix C how to extend the requirements on trapdoor hashing to yield a public coin CI hash.

Theorem 5.10. *Let t be a constant and suppose Π_{TDH} is a t-universal trapdoor hash for the function family $\mathcal{F}$. Let Π_{Hash} be the hash family defined as*

- $\mathsf{Gen}(1^\lambda, 1^k, 1^v)$: *Sample* $\mathsf{hk} \leftarrow \Pi_{\mathsf{TDH}}.\mathsf{Setup}(1^\lambda, 1^k)$, *and* $(\mathsf{ek_0}, \mathsf{td_0}) \leftarrow \Pi_{\mathsf{TDH}}.\mathsf{Gen}(\mathsf{hk}, (\mathbf{0}^{k\times v}))$. *Output* $(\mathsf{hk}, \mathsf{ek_0})$.
- $\mathsf{Hash}(K = (\mathsf{hk}, \mathsf{ek_0}), x) = \Pi_{\mathsf{TDH}}.\mathsf{E}(\mathsf{ek_0}, x)$.

Then there exists polynomial $s = s(\lambda)$ (independent of k, v) such that Π_{Hash} is t-correlation intractable for s-sparse efficiently checkable $\mathcal{F}$-shifted relation ensembles. In particular, instantiated with existing trapdoor hash constructions, this implies the multi-input CI where $\mathcal{F}$ is:

- *Degree c functions for any constant $c \in \mathbb{N}$ assuming one of DDH/DCR/LWE/QR by Theorem 5.2.*
- *Circuits of bounded depth $d = d(\lambda) \in \mathsf{poly}(\lambda)$ assuming LWE by Theorem 5.3 (where sparsity s also depends linearly on d).*
- *Circuits of size $c = c(\lambda) \in \mathsf{poly}(\lambda)$ assuming LWE and a circular security assumption by Theorem 5.4 (where sparsity s is* independent *of c).*

In particular, this is sufficient to instantiate Lemma 4.7 by simply taking $\beta = 0, q(\lambda) = s(\lambda)$.

We prove the above theorem in the following section Theorem 5.10.

Remark 5.11. We note that the correlation intractability our construction achieves can be mildly generalized beyond the above theorem statement in the following dimensions:

- We can support super-constant (up to polynomial) input arity t at the cost of decreasing the sparsity s of the relation ensemble supported.
- If our trapdoor hash is secure against time $T(\lambda)$ adversaries, we can relax the efficient checkability of $\mathcal{R} = \{(R_{\mathsf{out},\lambda}, f_\lambda)\}$ where $R_{\mathsf{out},\lambda}$ only needs to be checkable in $\ll T(\lambda)$ time.

5.2.1 Proof of Theorem 5.10

Loosely speaking, our proof builds on the ideas of [BKM20], which showed how to construct single-input *CI* functions. Following [BKM20], we rely on the fact that an evaluation key on a predicate f shifts the value of E with respect to D by $f(x)$. This allows us to (undetectably) sample the evaluation key so as to cancel out the shift induced by the relation, thereby reducing a shifted relation on E to an "approximate" output relation on D.

To generalize the result to multi-input relations, we require additional properties, which we describe and prove in Section 5.1.1. The first property is that H is a collision-resistant hash function. This guarantees that when the adversary chooses distinct $x^{(1)}, \ldots, x^{(t)}$, the corresponding values $\sigma^{(1)}, \ldots, \sigma^{(t)}$, where $\sigma^{(i)} = \mathsf{H}(\mathsf{hk}, x^{(i)})$, are also distinct. This enables the use of the second property: $\mathsf{D}(\mathsf{td}, \cdot)$ is a t-wise independent function. Specifically, for any fixed $\sigma^{(1)}, \ldots, \sigma^{(t)}$ and a random td, the values $\mathsf{D}(\mathsf{td}, \sigma^{(1)}), \ldots, \mathsf{D}(\mathsf{td}, \sigma^{(t)})$ are uniformly distributed, which implies that they are unlikely to lie in any fixed sparse relation R. Finally, since the set $\sigma^{(1)}, \ldots, \sigma^{(t)}$ is compact, we can apply a union bound over all possible distinct $\sigma^{(1)}, \ldots, \sigma^{(t)}$ to show that, with overwhelming probability, there exist no values for which $\mathsf{D}(\mathsf{td}, \sigma^{(1)}), \ldots, \mathsf{D}(\mathsf{td}, \sigma^{(t)})$ all belong to R. A formal proof and analysis follows below.

Proof of Theorem 5.10. We will show that our result holds for any $s \in o(\log^2 \lambda \cdot (\ell + \varepsilon))$ (where ℓ, ε are the compactness and closeness parameters of Π_{TDH} respectively). Since ℓ, ε are fixed polynomials independent of n and v, s is as well.

We define our series of experiments as follows:

- $\mathsf{Exp}^{(\mathcal{R})}_{\mathsf{real}}$: This is the real t-input correlation intractability game for a relation ensemble $\mathcal{R} = \{(R_{\mathsf{out},\lambda}, f_\lambda)\}_{\lambda \in \mathbb{N}}$ with sparsity defined as above. We will write R, f to refer to $R_{\mathsf{out},\lambda}, f_\lambda$ on security parameter of the experiment respectively.

  1. The challenger samples $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$ and $\mathsf{ek} \leftarrow \mathsf{Gen}(\mathsf{hk}, \mathbf{0}^{n \times v})$, which it sends to the adversary.
  2. The adversary sends inputs $x^{(1)}, \ldots, x^{(t)}$ to the challenger.
  3. The challenger computes $y^{(i)} = \mathsf{E}(\mathsf{ek}, x^{(i)})$ for all $i \in [t]$ and sets
  $$R(y^{(1)} - f(x^{(1)}), \ldots, y^{(t)} - f(x^{(t)}))$$
  as the output of the experiment.

- $\mathsf{Exp}^{(\mathcal{R})}_{1}$: In this experiment, the challenger samples ek with respect to predicate f.

  1. The challenger samples $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$ and $\mathsf{ek} \leftarrow \mathsf{Gen}(\mathsf{hk}, f)$, which it sends to the adversary.

- $\mathsf{Exp}^{(\mathcal{R})}_{2}$: In this experiment, the output of the experiment is 0 when there exists $k \neq k' \in [t]$ such that $\mathsf{H}(\mathsf{hk}, x^{(k)}) = \mathsf{H}(\mathsf{hk}, x^{(k')})$.

  3. The challenger computes $y^{(i)} = \mathsf{E}(\mathsf{ek}, x^{(i)})$ for all $i \in [t]$ and sets
  $$R(y^{(1)} - f(x^{(1)}), \ldots, y^{(t)} - f(x^{(t)})) \wedge \forall i \neq j \in [t] \;\; \mathsf{H}(\mathsf{hk}, x^{(i)}) \neq \mathsf{H}(\mathsf{hk}, x^{(j)})$$
  as the output of the experiment.

Lemma 5.12. *Assuming Π_{TDH} is function private, then for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that*

$$\left| \mathbf{Pr}[\mathsf{Exp}^{(\mathcal{R})}_{\mathsf{real}}(\mathcal{A}, \lambda) = 1] - \mathbf{Pr}[\mathsf{Exp}^{(\mathcal{R})}_{1}(\mathcal{A}, \lambda) = 1] \right| = \mathsf{negl}(\lambda).$$

Proof of Lemma 5.12. Suppose for contradiction adversary $\mathcal{A}$ has nonnegligible distinguishing advantage between $\mathsf{Exp}_{\mathsf{real}}$ and Exp_1. Then the following is a reduction $\mathcal{B}$ against function privacy of Π_{TDH}.

0. Reduction $\mathcal{B}$ receives security parameter λ and sends input length 1^k, and challenge functions $f_0 = \mathbf{0}^{k \times v}$ and $f_1 = f$ to the function privacy challenger.

1. $\mathcal{B}$ receives a hash key hk and evaluation key ek from the challenger. $\mathcal{B}$ begins running adversary $\mathcal{A}$, forwarding them hk, ek.

2. Adversary $\mathcal{A}$ responds with $x^{(1)}, \ldots, x^{(t)}$ to the challenger.

3. Reduction $\mathcal{B}$ computes $y_i = \mathsf{E}(\mathsf{ek}, x^{(i)})$ for $i \in [t]$, and sets $b' = R(y^{(1)} - f(x^{(1)}), \ldots, y^{(t)} - f(x^{(t)}))$, which it sends to the function privacy challenger.

Observe that when $b = 0$, this ek is an evaluation key for $\mathbf{0}^{k \times v}$, exactly as in $\mathsf{Exp}_{\mathsf{real}}$ □

Lemma 5.13. *Assuming Π_{TDH} is collision resistant, then for all efficient adversaries $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\left|\mathbf{Pr}[\mathsf{Exp}_1^{(\mathcal{R})}(\mathcal{A}, \lambda) = 1] - \mathbf{Pr}[\mathsf{Exp}_2^{(\mathcal{R})}(\mathcal{A}, \lambda) = 1]\right| = \mathsf{negl}(\lambda).$$

Proof of Lemma 5.13. Observe that the outcome of these experiments differ only when the additional condition of Exp_2 does *not* apply - i.e. there exists some i, j such that $\mathsf{H}(\mathsf{hk}, x^{(i)}) = \mathsf{H}(\mathsf{hk}, x^{(j)})$. Furthermore, recall that since we are considering distinct input relations, this means that $x^{(i)} \neq x^{(j)}$. As such, an attacker $\mathcal{A}$ with noticeable advantage between Exp_1 and Exp_2 must have found two such x's which breaks collision resistance of H as well. We formalize this as the following reduction $\mathcal{B}$:

1. Reduction $\mathcal{B}$ receives hash key hk from the collision resistance challenger, computes $\mathsf{ek} \leftarrow \mathsf{Gen}(\mathsf{hk}, f)$, and sends $(\mathsf{hk}, \mathsf{ek})$ to the CI adversary $\mathcal{A}$.

2. $\mathcal{A}$ returns inputs $x^{(1)} \ldots x^{(t)}$ to $\mathcal{B}$. $\mathcal{B}$ checks if any pair $x^{(i)} \neq x^{(j)}$ and $\mathsf{H}(\mathsf{hk}, x^{(i)}) = \mathsf{H}(\mathsf{hk}, x^{(j)})$ and forwards it to the challenger if so.

□

Lemma 5.14. *Assuming Π_{TDH} is compact, t-universal, and semi-perfect ε-close, then for all efficient adversaries $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\mathbf{Pr}[\mathsf{Exp}_2^{(\mathcal{R})}(\mathcal{A}, \lambda) = 1] = \mathsf{negl}(\lambda).$$

Proof of Lemma 5.14. Recall our relation family $\mathcal{R}$ only contains output relations R with 2^{-s}-sparsity. We will argue that for such sufficiently sparse R, it is statistically unlikely *any* inputs satisfy the relation in this experiment. Define relation $R'(z^{(1)}, \ldots z^{(t)}) = \exists z'^{(1)}, \ldots z'^{(t)} : \forall i \in [t] \;\; \Delta(z'^{(i)}, z^{(i)}) \leq \tau \wedge R(z'^{(1)}, \ldots z'^{(t)}) = 1$.

Claim 5.15. *If R is 2^{-s} sparse, then the relation R' is $2^{-s+O(t \cdot \varepsilon \log \lambda)}$ sparse.*

Proof of Claim 5.15. This follows from a simple counting argument. Consider the size ε hamming balls around each tuple which satisfies R. This ball has size $\leq v^{t \cdot \varepsilon}$. Since $v \in \mathsf{poly}(\lambda)$, this can be bound by some $2^{O(t \cdot \varepsilon \log \lambda)}$. Thus this new relation is at most $2^{-s+O(t \cdot \varepsilon \log \lambda)}$ sparse. □

Claim 5.16. *With all but negligible probability, there does not exist distinct $\sigma^{(1)} \ldots \sigma^{(t)} \in \{0,1\}^{\ell}$ such that $R'(\mathsf{D}(\mathsf{td}, \sigma^{(1)}),$ $\ldots \mathsf{D}(\mathsf{td}, \sigma^{(t)})) = 1$.*

Proof of Claim 5.16. For any tuple $\sigma^{(1)}, \ldots, \sigma^{(t)}$, by t-universality of Π_{TDH} and Claim 5.15, the probability

$$\Pr_{\mathsf{td}}[R'(\mathsf{D}(\mathsf{td}, \sigma^{(1)}), \ldots, \mathsf{D}(\mathsf{td}, \sigma^{(t)})) = 1] \leq 2^{t \cdot v - s + O(t \cdot \varepsilon \log \lambda) - t \cdot v}.$$

Union bounding over all the space of all $\leq 2^{t\ell}$ pairs, we get the probability any such $\sigma^{(1)}, \ldots \sigma^{(t)}$ exist as $2^{-s+O(t \cdot \varepsilon \log \lambda)+2\ell}$ $\cdot (1 + \mathsf{negl}(\lambda))$. Recall that since $s \in o(t \cdot \log^2 \lambda \cdot (\ell + \varepsilon))$, $2^s \ll 2^{O(t \cdot \varepsilon \log \lambda)+\ell}$ so this quantity is negligible. □

Finally, to tie everything together, we can observe that by ε-closeness of Π_{TDH}, the value of $\mathsf{D}(\mathsf{td}, \mathsf{H}(\mathsf{hk}, x))$ is hamming distance ε from $\mathsf{E}(\mathsf{ek}_f, x) - f(x) = y - f(x)$. Thus for any $x^{(1)}, \ldots x^{(t)}$ for which $\mathsf{E}(\mathsf{ek}_f, x^{(1)}) - f(x^{(1)}), \ldots, \mathsf{E}(\mathsf{ek}_f, x^{(t)}) - f(x^{(t)})$ satisfy R, $\mathsf{D}(\mathsf{td}, \mathsf{H}(\mathsf{hk}, x^{(1)})), \cdots, \mathsf{D}(\mathsf{td}, \mathsf{H}(\mathsf{hk}, x^{(t)}))$ must also satisfy R'. Since by Claim 5.16, we know that with overwhelming probability no such values exist, this concludes our proof. □

Taking together Lemmas 5.12 to 5.14 gives us our theorem statement. □

References

[AMR25] 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*, pages 1875–1886, 2025.

[BCI+10] Eric Brier, Jean-Sébastien Coron, Thomas Icart, David Madore, Hugues Randriam, and Mehdi Tibouchi. Efficient indifferentiable hashing into ordinary elliptic curves. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 237–254. Springer, 2010.

[BGGZ21] Jeremiah Blocki, Venkata Gandikota, Elena Grigorescu, and Samson Zhou. Relaxed locally correctable codes in computationally bounded channels. *IEEE Transactions on Information Theory*, 67(7):4338–4360, 2021.

[BIP+18] Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J Wu. Exploring crypto dark matter: New simple prf candidates and their applications. In *Theory of Cryptography Conference*, pages 699–729. Springer, 2018.

[BJSS25] Elette Boyle, Abhishek Jain, Sacha Servan-Schreiber, and Akshayaram Srinivasan. Simultaneous-message and succinct secure computation. In *EUROCRYPT*, 2025.

[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, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, volume 12172 of *Lecture Notes in Computer Science*, pages 738–767. Springer, 2020.

[BZ76] EL Blokh and VV Zyablov. Generalized concatenated codes. *Svyaz', Moscow*, 1976.

[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 *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1082–1090, 2019.

[CCR16] Ran Canetti, Yilei Chen, and Leonid Reyzin. On the correlation intractability of obfuscated pseudorandom functions. In *Proceedings, Part I, of the 13th International Conference on Theory of Cryptography - Volume 9562*, TCC 2016-A, page 389–415, Berlin, Heidelberg, 2016. Springer-Verlag.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D Rothblum. Fiat-shamir and correlation intractability from strong kdm-secure encryption. In *Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part I 37*, pages 91–122. Springer, 2018.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM (JACM)*, 51(4):557–594, 2004.

[DGI+19a] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor hash functions and their applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 3–32. Springer, 2019.

[DGI+19b] Nico Döttling, Sanjam Garg, Yuval Ishai, Giulio Malavolta, Tamer Mour, and Rafail Ostrovsky. Trapdoor hash functions and their applications. In *Annual International Cryptology Conference*, pages 3–32. Springer, 2019.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theory*, 22(6):644–654, 1976.

[GHY20] Ofer Grossman, Justin Holmgren, and Eylon Yogev. Transparent error correcting in a computationally bounded world. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part III*, volume 12552 of *Lecture Notes in Computer Science*, pages 530–549. Springer, 2020.

[GR08] Venkatesan Guruswami and Atri Rudra. Better binary list decodable codes via multilevel concatenation. *IEEE Transactions on Information Theory*, 55(1):19–26, 2008.

[GS16] V. Guruswami and A. Smith. Optimal rate code constructions for computationally simple channels. *Journal of the ACM (JACM)*, 63(4):35, 2016.

[Ham50] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950.

[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 IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 850–858. IEEE Computer Society, 2018.

[HLL23] Yao-Ching Hsieh, Huijia Lin, and Ji Luo. Attribute-based encryption for circuits of unbounded depth from lattices. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 415–434. IEEE, 2023.

[HOSW11] Brett Hemenway, Rafail Ostrovsky, Martin J. Strauss, and Mary Wootters. Public key locally decodable codes with short keys. In Leslie Ann Goldberg, Klaus Jansen, R. Ravi, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 14th International Workshop, APPROX 2011, and 15th International Workshop, RANDOM 2011, Princeton, NJ, USA, August 17-19, 2011. Proceedings*, volume 6845 of *Lecture Notes in Computer Science*, pages 605–615. Springer, 2011.

[Ica09] Thomas Icart. How to hash into elliptic curves. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 303–316. Springer, 2009.

[KRR17] Yael Tauman Kalai, Guy N Rothblum, and Ron D Rothblum. From obfuscation to the security of fiat-shamir for proofs. In *Advances in Cryptology–CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20–24, 2017, Proceedings, Part II 37*, pages 224–251. Springer, 2017.

[Lip94] R. J. Lipton. A new approach to information theory. In *11th Annual Symposium on Theoretical Aspects of Computer Science*, pages 699–708, 1994.

[LV20] Alex Lombardi and Vinod Vaikuntanathan. Fiat-shamir for repeated squaring with applications to ppad-hardness and vdfs. In *Annual International Cryptology Conference*, pages 632–651. Springer, 2020.

[LV22] Alex Lombardi and Vinod Vaikuntanathan. Correlation-intractable hash functions via shift-hiding. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 102:1–102:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[LW22] George Lu and Brent Waters. How to sample a discrete gaussian (and more) from a random oracle. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography - 20th International Conference, TCC 2022, Chicago, IL, USA, November 7-10, 2022, Proceedings, Part II*, volume 13748 of *Lecture Notes in Computer Science*, pages 653–682. Springer, 2022.

[MPSW10] Silvio Micali, Chris Peikert, Madhu Sudan, and David A. Wilson. Optimal error correction for computationally bounded noise. *IEEE Trans. Information Theory*, 56(11):5673–5680, 2010.

[MRH04] Ueli Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *Theory of Cryptography*, pages 21–39, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[OPS07] Rafail Ostrovsky, Omkant Pandey, and Amit Sahai. Private locally decodable codes. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *Automata, Languages and Programming, 34th International Colloquium, ICALP 2007, Wroclaw, Poland, July 9-13, 2007, Proceedings*, volume 4596 of *Lecture Notes in Computer Science*, pages 387–398. Springer, 2007.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for np from (plain) learning with errors. In *Annual International Cryptology Conference*, pages 89–114. Springer, 2019.

[QWW18] Willy Quach, Hoeteck Wee, and Daniel Wichs. Laconic function evaluation and applications. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 859–870, 2018.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.

[RSS11] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In *Proceedings of the 30th Annual International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT'11, page 487–506, Berlin, Heidelberg, 2011. Springer-Verlag.

[SS21a] R. Shaltiel and J. Silbak. Explicit list-decodable codes with optimal rate for computationally bounded channels. *Comput. Complex.*, 30(1):3, 2021.

[SS21b] R. Shaltiel and J. Silbak. Explicit uniquely decodable codes for space bounded channels that achieve list-decoding capacity. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1516–1526, 2021.

[SS22] R. Shaltiel and J. Silbak. Error correcting codes that achieve BSC capacity against channels that are poly-size circuits. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 13–23, 2022.

[SS24] R. Shaltiel and J. Silbak. Explicit codes for poly-size circuits and functions that are hard to sample on low entropy distributions. *Electronic Colloquium on Computational Complexity (ECCC)*, 2024.

[SW25a] Jad Silbak and Daniel Wichs. Binary codes for error detection and correction in a computationally bounded world. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 186–211. Springer, 2025.

[SW25b] Jad Silbak and Daniel Wichs. Detecting and correcting computationally bounded errors: A simple construction under minimal assumptions. In *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, pages 88–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2025.

A Proofs for Decomposable Binary Codes

A.1 Proof of Theorem 4.5

We will make use of the following fact regarding random matrices.

Fact A.1. Let $\mathbf{V} \in \mathbb{F}_2^{v \times v}$ be a random matrix, it holds that $\mathbf{Pr}\left[\mathsf{rank}(\mathbf{V}) = v\right] \geq \gamma$ for some constant $0 < \gamma$.

Proof of Fact A.1. Let W_i be a random variable that takes 1 iff the first i rows of $\mathbf{V}$ are linearly independent. It holds that

$$\mathbf{Pr}\left[\mathsf{rank}(\mathbf{V}) = v\right] = \mathbf{Pr}\left[W_v = 1\right] = \prod_{i=1}^{i=v} \mathbf{Pr}\left[W_i = 1 | W_{i-1} = 1\right] = \prod_{i=1}^{i=v} (1 - 1/2^{v-i+1}) \geq \gamma \tag{A.1}$$

where $0 < \gamma \approx 0.2885$ is some universal constant. □

Proof of Theorem 4.5. Let $k, v \in \mathbb{N}$ be parameters, and let $n \coloneqq k + v$. We define $\mathsf{GenM}(1^k, 1^v)$ as follows:

- Sample $\mathbf{B}' \leftarrow \mathbb{F}_2^{v \times n}$ uniformly at random. If $\mathbf{B}'$ has full-rank set $\mathbf{B} = \mathbf{B}'$, otherwise, set $\mathbf{B}$ to be an arbitrary full rank matrix of the same size.
- Set $\mathbf{A} \in \mathbb{F}_2^{k \times n}$ to be an arbitrary matrix, such that $\mathbf{G} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$ has full-rank, and output $\mathbf{G}$.

First note that by construction, for every $k, v \in \mathbb{N}$, it holds that $\mathbf{G} \leftarrow \mathsf{GenM}(1^k, 1^v)$ has full-rank, rate $R = 1$, and that $\mathbf{G} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$, for $\mathbf{A} \in \mathbb{F}_2^{k \times n}$ and $\mathbf{B} \in \mathbb{F}_2^{v \times n}$.

We now argue the column-robustness property. Recall that by assumption $v < k$, and let $\mathbf{B}' \in \mathbb{F}_2^{v \times n}$ the random matrix sampled uniformly at random in the construction above, and let E be the event that takes 1 iff $\mathbf{B}'$ has full-rank. Note that by construction if $v < k$ then,

$$\mathbf{Pr}\left[E = 1\right] > 1 - v \cdot \frac{1}{2^v},$$

and that also by construction $(\mathbf{B} | \{E = 1\}) \equiv (\mathbf{B}' | \{E = 1\})$. We will show that when $E = 1$, the matrix $(\mathbf{B} | \{E = 1\})$ is (q, α)-well formed for $\alpha = 1$ and $q = \gamma$ where gamma is some constant (and note that this suffices for showing column-robustness since by the above $\mathbf{Pr}\left[E = 1\right] \geq 1 - \mathsf{negl}(v)$).

Let T be a subset in $[n]$ such that $|T| = v$, observe that by linearity, if $\mathsf{rank}([\mathbf{B}']_T) = v$ then it holds that $E = 1$. It thus follows that, for any subset T in $[n]$ such that $|T| = v$

$$\begin{aligned} \mathbf{Pr}[\mathsf{rank}([\mathbf{B}]_T) = v | E = 1] &\geq \mathbf{Pr}[\mathsf{rank}([\mathbf{B}]_T) = v \wedge E = 1] \\ &= \mathbf{Pr}[\mathsf{rank}([\mathbf{B}]_T) = v] \geq \gamma \end{aligned} \tag{A.2}$$

for some constant $0 < \gamma$, where the last inequality follows by Fact A.1 since $[\mathbf{B}]_T$ is uniform random matrix over $\mathbb{F}_2^{v \times v}$. This concludes the proof for column-robustness, since Eq. (A.2) holds for any subset T in $[n]$ such that $|T| = v$, the same bound holds also when T is a random subset. □

A.2 Proof of Theorem 4.6

In this section we prove Theorem 4.6. In Appendix A.2.1 we start by defining dual-distance, concatenated codes and show how they relate to column-robustness, and in Appendix A.2.2 we make use of these definitions and observations to prove Theorem 4.6.

A.2.1 Dual-Distance of a Code Concatenation

Definition A.2 (Linear codes and dual codes)**.** Let $\mathbb{F}_q$ denote the field with q element. An $[n,k]_q$ ***linear code*** is a linear subspace of $C \subseteq \mathbb{F}_q^n$ of dimension k. We say that C has distance d, if the Hamming weight of every nonzero vector in C is at least d. Such codes are called $[n,k,d]_q$ codes. A linear map $\mathsf{Enc} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ is an ***encoding function*** for C, if $\mathsf{Enc}(\mathbb{F}_q^k) = C$. We say that $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ is the generating matrix of C, if for ever $m \in \mathbb{F}_q^k$: $\mathsf{Enc}(m) = m \cdot \mathbf{G}$. For a code C, we use $C^\perp$ to denote the dual vector space. We say that C has ***dual distance*** d if $C^\perp$ has distance d.

We will use the following claim which will help us reason about the linear independence of the columns of the generating matrix of a code.

Claim A.3. *Let* $\mathsf{Enc} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ *be an encoding function for a linear* $[n,k]_q$*-code* C *with a generating matrix* $\mathbf{G} \in \mathbb{F}_q^{k \times n}$*. Applying* Enc *on a uniformly chosen message* $m \leftarrow \mathbb{F}_q^k$ *(i.e.,* $\mathsf{Enc}(m) = m \cdot \mathbf{G}$*) yields a distribution* $(Z_1, \ldots, Z_n)$ *over* $\mathbb{F}_q^n$*. For a subset* $T \subseteq [n]$*, it holds that random variables* $(Z_i)_{i \in T}$ *are uniform and independent iff* $\mathsf{rank}([\mathbf{G}]_T) = |T|$*.*

Proof. Follows directly by linearity. □

t-Wise Independent. We say that Random variables $R_1, \ldots, R_n$ are t-wise independent if for every $i_1, \ldots, i_t \in [n]$, $R_{i_1}, \ldots, R_{i_t}$ are uniform and independent.

By the above claim, it is easy to derive the the following standard fact that encoding functions for codes with dual distance $t+1$ yield a t-wise independent distributions.

Lemma A.4 (t-wise independence from linear codes with $t+1$ dual-distance)**.** *Let* $\mathsf{Enc} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ *be an encoding function for a linear* $[n,k]_q$*-code* C *with dual distance* $t+1$*. Applying* Enc *on a uniformly chosen message* $m \leftarrow \mathbb{F}_q^k$ *yields a distribution* $(Z_1, \ldots, Z_n)$ *over* $\mathbb{F}_q^n$ *that is* t*-wise independent, and every* Z_i *is uniformly distributed over* $\mathbb{F}_q$*.*

Proof. Applying Enc on some $v \in \mathbb{F}^k$, can be seen as multiplying v by a generator matrix of C (which is a transposed parity check matrix of $C^\perp$). As $C^\perp$ has distance $t+1$, every t columns of the generator matrix of C are linearly independent, and the lemma follows. □

We now define code Concatenation. Looking ahead, the structure of the code will help us argue the column-robustness that we need.

Definition A.5 (Concatenated code)**.** Given functions:

- $\mathsf{Enc}_{\mathsf{out}} : \{0,1\}^{k_{\mathsf{out}}} \to (\{0,1\}^{\log q_{\mathsf{out}}})^{n_{\mathsf{out}}}$, and
- $\mathsf{Enc}_{\mathsf{in}} : \{0,1\}^{k_{\mathsf{in}}} \to (\{0,1\}^{\log q_{\mathsf{in}}})^{n_{\mathsf{in}}}$,

such that $\log q_{\mathsf{out}} = k_{\mathsf{in}}$ we define the ***concatenated encoding function*** $\mathsf{Enc} : \{0,1\}^{k_{\mathsf{out}}} \to (\{0,1\}^{\log q_{\mathsf{in}}})^{n_{\mathsf{out}} \cdot n_{\mathsf{in}}}$ denoted by $\mathsf{Enc}_{\mathsf{out}} \circ \mathsf{Enc}_{\mathsf{in}}$ as follows: For $i \in [n]$, let $\mathsf{out}(i) \in [n_{\mathsf{out}}]$, and $\mathsf{in}(i) \in [n_{\mathsf{in}}]$ be the values defined so that $i = (\mathsf{out}(i) - 1) \cdot n_{\mathsf{in}} + \mathsf{in}(i)$ we define $\mathsf{Enc}(m)_i = \mathsf{Enc}_{\mathsf{in}}(\mathsf{Enc}_{\mathsf{out}}(m)_{\mathsf{out}(i)})_{\mathsf{in}(i)}$.

We will make use of the following lemma that connects the dual-distance of the outer code to the independence of the columns of the generating matrix of the concatenated code (which will be an important step in arguing column-robustness).

Lemma A.6. *Let* $\mathsf{Enc}_{\mathsf{out}} : \mathbb{F}_q^k \to \mathbb{F}_q^{n_{\mathsf{out}}}$ *be an encoding function for a linear* $[n,k]_q$*-code* C_{out}*, and let* $\mathsf{Enc}_{\mathsf{in}} : \mathbb{F}_2^{\log q} \to \mathbb{F}_2^{n_{\mathsf{in}}}$ *be an encoding function for a linear* $[n_{\mathsf{in}}, \log(q)]_2$*-code* C_{in}*. Let* $\mathbf{G}$ *the generating matrix for the linear code* $[n := n_{\mathsf{in}} \cdot n_{\mathsf{out}}, v := \log(q) \cdot k]_2$ *defined by the code concatenation* $\mathsf{Enc} = \mathsf{Enc}_{\mathsf{out}} \circ \mathsf{Enc}_{\mathsf{in}}$*. Assuming,*

- C_{out} *has dual-distance* $t+1$*,*
- *the generating matrix* $\mathbf{G}_{\mathsf{in}}$ *for the inner code* C_{in} *does not have an all zero columns.*[12]

[12] Note that without lose of generality, we can assume that any linear code does not have an all zero columns in its generating matrix. As otherwise, we can simply delete/replace this column without effecting the rate/distance of the code.

then for every subset $T \subset [n]$, if $|T| \leq t$, and for every $i \neq i \in T$ it holds that $\mathsf{out}(i) \neq \mathsf{out}(i')$ (that is, the indexes are in different outer code symbols), it holds that $\mathsf{rank}([\mathbf{G}]_T) = |T|$.

Proof of Lemma A.6. Let $T \in [n]$ be a subset that satisfies the conditions stated in Lemma A.6. Applying Enc on a uniformly chosen message $m \leftarrow \mathbb{F}_2^v$ yields a distribution $(Z_1, \ldots, Z_n)$ over $\mathbb{F}_2^n$. Note that by Claim A.3 to show that $\mathsf{rank}([\mathbf{G}]_T) = |T|$ it suffices to show that $(Z_i)_{i \in T}$ are uniform and independent.

Let $(W_1, \cdots, W_{n_{\mathsf{out}}}) = \mathsf{Enc}_{\mathsf{out}}(m)$ and note that since C_{out} has a dual-distance $t+1$, by Lemma A.4, $(W_1, \cdots W_{n_{\mathsf{out}}})$ are t-wise independent. Since by assumption, $|T| \leq t$ and for every $i \neq i' \in T$ it holds that $\mathsf{out}(i) \neq \mathsf{out}(i')$, it follows that $(W_{\mathsf{out}(i)})_{i \in T}$ are uniform and independent random variables. Finally, by definition

$$Z_i = \mathsf{Enc}(m)_i = \mathsf{Enc}_{\mathsf{in}}(\mathsf{Enc}_{\mathsf{out}}(m)_{\mathsf{out}(i)})_{\mathsf{in}(i)} = \mathsf{Enc}_{\mathsf{in}}(W_{\mathsf{out}(i)})_{\mathsf{in}(i)} =< W_{\mathsf{out}(i)}, c_{\mathsf{in}(i)} >,$$

where $c_{\mathsf{in}(i)}$ is the $\mathsf{in}(i)$ column of $\mathbf{G}_{\mathsf{in}}$ the generating matrix of the inner code C_{in} (and note that by assumption $c_{\mathsf{in}(i)} \neq 0$). Thus, it follows that $(Z_i)_{i \in T} = (< W_{\mathsf{out}(i)}, c_{\mathsf{in}(i)} >)_{i \in T}$ are uniform and independent since $(W_{\mathsf{out}(i)})_{i \in T}$ are uniform and independent and for a fixed T, for every $i \in T$, $c_{\mathsf{in}(i)} \in \{0,1\}^{\log q}$ is fixed value different from $0^{\log q}$. □

A.2.2 List-Decodable Codes with Column-Robustness Approaching the Blokh-Zyablov Bound.

Proof of Theorem 4.6. Let $k, v \in \mathbb{N}$ be parameters, $k' = k + v$ and let $\mathbf{G}_{k,v} = \mathsf{GenM}(1^k, 1^v)$ be the list-decodable code by Guruswami and Rudra [GR08] for k' guarantied by Theorem 2.6 for our choice of constants $p \in [0, 1/2)$ and $0 < \varepsilon$. Note that by definition, the code is (p, L)-list decodable and has rate $R > R_{BZ}(p) - \varepsilon$.

Let $\mathbf{A} \in \mathbb{F}_2^{k \times n}$ and $\mathbf{B} \in \mathbb{F}_2^{v \times n}$ be defined such that $\mathbf{G} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$. We will use the following lemma, that states that the subcode defined by the generating matrix $\mathbf{B}$ is the result of a code concatenation of an inner code and an outer code where the outer code has dual distance.

Lemma A.7. *When $\log^4(k) < v \leq k^{0.49}$, for some universal constant $0 < c_0$, there exists an outer code $\mathsf{Enc}_{\mathsf{out}} \colon \{0,1\}^v \to (\{0,1\}^a)^{n_{\mathsf{out}}}$ and an inner code $\mathsf{Enc}_{\mathsf{in}} \colon \{0,1\}^a \to \{0,1\}^{n_{\mathsf{in}}}$ such that*

- *$a \leq \log^2(k)$, $\mathsf{Enc}_{\mathsf{in}}$ has constant rate $0 < R_{\mathsf{in}}$ and $n = n_{\mathsf{out}} \cdot n_{\mathsf{in}}$*
- *$\mathsf{Enc}_{\mathsf{out}}$ has dual-distance $t+1$ where $t > v^\alpha$ for any constant $\alpha < 1$,*
- *and $\mathbf{B}$ is the generating matrix for the concatenation code $\mathsf{Enc}_{\mathsf{out}} \circ \mathsf{Enc}_{\mathsf{in}}$.*

We prove Lemma A.7 below, but first we show that the column-robustness property holds. Let $v < k^{0.49}$. By Lemma A.7, there exists $\mathsf{Enc}_{\mathsf{out}} \colon \{0,1\}^v \to (\{0,1\}^a)^{n_{\mathsf{out}}}$ and $\mathsf{Enc}_{\mathsf{in}} \colon \{0,1\}^a \to \{0,1\}^{n_{\mathsf{in}}}$ such that $\mathbf{B}$ is the generating matrix of $\mathsf{Enc}_{\mathsf{out}} \circ \mathsf{Enc}_{\mathsf{in}}$.

Let $T \subset [n]$ be a random subset of size t and note that by Lemma A.7, $\mathsf{Enc}_{\mathsf{out}}$ has dual-distance $t+1$ and that $t > v^\alpha$ for any constant $\alpha < 1$. Recall that for $i \in [n]$, we use $\mathsf{out}(i) \in [n_{\mathsf{out}}]$, and $\mathsf{in}(i) \in [n_{\mathsf{in}}]$ to denote the values such that $i = (\mathsf{out}(i) - 1) \cdot n_{\mathsf{in}} + \mathsf{in}(i)$, (see Definition A.5). Let E be the event that for every distinct $i, j \in T$ it holds that $\mathsf{out}(i) \neq \mathsf{out}(j)$, and note that by a simple union bound over all pairs,

$$\begin{aligned} \mathbf{Pr}[E] &\geq 1 - |T|^2 \cdot n_{\mathsf{out}}/n \\ &\geq 1 - v^2 \cdot \left(\left(\frac{1}{R_{\mathsf{in}}} \cdot a \right) / n \right) \\ &\geq 1 - v^2 \cdot ((\log^5(k)/n) \\ &\geq 1 - v^2 \cdot (1/v^{2.01}) \\ &\geq 1 - (1/v^{0.01}) \end{aligned} \tag{A.3}$$

Finally note that since $|T| < t+1$, by Lemma A.6, whenever the event E holds, it follows that $\mathsf{rank}[\mathbf{B}]_T = t$, which implies that $\mathbf{B}$ is (q, α)-well formed for any $\alpha < 1$ and $q = 1 - (1/v^{0.01})$. This concludes the proof for the columns robustness property. □

Proof of Lemma A.7. We will make use of some observations made in [SW25a] on the construction of [GR08]. Specifically, [SW25a] showed the following claims which we adapt to our notation (adding relevant parameters implicit in the proof of [SW25a]):

Claim A.8 ([SW25a], Claim A.2 in Appendix A)**.** *There exists a constant $0 < \beta$ which depends on ε and p, such that when considering the matrix $\mathbf{Z} \in \{0,1\}^{z \times n}$ defined to be the last $z := \beta n$ rows of $\mathbf{G}_{k,v}$, it holds that $\mathbf{Z}$ is a generating matrix for a (normal code concatenation) $C_{\mathsf{out}} \circ C_{\mathsf{in}}$ where C_{out} is a folded Reed-Solomon code over the field F_{2^a} for some integer $1 < a \leq \log^2(n)$ and the inner code has constant rate $R_{\mathsf{in}} > 0$.*[13]

Claim A.9 ([SW25a], Claim A.3 in Appendix A)**.** *Let C_{out} be a folded Reed-Solomon code over the field F_{2^a} for some integer $a > 1$. Then for any $v < z$ the code defined so that for every $x \in \{0,1\}^v$: $\mathsf{Enc}_{\mathsf{out}}(x) := C_{\mathsf{out}}(0^{z-v}, x)$ is also a folded Reed-Solomon code.* [14]

Proof of Lemma A.7. By Claim A.8, for $z \geq \alpha \cdot n$ for some constant α, the matrix $\mathbf{Z}$ defined to be the last z rows of $\mathbf{G}_{k,\lambda}$ is a generating matrix for the concatenated code $C_{\mathsf{out}} \circ C_{\mathsf{in}}$ where C_{out} is a folded Reed-Solomon code over the field F_{2^a} for some integer $1 < a \leq \log^2(n)$ where the inner code has constant rate $R_{\mathsf{in}} > 0$. Note that since $v \leq k^{0.49}$, it follows that $v < z$. Recall that $\mathbf{B}$ is the matrix defined by the last v rows of $\mathbf{G}_{k,\lambda}$, which are the also the last v rows of $\mathbf{Z}$. Let Enc denote the code generated by $\mathbf{B}$, it follows that for every message $x \in \{0,1\}^v$:

$$\begin{aligned}\mathsf{Enc}(x) &= x \cdot \mathbf{B} \\ &= (0^{z-v}, x) \cdot \mathbf{Z} \\ &= C_{\mathsf{out}} \circ C_{\mathsf{in}}(0^{z-v}, x) \\ &= \mathsf{Enc}_{\mathsf{out}} \circ C_{\mathsf{in}}(x)\end{aligned}$$

Where $\mathsf{Enc}_{\mathsf{out}}$ is the folded Reed-Solomon code over the field F_{2^a} guarantied by Claim A.9. Taking $\mathsf{Enc}_{\mathsf{in}} := C_{\mathsf{in}}$, it follows that $\mathbf{B}$ is the generating matrix for the concatenated code $\mathsf{Enc}_{\mathsf{out}} \circ \mathsf{Enc}_{\mathsf{in}}$, where $\mathsf{Enc}_{\mathsf{out}}$ is the folded Reed-Solomon code over the field F_{2^a}, and $1 < a \leq \log^2(n) \leq \log^3(k)$ (where the last inequality, follows since the rate of the overall code is a positive constant). To conclude, we note the well known fact that a folded Reed-Solomon code over the field F_{2^a} has dual-distance $t+1$ for $t = v/a$ which by our choice of parameters if $\log^4(k) < v$ it means that $t > v^\alpha$ for any constant $\alpha < 1$. □

B Trapdoor Hashing for Circuits

In this section, we present a direct construction of trapdoor hashing for circuit predicates via laconic function evaluation (LFE). This construction can be seen as a modification and simplification of the construction of *reverse* trapdoor hashing [AMR25] which is sufficient for our application.

For this section, we will use $\boldsymbol{G}$ to refer to the standard gadget matrix $I_n \otimes (1, 2, \ldots, 2^{\log q})$ and $\boldsymbol{G}^{-1}(\cdot)$ to be the standard bit-decomposition function where $\boldsymbol{G} \cdot \boldsymbol{G}^{-1}(\boldsymbol{M}) = \boldsymbol{M}$. Additionally, we will use $\lceil \cdot \rfloor_2 : \mathbb{Z}_q \to \{0,1\}$ to denote rounding function mapping $[-q/4, q/4) \to 0$ and $[q/4, 3q/4) \to 1$.

Definition B.1 (Laconic Function Evaluation Pre-Encoding)**.** A laconic function evaluation (LFE) pre-encoding scheme Π_{LFE} for a function family $\mathcal{F} = \{\mathcal{F}_\lambda\}$ is a tuple of efficient algorithms (Setup, Hash, PreEnc, Eval) with the following syntax:

- $\mathsf{Setup}(1^\lambda, 1^n) \to \mathsf{pk}$: The setup algorithm Setup takes as input security parameter $\lambda \in \mathbb{N}$ and input length $n \in \mathbb{N}$ to sample a public key pk.
- $\mathsf{Hash}(\mathsf{pk}, f) \to \boldsymbol{A}_f$: The hash algorithm Hash takes in the public key pk along with function $f \in \mathcal{F}_\lambda$ and deterministically outputs a digest matrix $\boldsymbol{A}_f \in \mathbb{Z}_q^{k \times k \cdot \log q}$.

[13] The statement made in [SW25a], Claim A.2 in the appendix A, does not explicitly give a bound on a and R_{in}, however an inspection of the proof and the construction of [GR08] reveals that for any choice of constant p, the bounds stated on a and R_{in} hold.

[14] This claim is stated using generating matrices in [SW25a] and is more general. For our purse, we require only this simpler version.

- $\mathsf{PreEnc}(\mathsf{pk}, x) \to (E, \mathbf{s})$: The pre-encoding PreEnc takes in a public key pk and input $x \in \{0,1\}^n$ and outputs partial encoding E and secret $\mathbf{s}$ with last entry -1.
- $\mathsf{Eval}(E, f)$: On input partial encoding E, function f, the encode algorithm deterministically outputs a vector $\mathbf{v} \in \mathbb{Z}_q^{k \cdot \log q}$.

We require the following properties:

- **Perfect $\tilde{B}$-Special Correctness:** For all $\lambda \in \mathbb{N}$, $f : \{0,1\}^n \to \{0,1\} \in \mathcal{F}_\lambda$, $x \in \{0,1\}^n$, all $(E, \mathbf{s})$ in the support of $\mathsf{PreEnc}(\mathsf{pk}, x)$ and $\boldsymbol{A}_f = \mathsf{Hash}(\mathsf{pk}, f)$

$$\mathsf{Eval}(E, f) = \mathbf{s}\boldsymbol{A}_f + (\mathbf{s} \cdot f(x)) \cdot \boldsymbol{G} + \tilde{\mathbf{e}}^\top$$

  where $||\tilde{\mathbf{e}}||_\infty \leq \tilde{B}$.

- **Pre-Encoding Privacy:** For a bit $b \in \{0,1\}$ and a security parameter λ, the pre-encoding privacy game between an adversary $\mathcal{A}$ and a challenger $\mathsf{ExpPP}^{(b)}(\mathcal{A}, \lambda)$ is as follows:

  1. On input a parameter λ, the adversary outputs an input length 1^n and inputs $x_0, x_1 \in \mathcal{F}$.
  2. The challenger samples public key $\mathsf{pk} \leftarrow \mathsf{Setup}(1^\lambda)$, pre-encoding $(E, \mathbf{s}) \leftarrow \mathsf{PreEnc}(\mathsf{pk}, x_b)$, and sends (pk, E) to $\mathcal{A}$.
  3. The adversary outputs a guess $b' \in \{0,1\}$, which is the output of the experiment.

  We say that Π_{LFE} satisfies pre-encoding privacy if for all polynomial time adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that

$$\left|\mathbf{Pr}[\mathsf{ExpPP}^{(0)}(\mathcal{A}, \lambda) = 1] - \mathbf{Pr}[\mathsf{ExpPP}^{(1)}(\mathcal{A}, \lambda) = 1]\right| = \mathsf{negl}(\lambda)$$

- **Succinctness:** We require the size of the digest matrix $\boldsymbol{A}_f$ is independent of the input length n.

As observed in [AMR25], these requirements are satisfied by many existing LFE schemes, including:

- The direct LWE construction of LFE [QWW18] gives an LFE pre-encoding for circuits of bounded depth d assuming plain LWE. Here, the digest size grows as some fixed $\mathsf{poly}(\lambda, d)$.
- The construction of [HLL23] gives an LFE pre-encoding for circuits of unbounded depth assuming circular secure LWE. Here, the digest size grows as some fixed $\mathsf{poly}(\lambda)$.

We can now present our construction of trapdoor hashing:

Construction B.2. For a function family $\mathcal{F}$, this construction makes use of the following ingredients:

- An LFE pre-encoding scheme Π_{LFE} for the family of universal circuits $\mathcal{F}'$ supporting evaluation on functions in $\mathcal{F}$ on some super-polynomial modulus q. We will use the fact that a universal circuit supporting the family of depth d size s circuits can be implemented in depth $\mathsf{poly}(d)$ and size $\mathsf{poly}(s)$.

The construction proceeds as follows:

- $\mathsf{Setup}(1^\lambda, 1^n)$: Sample public keys

$$\mathsf{pk}_{\mathsf{LFE}} \leftarrow \Pi_{\mathsf{LFE}}.\mathsf{Setup}(1^\lambda)$$

  and output $\mathsf{hk} = \mathsf{pk}_{\mathsf{LFE}}$.

- $\mathsf{Gen}(\mathsf{hk}, f)$: Parse $\mathsf{hk} = \mathsf{pk}_{\mathsf{LFE}}$. Sample $r \xleftarrow{R} \mathbb{Z}_q$, and compute

$$(E, \mathbf{s}) \leftarrow \Pi_{\mathsf{LFE}}.\mathsf{PreEnc}(\mathsf{pk}_{\mathsf{LFE}}, f^{15})$$

  Output $\mathsf{ek} = (\mathsf{pk}_{\mathsf{LFE}}, E, r)$, $\mathsf{td} = (\mathbf{s}, r)$.

[15] where we interpret the description of f as some bitstring

- $\mathsf{H}(\mathsf{hk}, x)$: Parse $\mathsf{hk} = \mathsf{pk}_{\mathsf{LFE}}$. Define the universal circuit $U[x](C) = C(x)$.

$$\boldsymbol{A}_x \leftarrow \Pi_{\mathsf{LFE}}.\mathsf{Hash}(\mathsf{pk}_{\mathsf{LFE}}, U[x])$$

  Output $\sigma = \boldsymbol{A}_x$.

- $\mathsf{E}(\mathsf{ek}, x)$: Let $\mathbf{u}$ be the kth standard basis vector. Parse $\mathsf{ek} = (\mathsf{pk}_{\mathsf{LFE}}, E, r)$ and compute

$$e = \lceil r - \Pi_{\mathsf{LFE}}.\mathsf{Eval}(E, U[x]) \cdot \boldsymbol{G}^{-1}(-q/2 \cdot \mathbf{u}^\top) \rfloor_2$$

  and output bit e.

- $\mathsf{D}(\mathsf{td}, \sigma)$: Parse $\mathsf{td} = (\mathbf{s}, r)$ and $\sigma = \boldsymbol{A}_x$. Compute

$$e' = \lceil -r + \mathbf{s}^\top \cdot \boldsymbol{A}_x \cdot \boldsymbol{G}^{-1}(-q/2 \cdot \mathbf{u}^\top) \rfloor_2$$

  and output bit e'.

Theorem B.3 (Enhanced Correctness)**.** *Assuming Π_{LFE} satisfies perfect $\tilde{B}$-special correctness and $q \gg \tilde{B}$. Then Construction B.2 satisfies $(1 - \mathsf{negl}(\lambda))$-enhanced correctness.*

Proof. Let $s_0 = r - \Pi_{\mathsf{LFE}}.\mathsf{Eval}(E, U[x]) \cdot \boldsymbol{G}^{-1}(-q/2 \cdot \mathbf{u}^\top)$ and $s_1 = -r + \mathbf{s}^\top \cdot \boldsymbol{A}_f \cdot \boldsymbol{G}^{-1}(-q/2 \cdot \mathbf{u}^\top)$ so that we can write

$$\begin{aligned}
e + e' &= \lceil s_0 \rfloor_2 + \lceil s_1 \rfloor_2 \\
&\approx \lceil s_0 + s_1 \rfloor_2 \\
&= \lceil \Pi_{\mathsf{LFE}}.\mathsf{Eval}(E, U[x]) \cdot \boldsymbol{G}^{-1}(-q/2 \cdot \mathbf{u}^\top) + \mathbf{s} \cdot \boldsymbol{A}_x \cdot \boldsymbol{G}^{-1}(-q/2 \cdot \mathbf{u}^\top) \rfloor_2 \\
&= \lceil (\mathbf{s} \cdot \boldsymbol{A}_x + \mathbf{s} f(x) \cdot \boldsymbol{G} + \tilde{e}^\top) \cdot \boldsymbol{G}^{-1}(-q/2 \cdot \mathbf{u}^\top) + \mathbf{s} \cdot \boldsymbol{A}_x \cdot \boldsymbol{G}^{-1}(-q/2 \cdot \mathbf{u}^\top) \rfloor_2 \\
&= \lceil -f(x) \cdot q/2 \cdot \mathbf{s} \cdot \mathbf{u}^\top + \tilde{e} \cdot \boldsymbol{G}^{-1}(-q/2 \cdot \mathbf{u}^\top) \rfloor_2
\end{aligned}$$

By $\tilde{B}$-special correctness of Π_{LFE}, we can bound $|\tilde{e} \cdot \boldsymbol{G}^{-1}(-q/2 \cdot \mathbf{u}^\top)|$ with $\log q \cdot \tilde{B}$. Since $\mathbf{s} \cdot \mathbf{u}^\top = -1$, we can reason that

$$s_0 + s_1 = f(x) \cdot \frac{q}{2} + \hat{e} \bmod q$$

for some $|\hat{e}| \leq \log q \cdot \tilde{B}$. Thus, we can see correctness only fails when $\lceil s_0 \rfloor_2 + \lceil s_1 \rfloor_2 \neq \lceil s_0 + s_1 \rfloor_2$. Observe that when s_1 is not within distance $|\tilde{e}|$ of a rounding boundary $q/4, 3q/4$, correctness hold on all circuits $U[x]$/inputs x. Since conditional on a fixed digest value $\sigma = \boldsymbol{A}_x$, s_1 is otherwise independent of input x we can reason by uniform randomness of $r \xleftarrow{\mathrm{R}} \mathbb{Z}_q$ that

$$\Pr_{(\mathsf{ek}_f, \mathsf{td}) \leftarrow \mathsf{Gen}(\mathsf{hk}, f)} [\forall x \in \{0,1\}^n : \mathsf{H}(\mathsf{hk}, x) = \boldsymbol{A}_x \ \ \lceil s_0 \rfloor_2 + \lceil s_1 \rfloor_2 = \mathsf{E}(\mathsf{ek}_f, x) + \mathsf{D}(\mathsf{td}, \sigma) = f(x)] = 1 - \frac{|\hat{e}|}{q},$$

which is all but negligible for superpolynomial q, completing our argument. □

Theorem B.4 (Compactness)**.** *Assuming Π_{LFE} is succinct with respect to $\mathcal{F}'$, Construction B.2 is compact.*

Proof. Since the hash $\sigma = \mathsf{H}(\mathsf{hk}, x)$ is exactly the LFE digest $\boldsymbol{A}_x = \Pi_{\mathsf{LFE}}.\mathsf{Hash}(\mathsf{pk}_{\mathsf{LFE}}, U[x])$, this follows immediately. □

Theorem B.5 (Function Privacy)**.** *Assuming Π_{LFE} satisfies pre-encoding privacy, Construction B.2 satisfies function privacy.*

Proof. Since our evaluation key ek_f corresponds exactly to the pre-encoding E (in addition to public key $\mathsf{pk}_{\mathsf{LFE}}$ and independent randomness r), this follows immediately. □

Corollary B.6. *For any $d = d(\lambda) \in \mathsf{poly}(\lambda)$, there exists trapdoor hashing for depth-d circuit predicates satisfying enhanced $(1 - \mathsf{negl})$-correctness, function privacy, and compactness assuming LWE.*

Corollary B.7. *For any $s = s(\lambda, n) \in \mathsf{poly}(\lambda, n)$, there exists trapdoor hashing for trapdoor hashes for any size-s circuit predicates satisfying enhanced $(1 - \mathsf{negl})$-correctness, function privacy, and compactness assuming circularly secure LWE.*

C Public Coin Multi-Input Correlation Intractability

In this section, we show that if our trapdoor hash satisfies an additional explainability property, then our resulting CI-Hash (and seeded codes) can be made public coin. Informally, this property requires that given a hash or evaluation key hk, ek, there exists efficient algorithms which can 'explain' the randomness used to generate said keys. This allows us to rely on the usual properties of our trapdoor hash while 'simulating' our public coins in the security game from the challenger's keys. We note such a property can be loosely viewed as an extension of indifferentiability [MRH04, RSS11], and similar properties for other functionalities have been considered in other contexts, such as random oracle sampling [LW22]. We show that this property can be realized by the DDH and LWE constructions of [DGI+19a]. We will use $f(x;\rho)$ to denote randomized functionalities with input x and randomness ρ.

Definition C.1 (Public Coin Trapdoor Hashing)**.** Given a trapdoor hash $\Pi_{\mathsf{TDH}} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{H}, \mathsf{E}, \mathsf{D})$ for a predicate family $\mathcal{F}$, suppose there exists additional efficient randomized algorithms $(\mathsf{SetupExp}, \mathsf{GenSim}, \mathsf{GenExp})$ with properties described below. We assume without loss of generality there exists some fixed polynomial $r = r(\lambda, k)$ bits of randomness used by all algorithms in Π_{TDH}.

- **Simulation Soundness:** For a bit $b \in \{0,1\}$ and a security parameter λ, the explainability game between an adversary $\mathcal{A}$ and a challenger ExpSim is as follows:

  1. On input security parameter λ, the adversary outputs an input length 1^k.
  2. The challenger samples hash key $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$,
     - If $b = 0$, set $\mathsf{ek} \leftarrow \mathsf{GenSim}(\mathsf{hk})$.
     - If $b = 1$, set $(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}(\mathsf{hk}, \mathbf{0}^k)$.

     sending $(\mathsf{hk}, \mathsf{ek})$ to $\mathcal{A}$.
  3. The adversary outputs a guess $b' \in \{0,1\}$, which is the output of the experiment.

  We say that Π_{TDH} satisfies simulation soundness if for all polynomial time adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that

  $$\left|\mathbf{Pr}[\mathsf{ExpSim}^{(0)}(\mathcal{A}, \lambda) = 1] - \mathbf{Pr}[\mathsf{ExpSim}^{(1)}(\mathcal{A}, \lambda) = 1]\right| = \mathsf{negl}(\lambda)$$

- **Explainability:** For a bit $b \in \{0,1\}$ and a security parameter λ, the explainability game between an adversary $\mathcal{A}$ and a challenger ExpExpl is as follows:

  1. On input security parameter λ, the adversary outputs an input length 1^k.
  2. The challenger samples hash key $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$,
     - If $b = 0$, sample the following distribution:
       - $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$.
       - $\mathsf{ek} \leftarrow \mathsf{GenSim}(\mathsf{hk})$.
       - $\rho_0 \leftarrow \mathsf{SetupExp}(\mathsf{hk})$.
       - $\rho_1 \leftarrow \mathsf{GenExp}(\mathsf{hk}, \mathsf{ek})$.
     - If $b = 1$, sample the following distribution:
       - $\rho_0 \xleftarrow{R} \{0,1\}^r$.
       - $\rho_1 \xleftarrow{R} \{0,1\}^r$.
       - $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k; \rho_0)$.
       - $\mathsf{ek} \leftarrow \mathsf{GenSim}(\mathsf{hk}; \rho_1)$.

     sending $(\mathsf{hk}, \mathsf{ek}, \rho_0, \rho_1)$ to $\mathcal{A}$.
  3. The adversary outputs a guess $b' \in \{0,1\}$, which is the output of the experiment.

  We say that Π_{TDH} is explainable if for all polynomial time adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that

  $$\left|\mathbf{Pr}[\mathsf{ExpExpl}^{(0)}(\mathcal{A}, \lambda) = 1] - \mathbf{Pr}[\mathsf{ExpExpl}^{(1)}(\mathcal{A}, \lambda) = 1]\right| = \mathsf{negl}(\lambda)$$

C.1 Public Coin Multi-Input CI From Explainability

We now present a modified hash function Π_{Hash} whose hash key consists of uniform randomness, and show that trapdoor hashing scheme which satisfies the additional properties described in the previous section yields a public coin CI-Hash (which also directly yields a public coin seeded code through Lemma 4.7).

Theorem C.2. *Suppose Π_{TDH} is an* explainable *trapdoor hash for predicate family $\mathcal{F}$, then the hash function Π_{Hash} defined as*

- $\mathsf{Gen}(1^\lambda, 1^k, 1^v)$: *Sample* $\rho_0, \rho_1 \xleftarrow{R} \{0,1\}^r$.
- $\mathsf{Hash}(K = (\rho_0, \rho_1), x)$: *Compute* $\mathsf{hk} \leftarrow \Pi_{\mathsf{TDH}}.\mathsf{Setup}(1^\lambda, 1^k; \rho_0)$ *and* $\mathsf{ek} \leftarrow \Pi_{\mathsf{TDH}}.\mathsf{GenSim}(\mathsf{hk}; \rho_1)$. *Output* $y \leftarrow \Pi_{\mathsf{TDH}}.\mathsf{E}(\mathsf{ek}, x)$.

Then there exists polynomial $s = s(\lambda)$ (independent of n, v) such that Π_{Hash} is t-correlation intractable for the sparse efficiently checkable relation family $\mathcal{R}_{s,\mathcal{F}}$. Observe that this Π_{Hash} is public coin.

Proof. At a high level, the explainability of our trapdoor hash allow us to simulate the 'true randomness' (ρ_0, ρ_1) given any hash key hk and evaluation key ek. This allows us to directly reduce to the security of our previous construction Theorem 5.10. Consider the following sequence of experiments:

- $\mathsf{Exp}^{(\mathcal{R})}_{\mathsf{real}}$: This is the real t-input correlation intractability game for the relation family $\mathcal{R}$ and public coin hash Π_{Hash} as described above.
  1. The adversary samples $R, \mathbf{f} = f_1, \ldots, f_v$ which it sends to the challenger.
  2. The challenger samples the hash key (ρ_0, ρ_1), which they send to the adversary, as follows:
     - $\rho_0, \rho_1 \xleftarrow{R} \{0,1\}^r$.
  3. The adversary sends inputs $x^{(t)}, \ldots, x^{(t)}$ to the challenger.
  4. The challenger computes $y^{(i)} = \mathsf{E}(\mathsf{ek}, x^{(i)})$ for all $i \in [t]$ and sets
  $$R(y^{(1)} - \mathbf{f}(x^{(1)}), \ldots, y^{(t)} - \mathbf{f}(x^{(t)}))$$
  as the output of the experiment.
- $\mathsf{Exp}^{(\mathcal{R})}_{1}$: In this experiment, the challenger samples ρ_0, ρ_1 as an 'explanation' of the randomness of a hash key hk and evaluation key ek they generate.
  2. The challenger samples the hash key (ρ_0, ρ_1), which they send to the adversary, as follows:
     - Sample $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$.
     - Sample $\mathsf{ek} \leftarrow \mathsf{GenSim}(\mathsf{hk})$.
     - Sample $\rho_0 \leftarrow \mathsf{SetupExp}(\mathsf{hk})$.
     - Sample $\rho_1 \leftarrow \mathsf{GenExp}(\mathsf{hk}, \mathsf{ek})$.
- $\mathsf{Exp}^{(\mathcal{R})}_{2}$: In this experiment, the challenger switches to sampling ek using the real Gen algorithm.
  2. The challenger samples the hash key (ρ_0, ρ_1), which they send to the adversary, as follows:
     - Sample $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$.
     - Sample $\rho_0 \leftarrow \mathsf{SetupExp}(\mathsf{hk})$.
     - Sample $(\mathsf{ek}, \mathsf{td}) \leftarrow \mathsf{Gen}(\mathsf{hk}, \mathbf{0}^{k \times v})$.
     - Sample $\rho_1 \leftarrow \mathsf{GenExp}(\mathsf{hk}, \mathsf{ek})$.

Lemma C.3. *Assuming Π_{TDH} is* Setup*-explainable, then for all efficient adversaries $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\lambda)$ *such that*

$$\left|\mathbf{Pr}[\mathsf{Exp}^{(\mathcal{R})}_{\mathsf{real}}(\mathcal{A}, \lambda) = 1] - \mathbf{Pr}[\mathsf{Exp}^{(\mathcal{R})}_{1}(\mathcal{A}, \lambda) = 1]\right| = \mathsf{negl}(\lambda).$$

Proof. Suppose $\mathcal{A}$ is an adversary which has some nonnegligible advantage in distinguishing $\mathsf{Exp}^{(\mathcal{R})}_{\mathsf{real}}$ and $\mathsf{Exp}^{(\mathcal{R})}_{1}$. Consider the following reduction $\mathcal{B}$ which breaks explainability with the same probability.

1. Reduction $\mathcal{B}$ begins by simulating $\mathcal{A}$, who sends input length 1^k, output length 1^v and challenge relation $(R, \mathbf{f})$ to $\mathcal{B}$, who sends $1^k, 1^v$ to the challenger.
2. Reduction $\mathcal{B}$ receives $\mathsf{hk}, \mathsf{ek}, \rho_0, \rho_1$ and sends (ρ_0, ρ_1) to $\mathcal{A}$.
3. Adversary $\mathcal{A}$ outputs a guess bit b' which $\mathcal{B}$ simply forwards to the challenger.

Observe that when ρ_0, ρ_1 are true randomness, $\mathcal{A}$ receives the distribution of $\mathsf{Exp}^{(\mathcal{R})}_{0}$, and when ρ_0, ρ_1 is sampled from SetupExp, GenExp respectively, $\mathcal{A}$ receives the distribution of $\mathsf{Exp}^{(\mathcal{R})}_{1}$. □

Lemma C.4. *Assuming Π_{TDH} is simulation sound, then for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that*

$$\left|\mathbf{Pr}[\mathsf{Exp}^{(\mathcal{R})}_{1}(\mathcal{A}, \lambda) = 1] - \mathbf{Pr}[\mathsf{Exp}^{(\mathcal{R})}_{2}(\mathcal{A}, \lambda) = 1]\right| = \mathsf{negl}(\lambda).$$

Proof. Suppose $\mathcal{A}$ is an adversary which has some nonnegligible advantage in distinguishing $\mathsf{Exp}^{(\mathcal{R})}_{1}$ and $\mathsf{Exp}^{(\mathcal{R})}_{2}$. Consider the following reduction $\mathcal{B}$ which breaks simulation soundness with the same probability.

1. Reduction $\mathcal{B}$ begins by simulating $\mathcal{A}$, who sends input length 1^k, output length 1^v and challenge relation $(R, \mathbf{f})$ to $\mathcal{B}$, who sends $1^k, 1^v$ to the challenger.
2. Reduction $\mathcal{B}$ receives hash key hk, ek, samples $\rho_0 \leftarrow \mathsf{SetupExp}(\mathsf{hk})$, $\rho_1 \leftarrow \mathsf{GenExp}(\mathsf{hk}, \mathsf{ek})$, and sends (ρ_0, ρ_1) to $\mathcal{A}$.
3. Adversary $\mathcal{A}$ outputs a guess bit b' which $\mathcal{B}$ simply forwards to the challenger.

Observe that when ek is sampled from $\mathsf{GenSim}(\mathsf{hk})$, $\mathcal{A}$ receives the distribution of $\mathsf{Exp}^{(\mathcal{R})}_{1}$, and when similarly receives the distribution of $\mathsf{Exp}^{(\mathcal{R})}_{2}$ when ek is from $\mathsf{Gen}(\mathsf{hk}, \mathbf{0}^k)$. □

Lemma C.5. *Assuming Π_{TDH} is function private, collision resistant, compact, t-universal, and semi-perfect ϵ-close, then for all efficient adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that*

$$\mathbf{Pr}[\mathsf{Exp}^{(\mathcal{R})}_{2}(\mathcal{A}, \lambda) = 1] = \mathsf{negl}(\lambda).$$

Proof. Observe in $\mathsf{Exp}^{(\mathcal{R})}_{2}$, our challenger derives ρ_0, ρ_1 entirely from a hash key hk and evaluation key ek generated exactly as in Theorem 5.10. Thus, to argue security, we can directly reduce an adversary in $\mathsf{Exp}^{(\mathcal{R})}_{3}$ to that of the real security experiment $\mathsf{Exp}^{(\mathcal{R})}_{\mathsf{real}}$ in Theorem 5.10. Concretely, let $\mathcal{A}$ be an adversary in $\mathsf{Exp}^{(\mathcal{R})}_{3}$ above. We will reduce this to adversary $\mathcal{B}$ against the security of $\mathsf{Exp}^{(\mathcal{R})}_{\mathsf{real}}$ in Theorem 5.10.

1. Reduction $\mathcal{B}$ begins by simulating $\mathcal{A}$, who sends input length 1^k, output length 1^v and challenge relation $(R, \mathbf{f})$, which $\mathcal{B}$ simply forwards to the challenger.
2. Reduction $\mathcal{B}$ receives hash key $K = (\mathsf{hk}, \mathsf{ek})$. $\mathcal{B}$ runs $\rho_0 \leftarrow \mathsf{SetupExp}(1^\lambda, 1^k)$ and $\rho_1 \leftarrow \mathsf{GenExp}(\mathsf{hk})$, returning hash key (ρ_0, ρ_1) to $\mathcal{A}$.
3. $\mathcal{A}$ sends inputs $x^{(1)}, \ldots, x^{(t)}$, which $\mathcal{B}$ simply forwards to the challenger.

Recall in Theorem 5.10, $\mathcal{B}$ receives hash key K generated as $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$ and $\mathsf{ek} \leftarrow \mathsf{Gen}(\mathsf{hk}, \mathbf{0}^{k \times v})$. Since applying SetupExp, GenExp to hk, ek respectively is exactly the distribution received by $\mathcal{A}$ in $\mathsf{Exp}^{(\mathcal{R})}_{2}$, $\mathcal{B}$ accurately simulated $\mathcal{A}$ and since $\mathcal{B}$ submits the same challenge relation, succeeds with the exact same probability. □

Taking together Lemmas C.3 to C.5 gives us the proof of our theorem. □

Corollary C.6. *Applying Theorem C.2 to the constructions described below, there exists public coin t-correlation intractable hashes for the sparse efficiently checkable relation family $\mathcal{R}_{s,\mathcal{F}}$ for some fixed polynomial $s = s(\lambda)$ and $\mathcal{F}$ as the family of constant degree functions assuming DDH or LWE.*

C.2 Explainability of [DGI+19a] Trapdoor Hashing

We now recall the assumptions and constructions of trapdoor hashing presented in [DGI+19a] (in particular those from DDH/LWE), and show that they satisfy our notion of explainability. For simplicity, we will only present the $\mathsf{hk} \leftarrow \mathsf{Setup}(1^\lambda, 1^k)$ and $(\mathsf{ek}, _) \leftarrow \mathsf{Gen}(\mathsf{hk}, \mathbf{0}^k)$ algorithms of the respective trapdoor hash constructions, as these are the only portions relevant to our new properties.

C.2.1 Learning with Errors

We begin by recalling the standard learning with errors (LWE) assumption [Reg05]:

Assumption C.7 (Learning with Errors [Reg05])**.** Let λ be a security parameter and $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$, and $\chi = \chi(\lambda)$ be lattice parameters. We say that the $\mathsf{LWE}_{n,m,q,\chi}$ assumption holds if for all PPT adversaries $\mathcal{A}$:

$$\left|\mathbf{Pr}\left[\mathcal{A}\left(\mathbf{A}, \mathbf{s}^\intercal\mathbf{A} + \mathbf{e}^\intercal\right) = 1\right] - \mathbf{Pr}\left[\mathcal{A}\left(\mathbf{A}, \mathbf{u}^\intercal\right) = 1\right]\right| = \mathsf{negl}(\lambda)$$

where $\mathbf{A} \xleftarrow{R} \mathbb{Z}_q^{n\times m}, \mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n, \mathbf{e} \leftarrow \chi^m, \mathbf{u} \xleftarrow{R} \mathbb{Z}_q^m$.

The Setup, Gen algorithms of this trapdoor hash scheme are as follows:

- $\mathsf{Setup}(1^\lambda, 1^k)$: On input security parameter λ and input length k, Let q, χ, n be the corresponding LWE parameters:
  - Sample $\boldsymbol{A} \xleftarrow{R} \mathbb{Z}_q^{n\times 2k}$
  - Output $\mathsf{hk} = \boldsymbol{A}$
- $\mathsf{Gen}(\mathsf{hk}, \mathbf{0}^k)$: On input hash key hk:
  - Sample $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n, \mathbf{e} \xleftarrow{R} \chi^{2k}$.
  - Set $\mathbf{u} = \mathbf{s}^\top\boldsymbol{A} + \mathbf{e}$.
  - Sample PRF Key $K \xleftarrow{R} \{0,1\}^\lambda$.
  - Output $\mathsf{ek}_0 = (\mathsf{hk}, \mathbf{u}, K)$.

To argue explainability, we define $\mathsf{GenSim}(\mathsf{hk})$ as follows:

- Sample $\mathbf{u}' \xleftarrow{R} \mathbb{Z}_q^{2n}, K \xleftarrow{R} \{0,1\}^\lambda$.
- Output $(\mathsf{hk}, \mathbf{u}', K)$.

Observe that GenSim only differs from Gen in the distribution of $\mathbf{u}'$ as a uniform random vector in $\mathbb{Z}_q^{2n}$, simulation soundness follows from a straightforward reduction to LWE. Furthermore, since Setup, GenSim only make use of uniform $\mathbb{Z}_q$ and bit elements, this randomness can simply be recovered from said output, giving explainability.

C.2.2 Decisional Diffie Hellman

We begin by recalling the decisional Diffie Hellman (DDH) assumption [DH76]:

Assumption C.8 (Decisional Diffie Hellman (DDH) Assumption)**.** Let λ be the security parameter, and $\mathcal{G} = (\mathbb{G}, g, p)$ be the description of a group. We say $\mathcal{G}$ is DDH hard if for all PPT adversaries $\mathcal{A}$:

$$\left|\mathbf{Pr}[\mathcal{A}(\mathcal{G}, g^a, g^b, g^{ab}) = 1] - \mathbf{Pr}[\mathcal{A}(\mathcal{G}, g^a, g^b, g^c) = 1]\right| = \mathsf{negl}(\lambda)$$

where $a, b, c \xleftarrow{R} \mathbb{Z}_p$.

Corollary C.9. *It follows from a straightforward hybrid argument that for all $n, m \in \mathsf{poly}(\lambda)$,*

$$\left|\mathbf{Pr}[\mathcal{A}(\mathcal{G}, g^{\mathbf{v}^\top \mathbf{w}}) = 1] - \mathbf{Pr}[\mathcal{A}(\mathcal{G}, g^{U}) = 1]\right| = \mathsf{negl}(\lambda)$$

where $\mathbf{v} \xleftarrow{R} \mathbb{Z}_p^n, \mathbf{w} \xleftarrow{R} \mathbb{Z}_p^m, \boldsymbol{U} \xleftarrow{R} \mathbb{Z}_p^{n \times m}$.

To show explainability in the group setting, we will also require a 'hash-to-point' function for the group we are working over. This is essentially an explainable sampler for group elements. Like our general explainability definition, this is requirement generally comes up in the context of instantiating constructions using random oracles to groups and is known to exist for many cryptographic groups such as elliptic curve groups [Ica09, BCI+10].

$$(h, \rho) : h \leftarrow \mathsf{Samp}(\mathbb{G}; \rho) \approx_c (h, \rho) : h \xleftarrow{R} \mathbb{G}; \rho \leftarrow \mathsf{Expl}(\mathbb{G}, h)$$

For notational convenience, we will use $\mathsf{Samp}(\mathbb{G}^{n \times m})$ to denote sampling a uniform $n \times m$ matrix of group elements (and similarly for Expl). Now we can review the Setup, Gen algorithms of this trapdoor hash scheme. We will explicitly require that Setup samples group elements through our 'hash-to-point' function:

- $\mathsf{Setup}(1^\lambda, 1^k; \rho)$: On input security parameter λ and input length n:
  - Sample matrix $\boldsymbol{A} \xleftarrow{R} \mathsf{Samp}(\mathbb{G}^{2 \times k}; \rho)$.
  - Output $\mathsf{hk} = \boldsymbol{A}$.
- $\mathsf{Gen}(\mathsf{hk}, \mathbf{0}^n)$: On input hash key hk:
  - Sample $s \xleftarrow{R} \mathbb{Z}_p$, and set $u = g^s$.
  - Set $\boldsymbol{B} = \boldsymbol{A}^s$.
  - Output $\mathsf{ek}_0 = (u, \boldsymbol{B})$.

As before, we can show explainability by defining $\mathsf{GenSim}(\mathsf{hk}; \rho)$ as follows:

- Parse $\rho = \rho_u, \rho_{\boldsymbol{B}}$
- Sample $u \leftarrow \mathsf{Samp}(\mathbb{G}; \rho_u)$.
- Sample $\boldsymbol{B} \leftarrow \mathsf{Samp}(\mathbb{G}^{2 \times k}; \rho_{\boldsymbol{B}})$.
- Output $\mathsf{ek} = (u, \boldsymbol{B})$.

As in the LWE case, we can observe Gen outputs a DDH matrix compared to the uniform group elements of GenSim, allowing simulation soundness to follow from a straightforward reduction to DDH. To argue explainability, we can simply set

$$\mathsf{GenExp}(\mathsf{hk}, (u, \boldsymbol{B})) = \mathsf{Expl}(\mathbb{G}, u), \mathsf{Expl}(\mathbb{G}^{2 \times k}, \boldsymbol{B}) \quad , \quad \mathsf{SetupExp}(\boldsymbol{A}) = \mathsf{Expl}(\mathbb{G}, \boldsymbol{A})$$

and since Setup, GenSim simply output independent uniform group elements, explainability follow from our hash-to-point function.