Transparent Error Correcting in a Computationally Bounded World

Ofer Grossman[*] Justin Holmgren[†] Eylon Yogev[‡]

September 27, 2020

Abstract

We construct uniquely decodable codes against channels which are computationally bounded. Our construction requires only a public-coin (transparent) setup. All prior work for such channels either required a setup with secret keys and states, could not achieve unique decoding, or got worse rates (for a given bound on codeword corruptions). On the other hand, our construction relies on a strong cryptographic hash function with security properties that we only instantiate in the random oracle model.

[*] MIT, email: `ofer.grossman@gmail.edu`.

[†] NTT Research, email: `justin.holmgren@ntt-research.com`.

[‡] Tel Aviv University, email: `eylony@gmail.com`.

Contents

1 Introduction

Error correcting codes (ECCs) are a tool for handling errors when transmitting messages over an unreliable communication channel. They work by first encoding the message with additional redundant information, which is then sent over the channel. This allows the recipient to recover the original encoded message, even in the presence of a limited number of errors that might occur during transmission.

Since their introduction in the 1950s, error correcting codes [Ham50] have been a thriving research area due to their role both in practical applications and in theoretical computer science. One of the central open questions concerns the exact tradeoff between a code's *rate* (message length divided by codeword length) and the code's *error tolerance* (the number of errors that its decoding algorithm can tolerate). There are several known fundamental bounds (e.g. the Hamming, Singleton, and Plotkin bounds) on the maximum rate of a code in terms of its distance, and state of the art codes (especially over small alphabets) often only achieve significantly lower rates.

To achieve better rates, two major relaxations of error correction have been proposed. In the first, called *list decoding* [Eli57, Woz58], a decoding algorithm is no longer required to output the originally encoded message, but may instead output a short *list* of messages which is required to *contain* the original message. In this work, we will focus on standard (unique) decoding, but we will use list-decodable codes as a central building block.

In the second relaxation, the communication channel between the sender and receiver is assumed to be restricted in some way. In other words, the code is no longer required to handle fully worst-case errors. The most relevant model for us is the *computationally bounded channel* [Lip94], which loosely speaking, models codeword errors as generated by a *polynomial-time* process.

Lipton [Lip94] and Micali et al. [MPSW10] construct codes for the computationally bounded channel with better rates than are achievable by codes for worst-case errors, but their codes require a trusted setup. Specifically, the encoding algorithms for their codes (and in the case of [Lip94], also the decoding algorithm) require a secret key that, if leaked, allows an efficient channel to thwart the decoding algorithm with a relatively small number of corruptions. Secret randomness is much more difficult to instantiate than public randomness (also known as transparent), which leads us to ask:

> *Are there "good" uniquely decodable codes for the computationally bounded channel with transparent setup?*

An additional drawback of the constructions of [Lip94] and [MPSW10] is that they require a *stateful* encoder, which may render them unsuitable for use in data storage or in applications requiring concurrent transmission of multiple messages. In [Lip94], it is essential for security that the encoder's state never repeats, and essential for correctness that the decoder's state is synchronized with the encoder's state. In [MPSW10], the decoder is stateless, but it is essential for security that errors are chosen in an online fashion. In other words, there are no guarantees if a codeword c is corrupted after seeing a codeword c' that was encoded after c. This exemplifies the undesirable dependence, induced by the encoder's statefulness, of the code's error tolerance on the precise environment in which it is used. Thus we ask:

> *Are there "good" uniquely decodable codes for the computationally bounded channel with a stateless encoder?*

1.1 Our Contributions

We answer both questions affirmatively, constructing a code for computationally bounded channels (with transparent setup *and* stateless encoding) that outperforms codes for worst-case errors. As a contribution that may be of independent interest, we also construct codes with high "pseudodistance", i.e., codes for which it is hard to find two codewords that are close in Hamming distance.

Pseudounique Decoding. The main goal of an error correcting code C is to facilitate the recovery of a transmitted message given a partially corrupted copy of $C(m)$. To formalize this (in the information-theoretic setting), a polynomial-time algorithm D is said to be a *unique decoding algorithm for C against*

ρ *errors* if for all messages m and all strings c' that are ρ-close in Hamming distance to $C(m)$, we have $D(c') = m$.

In reality, messages and noise are created by nature, which can be conservatively modeled as a computationally bounded adversary. We thus relax the above *for all* quantification and only require efficient decoding when *both* m and c' are chosen by a computationally bounded process. Our codes will be described by a randomly generated seed that is used in the encoding and decoding procedures. In other words, we will work with a *seeded family* of codes $\{C_{\mathsf{pp}}\}$, where pp is the seed, which we will also refer to as the *public parameters* for the code. In our constructions, the public parameters are merely unstructured uniformly random strings of a certain length.

More formally, we say that a polynomial-time algorithm D is a *pseudounique decoding algorithm for* $\{C_{\mathsf{pp}}\}$ *against* ρ *errors* if no polynomial-time adversary A can win the following game with noticeable probability. The public parameters pp are first sampled uniformly at random and given to A. The adversary then produces a message m and a string c', and is said to win if c' is ρ-close to $C_{\mathsf{pp}}(m)$ *and* $D(\mathsf{pp}, c') \neq m$.

Under cryptographic assumptions (or in the random oracle model), we construct codes with pseudounique decoding algorithms for a larger fraction of errors than is possible in the standard setting. Our main theorem requires a "good" cryptographic hash function (which is used as a black box), where we defer the formalization of the necessary security requirements to Section 3. For now, we simply mention that it is a *multi-input* generalization of correlation intractability, and in Section 3 we show that it can be instantiated by a (non-programmable) random oracle. The precise statement and details about the construction appear in Section 4. For any $r \in (0, 1)$ and any $\rho < \min(1 - r, \frac{1}{2})$ there exist rate-r codes, over large (polynomial-sized) alphabets, that are efficiently pseudouniquely decodable against up to a ρ fraction of errors, assuming good hash functions exist (or in the random oracle model).

This should be contrasted with the Singleton bound, which rules out (standard) unique decoding for more than $\min(\frac{1-r}{2}, \frac{1}{2})$ errors. Our positive result is a corollary of a more general connection to efficient list-decodability, which we prove in Section 4. This connection also implies results over binary alphabets, albeit with bounds that are harder to state (see Section 4.4) because known binary codes do not achieve list-decoding capacity and instead have messy rate vs. error correction tradeoffs.

Pseudodistance. Our second notion is an analogue of distance. Recall that a code C is said to have distance d if for all pairs of distinct messages m_0, m_1, their encodings $C(m_0)$ and $C(m_1)$ have Hamming distance d. We can similarly replace this *for all* quantifier and only require $C_{\mathsf{pp}}(m_0)$ and $C_{\mathsf{pp}}(m_1)$ to be far for pairs m_0, m_1 that are computed from pp by a computationally bounded adversary.

We note that a code's pseudodistance may be arbitrarily high without implying anything about its decodability, even by an inefficient algorithm. It is instructive to imagine a rate-1 code whose encoding algorithm is given by a (sufficiently obfuscated) random permutation mapping $\{0,1\}^n \to \{0,1\}^n$. The pseudodistance of this code will be roughly $n/2$, but it is information theoretically impossible to decode in the presence of even a single error.

Still, pseudodistance is a useful intermediate notion for us in the construction of pseudouniquely decodable codes, and the notion may be of independent interest.

1.2 Main Definitions and Main Theorem Statement

The preceding discussion is formalized in the following definitions. A $\mathsf{seeded\ code}$ with alphabet size $q(\cdot)$ is a pair $\mathcal{C} = (\mathsf{Setup}, \mathsf{Enc})$ of polynomial-time algorithms with the following syntax:

- Setup is probabilistic, takes a domain length $k \in \mathbb{Z}^+$ (in unary), and outputs public parameters pp.
- Enc is deterministic, takes parameters pp and a message $m \in \{0,1\}^k$, and outputs a codeword $c \in [q(k)]^{n(k)}$, where $n(\cdot)$ is called the length of $\mathcal{C}$.

When $\lim_{k\to\infty} \frac{k}{n(k)\log_2 q(k)} \in [0,1]$ is well-defined it is called the rate of $\mathcal{C}$. If Setup simply outputs a uniformly random binary string of some length that depends on k, then we say that $\mathcal{C}$ is $\mathsf{public\text{-}coin}$.

A seeded code $\mathcal{C} = (\mathsf{Setup}, \mathsf{Enc})$ is said to have $\big(s(\cdot), \epsilon(\cdot)\big)$-pseudodistance $d(\cdot)$ if for all size-$s(\cdot)$ circuit ensembles $\{\mathcal{A}_k\}_{k\in\mathbb{Z}^+}$, we have

$$\Pr_{\substack{\mathsf{pp}\leftarrow\mathsf{Setup}(1^k)\\(m_0,m_1)\leftarrow\mathcal{A}_k(\mathsf{pp})}}\Big[\Delta\big(\mathsf{Enc}(\mathsf{pp}, m_0), \mathsf{Enc}(\mathsf{pp}, m_1)\big) < d\Big] \leq \epsilon(k),$$

where $\Delta(\cdot,\cdot)$ denotes the (absolute) Hamming distance.

$\mathcal{C}$ is said simply to have pseudodistance $d(\cdot)$ if for all $s(k) \leq k^{O(1)}$, there exists $\epsilon(k) \leq k^{-\omega(1)}$ such that $\mathcal{C}$ has (s, ϵ)-pseudodistance d. An algorithm Dec is said to be an $\big(s(\cdot), \epsilon(\cdot)\big)$-pseudounique decoder for $\mathcal{C} = (\mathsf{Setup}, \mathsf{Enc})$ against $d(\cdot)$ errors if for all size-$s(\cdot)$ circuit ensembles $\{\mathcal{A}_k\}_{k\in\mathbb{Z}^+}$

$$\Pr_{\substack{\mathsf{pp}\leftarrow\mathsf{Setup}(1^k)\\(m,c)\leftarrow\mathcal{A}_k(\mathsf{pp})}}\Big[\Delta\big(c, \mathsf{Enc}(\mathsf{pp}, m)\big) \leq d(k) \ \wedge\ \mathsf{Dec}(\mathsf{pp}, c) \neq m\Big] \leq \epsilon(k).$$

We say that $\mathcal{C}$ is efficiently $\big(s(\cdot), \epsilon(\cdot)\big)$-pseudouniquely decodable against $d(\cdot)$ errors if there is a *polynomial-time* algorithm Dec that is an $\big(s(\cdot), \epsilon(\cdot)\big)$-pseudounique decoder for $\mathcal{C}$. We omit s and ϵ in usage of the above definitions when for all $s(k) \leq k^{O(1)}$, there exists $\epsilon(k) \leq k^{-\omega(1)}$ such that the definition is satisfied.

We sometimes say a "ρ fraction of errors" to refer to some $d(k)$ such that $\lim_{k\to\infty} \frac{d(k)}{n(k)} = \rho$, where $n(\cdot)$ is the length of $\mathcal{C}$.

As in the previous theorem, we assume the existence of random-like hash functions to obtain our result. These hash functions can be instantiated in the random oracle model. If $\{C : \{0,1\}^k \to [q]^{n_k}\}$ is a rate-r ensemble of codes that is efficiently list-decodable against a ρ fraction of errors, and if good hash functions exist, then there exists a rate-r seeded code that is efficiently pseudouniquely decodable against a $\min\left(\rho, \frac{H_q^{-1}\left(r+H_q(\rho)\right)}{2}\right)$ fraction of errors.

The above bound has a nice interpretation when C approaches capacity, i.e. when $r + H_q(\rho) \approx 1$. Then $\frac{H_q^{-1}(r+H_q(\rho))}{2} \approx \frac{1}{2} \cdot \left(1 - \frac{1}{q}\right)$, which upper bounds the pseudo-unique decodability of any positive-rate code (implied by the proof of the Plotkin bound, and made explicit in [MPSW10]). So if C achieves capacity, Section 1.2 says that one can uniquely decode up to the (efficient) list-decoding radius of C, as long as that doesn't exceed $\frac{1}{2} \cdot \left(1 - \frac{1}{q}\right)$.

1.3 Related Work

The notion of a computationally bounded channel was first studied by Lipton [Lip94], and has subsequently been studied in a variety of coding theory settings including local decodability, local correctability, and list decoding, with channels that are bounded either in time complexity or space complexity [DGL04, GS16, BGGZ19, MPSW10, SS16, HOSW11, HO08, OPS07]. We compare some of these works in Table 1. Focusing on unique decoding against polynomial-time computationally bounded errors, the work most relevant to us is [MPSW10], improving on [Lip94].

Lipton [Lip94] showed that assuming one-way functions, any code that is (efficiently) uniquely decodable against ρ *random* errors can be upgraded to a "secret-key, stateful code" that is (efficiently) uniquely decodable against any ρ errors that are computed in polynomial time. Using known results on codes for random errors, this gives rate-r (large alphabet) codes that are uniquely decodable against a $1 - r$ fraction of errors. However, these codes require the sender and receiver to share a secret key, and to be stateful (incrementing a counter for each message sent / received).

Micali et al. [MPSW10] improve on this result, obtaining a coding scheme where only the sender needs a secret key (the receiver only needs a corresponding public key), and only the sender needs to maintain a counter. They show that these limitations are inherent in the high-error regime; namely, it is impossible to uniquely decode beyond error rates 1/4 (in the binary case) and more generally $\frac{1}{2}\cdot(1-\frac{1}{q})$ over q-ary alphabets, even if the errors are computationally bounded. Compared to [Lip94], [MPSW10] starts with codes that are efficiently list decodable, rather than codes that are uniquely decodable against random errors. The crux of their technique is using cryptographic signatures to "sieve" out all but one of the candidate messages returned by a list-decoding algorithm. Our construction also uses list decodability in a similar way. The key difference is that we use a different sieving mechanism that is stateless and transparent (i.e., the only setup is a public uniformly random string), but is only applicable for error rates below $\frac{1}{2} \cdot (1 - \frac{1}{q})$.

Our work improves over [MPSW10] in the amount of setup required for the code. In [MPSW10], the sender must initially create a secret key and share the corresponding public key with the receiver (and the adversarial channel is also allowed to depend on the public key). In contrast, our code allows anyone to send messages—no secret key is needed. This property may be useful in applications such as Wi-Fi and cellular networks, where many parties need to communicate.

Another important difference between [MPSW10] and our work is that in [MPSW10], the sender is stateful. That is, whenever the sender sends a message, he updates some internal state which affects the way the next message will be encoded. We do not make such an assumption. Note that in some situations, maintaining a state may not be possible. For example, if there are multiple senders (or a single sender who is operating several servers in different locations), it is unclear how to collectively maintain state. Whenever one of the senders sends a message, he must inform all the other senders so they can update their state accordingly, which may not be possible, or significantly slow down communication. Moreover, the guarantees of [MPSW10] only apply to adversaries that operate in a totally "online" fashion. The error tolerance guarantees break down if an adversary is able to corrupt a codeword after seeing a subsequently encoded message. In our construction, the sender and receiver are both stateless, so these issues do not arise.

One drawback of our construction compared to [MPSW10] is that our construction is not applicable in the high-error regime (error rates above 1/4 for binary codes or 1/2 for large alphabet codes). However, over large alphabets we match the performance of [MPSW10] for all error rates below 1/2.

2 Preliminaries

2.1 Combinatorics

The i^{th} falling factorial of $n \in \mathbb{R}$ is $(n)_i \overset{\mathsf{def}}{=} n \cdot (n-1) \cdots (n-i+1)$. The q-ary entropy function $H_q : [0,1] \to [0,1]$ is defined as

$$H_q(x) \overset{\mathsf{def}}{=} x \log_q(q-1) - x \log_q x - (1-x) \log_q(1-x).$$

We write $H_\infty(x)$ to denote $\lim_{q \to \infty} H_q(x)$, which is equal to x. If we write $H(x)$, omitting the subscript, we mean $H_2(x)$ by default.

For any alphabet Σ, any n, and any $u, v \in \Sigma^n$, the Hamming distance between u and v, denoted $\Delta(u,v)$, is

$$\Delta(u,v) \overset{\mathsf{def}}{=} \Big| \big\{ i \in [n] : u_i \neq v_i \big\} \Big|.$$

When $\Delta(u,v) \leq \delta n$, we write $u \approx_\delta v$. If S is a set, we write $\Delta(u,S)$ to denote $\min_{v \in S} \Delta(u,v)$.

2.2 Codes

A deterministic q-ary code is a function $C : [K] \to [q]^n$, where n is called the block length of C, $[K]$ is called the message space, and $[q]$ is called the alphabet. The distance of C is the minimum Hamming distance between $C(m)$ and $C(m')$ for distinct $m, m' \in [K]$. A probabilistic q-ary code of block length n and message space $[K]$ is a randomized function $\mathcal{C} : [K] \xrightarrow{\$} [q]^n$.

When discussing the asymptotic performance of (deterministic or probabilistic) codes, it makes sense to consider ensembles of codes $\{C_i : [K_i] \to [q_i]^{n_i}\}$ with varying message spaces, block lengths, and alphabet sizes. We will assume several restrictions on K_i, n_i, and q_i that rule out various pathologies. Specifically, we will assume that:

- K_i, q_i, and n_i increase weakly monotonically with i and are computable from i in polynomial time (i.e. in time $\operatorname{polylog}(i)$).

work	**setup**	**noise**	**decoding**	**rate**	**notes/assumptions**
This paper	URS	$\mathsf{P/poly}$	unique	arbitrarily close to $1-p$ for large alphabets	two-input correlation intractablility
[GS16]	URS	$\mathsf{SIZE}(n^c)$	list	arbitrarily close to $1-H(p)$	no assumptions
[SS16]	none	$\mathsf{SIZE}(n^c)$	list	arbitrarily close to $1-H(p)$	PRGs for small circuits
[SS20]	none	$\mathsf{SPACE}(n^\delta)$	unique	arbitrarily close to $1-H(p)$	none
[MPSW10]	public key	$\mathsf{P/poly}$	unique	matches list decoding radius	stateful sender and one-way functions
[OPS07]	private shared randomness	$\mathsf{P/poly}$	local	$\Omega(1)$ (for error rate $\Omega(1)$)	one-way functions
[HOSW11]	public key	$\mathsf{P/poly}$	local	$\Omega(1)$ (for error rate $\Omega(1)$)	public-key encryption
[BGGZ19]	URS	$\mathsf{P/poly}$	local correction	$\Omega(1)$ (for error rate $\Omega(1)$)	collision-resistant hash function
[Lip94]	private shared randomness	$\mathsf{P/poly}$	unique	matches BSC channel	stateful sender and one-way functions

Table 1: Summary of related work. The column "message" refers to how the message are generated. The column "noise" describes the computational power of the adversary adding noise. URS stands for uniform random string (shared publicly between the sender, receiver, and adversary), BSC for binary symmetric channel, and PRG for pseudorandom generator.

- q_i is at most $\mathrm{polylog}(K_i)$.
- There is a polynomial-time algorithm E that given (i, x) for $x \in [K_i]$ outputs $C_i(x)$.
- The limit $r = \lim_{i\to\infty} \frac{\log K_i}{n_i \cdot \log q_i}$ exists with $r \in (0, 1)$. We call r the rate of the ensemble.
- $\limsup_{i\to\infty} \frac{\log K_{i+1}}{\log K_i} = 1$. This is important so that the cost of padding (to encode arbitrary-length messages) is insignificant.

One implication of these restrictions is that without loss of generality we can assume that $\{K_i\}_{i\in\mathbb{Z}^+} = \left\{2^k\right\}_{k\in\mathbb{Z}^+}$ and we can index our codes by k rather than by i. We say that an ensemble of codes $\left\{C_k : \{0,1\}^k \to [q_k]^{n_k}\right\}_{k\in\mathbb{Z}^+}$ is combinatorially ρ-list decodable if for any $y \in [q_k]^{n_k}$, there are at most $\mathrm{poly}(k)$ values of $m \in \{0,1\}^k$ for which $C_k(m) \approx_\rho y$. If there is a polynomial-time algorithm that outputs all such m given y (and 1^k), we say that $\{C_k\}$ is *efficiently* ρ-list decodable.

2.3 Pseudorandomness

Random variables $X_1, \ldots, X_n$ are said to be t-wise independent if for any set $S \subseteq [n]$ with size $|S| = t$, the random variables $\{X_i\}_{i\in S}$ are mutually independent.

Discrete random variables $X_1, \dots, X_n$ are said to be t-wise β-dependent in Rényi∞-divergence if for all sets $S \subseteq [n]$ of size $|S| = t$, it holds for all $(x_i)_{i \in S}$ that

$$\Pr\left[\bigwedge_{i \in S} X_i = x_i\right] \leq \beta \cdot \prod_{i \in S} \Pr[X_i = x_i].$$

2.3.1 Permutations

If X is a finite set, we write S_X to denote the set of all permutations of X. A family of permutations $\Pi \subseteq S_X$ is said to be t-wise ϵ-dependent if for all distinct $x_1, \dots, x_t \in X$, the distribution of $\big(\pi(x_1), \dots, \pi(x_t)\big)$ for uniformly random $\pi \leftarrow \Pi$ is ϵ-close in statistical distance to uniform on $\{(y_1, \dots, y_t) : y_1, \dots, y_t \text{ are distinct.}\}$

To avoid pathological issues regarding the domains of permutation families (e.g. their sampleability, decidability, and compressability), we will restrict our attention to permutations on sets of the form $\{0,1\}^k$ for $k \in \mathbb{Z}^+$.

We say that an ensemble $\{\Pi_k \subseteq S_{\{0,1\}^k}\}_{k \in \mathbb{Z}^+}$ of permutation families is fully explicit if there are poly(k)-time algorithms for:

- sampling a description of $\pi \leftarrow \Pi_k$; and
- computing $\pi(x)$ and $\pi^{-1}(x)$ given x and a description of $\pi \in \Pi_k$.

[[KNR09]] For any $t = t(k) \leq k^{O(1)}$, and any $\epsilon = \epsilon(k) \geq 2^{-k^{O(1)}}$, there is a fully explicit t-wise ϵ-dependent ensemble $\{\Pi_k \subseteq S_{\{0,1\}^k}\}_{k \in \mathbb{Z}^+}$ of permutation families.

The following non-standard variation on the notion of t-wise almost-independence will prove to be more convenient for us. A probability distribution P is said to be β-close in Rényi∞-divergence to a distribution Q if for all x, $P(x) \leq \beta \cdot Q(x)$. We say that a family $\Pi \subseteq S_X$ is t-wise β-dependent in Rényi∞-divergence if for all distinct $x_1, \dots, x_t \in X$, the distribution of $\big(\pi(x_1), \dots, \pi(x_t)\big)$ is β-close in Rényi∞-divergence to the uniform distribution on X^t.

It is easily verified that any family of permutations $\Pi \subseteq S_{[K]}$ that is t-wise ϵ-dependent as in Section 2.3.1 is also t-wise β-dependent in Rényi∞-divergence with $\beta = \epsilon \cdot K^t + \frac{K^t}{(K)_t}$. Thus Section 2.3.1 gives us the following. For any $t = t(k) \leq k^{O(1)}$, there is a fully explicit t-wise $O(1)$-dependent (in Rényi∞-divergence) ensemble $\{\Pi_k \subseteq S_{\{0,1\}^k}\}_{k \in \mathbb{Z}^+}$ of permutation families.

3 Multi-input Correlation Intractability

Correlation intractability was introduced by Canetti Goldreich and Halevi [CGH04] as a way to model a large class of random oracle-like security properties of hash functions. Roughly speaking, H is said to be correlation intractable if for any sparse relation R it is hard to find x such that $(x, H(x)) \in R$. In recent years, CI hash functions have been under the spotlight with surprising results on instantiating CI hash families from concrete computational assumptions (e.g., [CCR16, KRR17, CCRR18, CCH$^+$18, PS19]).

In this work, we need a stronger *multi-input* variant of correlation intractability. We formulate a notion of multi-input sparsity such that a hash function can plausibly be correlation intractable for all sparse multi-input relations. Indeed, we prove that a random oracle has this property.

[Multi-Input Relations] For sets $\mathcal{X}$ and $\mathcal{Y}$, an ℓ-input relation on $(\mathcal{X}, \mathcal{Y})$ is a subset $R \subseteq \mathcal{X}^\ell \times \mathcal{Y}^\ell$.

We say that R is p-sparse if for all $i \in [\ell]$, all distinct $x_1, \dots, x_\ell \in \mathcal{X}$, and all $y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_\ell \in \mathcal{Y}$, we have

$$\Pr_{y_i \leftarrow \mathcal{Y}}[(x_1, \dots, x_\ell, y_1, \dots, y_\ell) \in R] \leq p.$$

An ensemble of ℓ-input relations $\{R_\lambda\}_{\lambda \in \mathbb{Z}^+}$ is said simply to be sparse if there is a negligible function $p \colon \mathbb{Z}^+ \to \mathbb{R}$ such that each R_λ is $p(\lambda)$-sparse. A natural but flawed generalization of single-input sparsity for an ℓ-input relation R might instead require that for all $x_1, \ldots, x_\ell$, it holds with overwhelming probability over a uniform choice of $y_1, \ldots, y_\ell$ that $(x_1, \ldots, x_\ell, y_1, \ldots, y_\ell) \notin R$. Unfortunately this definition does not account for an adversary's ability to choose some x_i adaptively. Indeed, even a random oracle would not be 2-input correlation intractable under this definition for the relation $\{(x_1, x_2, y_1, y_2) : x_2 = y_1\}$, which does satisfy the aforementioned "sparsity" property.

[Multi-Input Correlation Intractability] An ensemble $\mathcal{H} = \{\mathcal{H}_\lambda\}_{\lambda \in \mathbb{Z}^+}$ of function families $\mathcal{H}_\lambda = \{H_k : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda\}_{k \in \mathcal{K}_\lambda}$ is ℓ-input $\big(s(\cdot), \epsilon(\cdot)\big)$-correlation intractable for a relation ensemble $\{R_\lambda \subseteq \mathcal{X}_\lambda^\ell \times \mathcal{Y}_\lambda^\ell\}$ if for every size-$s(\lambda)$ adversary $\mathcal{A}$:

$$\Pr_{\substack{k \leftarrow \mathcal{K}_\lambda \\ (x_1, \ldots, x_\ell) \leftarrow \mathcal{A}(k)}} \Big[\big(x_1, \ldots, x_\ell, H_k(x_1), \ldots, H_k(x_\ell)\big) \in R_\lambda\Big] \leq \epsilon(\lambda).$$

3.1 Multi-Input Correlation Intractability of Random Oracles

We show that a random oracle is ℓ-input correlation intractable as in Section 3. Let F be a uniformly random function mapping $\mathcal{X} \to \mathcal{Y}$, and let $\ell \in \mathbb{Z}^+$ be a constant. Then, for any p-sparse ℓ-distinct-input relation R on $(\mathcal{X}, \mathcal{Y})$, and any T-query oracle algorithm $\mathcal{A}^{(\cdot)}$, we have

$$\Pr\left[\mathcal{A}^F \text{ outputs } (x_1, \ldots, x_\ell) \in \mathcal{X}^\ell \text{ s.t. } \big(x_1, \ldots, x_\ell, F(x_1), \ldots, F(x_\ell)\big) \in R\right] \leq p \cdot (T)_\ell \leq p \cdot T^\ell.$$

Proof overview. We give an overview of the proof which should give some intuition as to why get the expression $p \cdot T^\ell$. Fix a set of elements $x_1, \ldots, x_\ell$ then the probability, over the random oracle, that these elements will be in the relation with respect with the random oracle is at most p, which follows from the definition of sparsity. However, for a longer list of elements of length, we would need to take into account all the possible tuples of size ℓ in that list, and apply a union bound. Since the number of queries is bounded by T, we get that the probability is at most $p \cdot T^\ell$.

The above arguments work for a *fixed* list of elements, and gives intuition for the probability expression achieved in the theorem. However, an oracle algorithm is allowed to perform *adaptive* queries where the next query might depend on the result of the random oracle for previous queries. This makes the proof more challenging and, in particular, much more technical.

Proof. We begin the proof by stating a few assumptions about the algorithm $\mathcal{A}$, and observe that these assumption hold without loss of generality:

- $\mathcal{A}$ is deterministic;
- $\mathcal{A}$ never makes repeated queries to F nor does $\mathcal{A}$ output non-distinct $x_1, \ldots, x_\ell$; and
- If at any point $\mathcal{A}$ has made queries $q_1, \ldots, q_j$ and received answers $a_1, \ldots, a_j$ such that for some $i_1, \ldots, i_k \in [j]^\ell$ the tuple $(q_{i_1}, \ldots, q_{i_\ell}, a_{i_1}, \ldots, a_{i_\ell})$ is in R, then $\mathcal{A}$ immediately outputs one such tuple (without making any further queries).

We denote the random variables representing the various queries of the algorithm $\mathcal{A}$, and their responses from the oracle. Let M be a random variable denoting the number of queries made by $\mathcal{A}$, let $Q_1, \ldots, Q_M$ denote the queries made by $\mathcal{A}$ to F, let $A_1, \ldots, A_M$ denote the corresponding evaluations of F, let $\mathcal{Q}$ denote $\{Q_1, \ldots, Q_M\}$, let $X_1, \ldots, X_\ell$ denote the output of $\mathcal{A}^F$, and let $Y_1, \ldots, Y_\ell$ denote the corresponding evaluations of F.

We split our analysis into two cases: either $(X_1, \ldots, X_\ell) \in \mathcal{Q}^k$, or not, meaning that the algorithm did not query all of the ℓ elements it outputs. We argue about each case separately and at the end combine to the two to a single argument. We begin with the second case. For any algorithm $\mathcal{A}$ it holds that

$$\Pr\left[(X_1, \ldots, X_\ell, Y_1, \ldots, Y_\ell) \in R \;\mid\; (X_1, \ldots, X_\ell) \notin \mathcal{Q}^\ell\right] \leq p \cdot \ell \;.$$

where the probability is over the random oracle.

Proof sketch. There exists some component of $\mathcal{A}$'s output whose image under F is independent of $\mathcal{A}$'s view, and thus is uniformly random. Since R is p-sparse, this ensures that $(X_1, \ldots, X_\ell, Y_1, \ldots, Y_\ell)$ is in R with probability at most p.

Proof. Fix any $i \in [\ell]$ and any $(q_1, \ldots, q_\ell, a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_\ell)$. Since the relation is p-sparse, we know that

$$\Pr[(q_1, \ldots, q_\ell, a_1, \ldots, a_{i-1}, Y_i, a_{i+1}, \ldots, a_\ell) \in R] \leq p \ .$$

Thus, we can write:

$$\begin{aligned}
&\Pr\left[(X_1, \ldots, X_\ell, Y_1, \ldots, Y_\ell) \in R \ \mid \ (X_1, \ldots, X_\ell) \notin \mathcal{Q}^\ell\right] \\
&\quad \leq \sum_{i \in [\ell]} \Pr\left[(X_1, \ldots, X_\ell, Y_1, \ldots, Y_\ell) \in R \ \mid \ X_i \in \mathcal{Q}\right] \cdot \Pr[X_i \notin \mathcal{Q}] \\
&\quad \leq \sum_{i \in [\ell]} \sum_{\substack{q_1, \ldots, q_\ell \\ a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_\ell}} \Pr\left[(q_1, \ldots, q_\ell, a_1, \ldots, a_{i-1}, Y_i, a_{i+1}, \ldots, a_\ell) \in R\right] \cdot \\
&\qquad \Pr[\forall j \neq i : X_j = q_j, Y_j = a_j, X_i = a_i] \cdot \Pr[X_i \notin \mathcal{Q}] \\
&\quad \leq p \cdot \sum_{i \in [\ell]} \Pr[X_i \notin \mathcal{Q}] \cdot \sum_{\substack{q_1, \ldots, q_\ell \\ a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_\ell}} \Pr[\forall j \neq i : X_j = q_j, Y_j = a_j, X_i = a_i] \\
&\quad \leq p \cdot \sum_{i \in [\ell]} \Pr[X_i \notin \mathcal{Q}] \leq p \cdot \ell \ .
\end{aligned}$$

□

We turn to prove the first case, where all the elements in the algorithm's output where queried. This case is where we pay the $p \cdot T^k$ in the probability. For any T-query algorithm $\mathcal{A}$ it holds that:

$$\Pr\left[(X_1, \ldots, X_\ell, Y_1, \ldots, Y_\ell) \in R \ \mid \ (X_1, \ldots, X_\ell) \in \mathcal{Q}^\ell\right] \leq p \cdot T^k.$$

Proof. For any $m \in [T]$, let Z_m be an indicator random variable to the event that the m^{th} query of the algorithm $\mathcal{A}$ along with some $\ell - 1$ previous queries form an instance in the relation. Formally, we define:

$$Z_m = \begin{cases} 1 & \text{if } \exists i_1, \ldots, i_\ell \in [m] \text{ such that } (Q_{i_1}, \ldots, Q_{i_\ell}, A_{i_1}, \ldots, A_{i_{\ell-1}}, A_m) \in R \text{ and } m \in \{i_1, \ldots, i_\ell\} \\ 0 & \text{otherwise} \end{cases} .$$

Observe that using this notation, we have that if the event $(X_1, \ldots, X_\ell, Y_1, \ldots, Y_\ell) \in R$ implies that there exist an $m \in [T]$ such that $Z_m = 1$. Using the fact that R is p-sparse, we bound $\Pr[Z_m = 1]$, for any $m \in [T]$ as follows:

$$\begin{aligned}
\Pr[Z_m] = 1 &= \Pr[\exists i_1, \ldots, i_\ell \in [m] \text{ such that } (Q_{i_1}, \ldots, Q_{i_\ell}, A_{i_1}, \ldots, A_{i_\ell}) \in R \text{ and } m \in \{i_1, \ldots, i_\ell\}] \\
&\leq \sum_{i_1, \ldots, i_\ell \in [m],\ m \in \{i_1, \ldots, i_\ell\}} \Pr[(Q_{i_1}, \ldots, Q_{i_\ell}, A_{i_1}, \ldots, A_{i_\ell}) \in R] \\
&\leq \sum_{i_1, \ldots, i_\ell \in [m],\ m \in \{i_1, \ldots, i_\ell\}} p \leq p \cdot \ell \cdot (m-1)_{\ell-1} \ .
\end{aligned}$$

Then, we union bound over all z_m for $m \in [T]$ and get that

$$\begin{aligned}
&\Pr\left[(X_1, \ldots, X_\ell, Y_1, \ldots, Y_\ell) \in R \ \mid \ (X_1, \ldots, X_\ell) \in \mathcal{Q}^\ell\right] \leq \Pr[\exists m \in [T] : Z_m = 1] \\
&\quad \leq \sum_{m=1}^{T} \Pr[Z_m = 1] \leq \sum_{m=1}^{T} p \cdot \ell \cdot (m-1)_{\ell-1} \leq p \cdot (T)_\ell \ .
\end{aligned}$$

□

Finally, using the two claims we get that

$$\begin{aligned}
&\Pr\left[(X_1,\ldots,X_\ell,Y_1,\ldots,Y_\ell)\in R\right] \\
&\quad= \Pr\left[(X_1,\ldots,X_\ell,Y_1,\ldots,Y_\ell)\in R \;\middle|\; (X_1,\ldots,X_\ell)\in\mathcal{Q}^\ell\right]\cdot\Pr[(X_1,\ldots,X_\ell)\in\mathcal{Q}^\ell] \\
&\quad+ \Pr\left[(X_1,\ldots,X_\ell,Y_1,\ldots,Y_\ell)\in R \;\middle|\; (X_1,\ldots,X_\ell)\notin\mathcal{Q}^\ell\right]\cdot\Pr[(X_1,\ldots,X_\ell)\notin\mathcal{Q}^\ell] \\
&\quad\le p\cdot(T)_\ell\cdot\Pr[(X_1,\ldots,X_\ell)\in\mathcal{Q}^\ell] + p\cdot\ell\cdot\Pr[(X_1,\ldots,X_\ell)\notin\mathcal{Q}^\ell] \\
&\quad\le p\cdot(T)_\ell\cdot\Pr[(X_1,\ldots,X_\ell)\in\mathcal{Q}^\ell] + p\cdot(T)_\ell\cdot\Pr[(X_1,\ldots,X_\ell)\notin\mathcal{Q}^\ell] \\
&\quad= p\cdot(T)_\ell \le p\cdot T^\ell \;\;.
\end{aligned}$$

□

4 Our Construction

We have defined the notion of a multi-input correlation intractable hash, and showed that they can be constructed in the random oracle model. We now construct a seeded family of codes that is pseudouniquely decodable against a large fraction of errors, using 2-input correlation intractable hash functions as a central tool (in a black-box way). Loosely speaking, our construction starts with any efficiently list-decodable code $\mathcal{C}\colon\{0,1\}^k\to[q]^n$, and modifies it in several steps.

1. We first apply a decodability- and rate-preserving seeded transformation to $\mathcal{C}$ to obtain (a seeded family of) *stochastic* codes in which with all pairs of messages are mapped to far apart codewords with overwhelmingly probability.

   Specifically, the seed is (loosely speaking) a pseudorandom permutation $\pi\colon\{0,1\}^k\to\{0,1\}^k$, and the stochastic code maps $m'\in\{0,1\}^{k-\ell}$ to $\mathcal{C}\Big(\pi\big(m'\|r\big)\Big)$ for uniformly random $r\leftarrow\{0,1\}^\ell$, where ℓ satisfies $\omega(k)\le\ell\le o(k)$.

2. We derandomize these codes by generating randomness deterministically as a hash of the message.

More formally, we will consider the following parameterized construction of a seeded code family. Suppose that

- $\mathcal{C}=\{C_k:\{0,1\}^k\to[q_k]^{n_k}\}_{k\in\mathbb{Z}^+}$ is a fully explicit ensemble of codes,
- $\Pi=\{\Pi_k\subseteq S_{\{0,1\}^k}\}_{k\in\mathbb{Z}^+}$ is a fully explicit ensemble of permutation families, and
- $\mathcal{H}=\{\mathcal{H}_k\}$ is a fully explicit ensemble of hash function families, where functions in $\mathcal{H}_k$ map $\{0,1\}^{k-\ell_k}$ to $\{0,1\}^{\ell_k}$ for some $\ell=\ell_k$ satisfying $\omega(\log k)\le\ell_k\le o(k)$.

Then we define a seeded family of codes $\mathcal{SC}[\mathcal{C},\Pi,\mathcal{H}]$ by the following algorithms $(\mathsf{Setup},\mathsf{Enc})$:

- Setup takes 1^k as input, samples $\pi\leftarrow\Pi_k$ and $h\leftarrow\mathcal{H}_k$, and outputs (π,h).
- Enc takes (π,h) as input, as well as a message $m\in\{0,1\}^{k-\ell}$, and outputs $C_k\Big(\pi\big(m,h(m)\big)\Big)$.

$\mathcal{SC}[\mathcal{C},\Pi,\mathcal{H}]$ inherits several basic properties from $\mathcal{C}$, including alphabet size and block length. We only consider hash family ensembles $\{\mathcal{H}_k\}$ in which the output length ℓ_k of functions in $\mathcal{H}_k$ satisfies $\ell_k\le o(k)$. With such parameters, the resulting coding scheme $\mathcal{SC}[\mathcal{C},\Pi,\mathcal{H}]$ has the same rate as $\mathcal{C}$.

4.1 From 2-Input Correlation Intractability to Pseudodistance

In this section, we show that if $\mathcal{C}$ is a sufficiently good ensemble of codes, $\mathcal{H}$ is a two-input correlation intractable hash with an appropriate output length, and Π is pseudorandom, then $\mathcal{SC}[\mathcal{C},\Pi,\mathcal{H}]$ has high pseudodistance.

For any:

- rate-r (combinatorially) ρ-list decodable ensemble of codes $\{C_k : \{0,1\}^k \to [q_k]^{n_k}\}_{k\in\mathbb{Z}^+}$;
- ensemble $\Pi = \{\Pi_k \subseteq S_{\{0,1\}^k}\}_{k\in\mathbb{Z}^+}$ of $\omega(1)$-wise $O(1)$-dependent (in Rényi∞-divergence) permutation families;
- $\delta \in (0,1)$ satisfying $H_q(\delta) - H_q(\rho) < r$, where $q = \lim_{k\to\infty} q_k$

$\mathcal{SC}[\mathcal{C}, \Pi, \mathcal{H}]$ has relative pseudodistance δ as long as $\mathcal{H}$ is 2-input correlation intractable for a specific family of sparse relations.

Proof. By construction, $\mathcal{SC}[\mathcal{C}, \Pi, \mathcal{H}]$ has relative pseudodistance δ if and only if given $\pi \leftarrow \Pi_k$ and $h \leftarrow \mathcal{H}_k$, it is hard to find $m_0, m_1 \in \{0,1\}^{k-\ell_k}$ such that $C_k\Big(\pi\big(m_0, h(m_0)\big)\Big) \approx_\delta C_k\Big(\pi\big(m_1, h(m_1)\big)\Big)$, i.e. if $\big(m_0, m_1, h(m_0), h(m_1)\big)$ is in the relation:

$$\mathcal{R}^{\mathsf{close}}_{\mathcal{C},\pi,\delta,\ell_k} \subseteq \left(\{0,1\}^{k-\ell_k}\right)^2 \times \left(\{0,1\}^{\ell_k}\right)^2$$
$$\mathcal{R}^{\mathsf{close}}_{\mathcal{C},\pi,\delta,\ell_k} \overset{\mathsf{def}}{=} \Big\{(m_0, m_1, r_0, r_1) : C_k\Big(\pi\big(m_0, r_0\big)\Big) \approx_\delta C_k\Big(\pi\big(m_1, r_1\big)\Big)\Big\}.$$

To finish the proof of Section 4.1, it suffices to show that this relation is sparse with high probability (over the choice of $\pi \leftarrow \Pi_k$), which is established by the following claim. For any:

- rate-r combinatorially ρ-list decodable ensemble of codes $\{C_k : \{0,1\}^k \to [q_k]^{n_k}\}_{k\in\mathbb{Z}^+}$;
- $\delta \in (0,1)$ satisfying $\lim_k \big(H_{q_k}(\delta) - H_{q_k}(\rho)\big) < r$;

for $t_k \geq \omega(1)$ and all t_k-wise $O(1)$-dependent (in Rényi∞-divergence) permutation families $\{\Pi_k \subseteq S_{\{0,1\}^k}\}$ and all $\ell_k \leq o(k)$, it holds for random $\pi \leftarrow \Pi_k$ that the relation $\mathcal{R}^{\mathsf{close}}_{C_k,\pi,\delta,\ell_k}$ is $t_k \cdot 2^{-\ell_k}$-sparse with all but $2^{-\Omega(k\cdot t_k)}$ probability.

Proof of Section 4.1. For simplicity of presentation, we omit the explicit dependencies of C_k, Π_k, q_k, n_k, t_k, and ℓ_k on k, simply writing C, q, n, t, and ℓ respectively in statements that are to be interpreted as holding for all sufficiently large k.

Fix any $(x_1, y_1) \in \{0,1\}^{k-\ell} \times \{0,1\}^\ell$ and consider the Hamming ball $B \subseteq [q]^n$ of relative radius δ around $C(x_1, y_1)$. Using Appendix B, we get that B can be covered by $q^{n\cdot(H_q(\delta)-H_q(\rho))} \cdot \mathrm{poly}(n)$ balls of relative radius ρ. By C's combinatorial ρ-list decodability, each such ball contains at most $\mathrm{poly}(k)$ codewords of C. The total number of codewords in C is at most $2^k \approx q^{rn}$ which lets us write:

$$\Pr_{c\leftarrow C}\left[c \approx_\delta C(x_1, y_1)\right] \leq \mathrm{poly}(k) \cdot \mathrm{poly}(n) \cdot q^{n\cdot\big(H_q(\delta)-H_q(\rho)\big)} \cdot q^{-nr} \leq q^{-\Omega(n)} \leq 2^{-\Omega(k)}.$$

Now, observe that as long as $t \geq 2$, by the t-wise $O(1)$-dependence of Π, there exists a constant c such that for any x_1, x_2, y_1, y_2 with $x_1 \neq x_2$ it holds that:

$$\Pr_\pi[(x_1, x_2, y_1, y_2) \in \mathcal{R}^{\mathsf{close}}_{\mathcal{C},\pi,\delta,\ell}] \leq c \cdot \Pr_{c\leftarrow C}\left[c \approx_\delta C(x_1, y_1)\right] \leq 2^{-\Omega(k)} \quad .$$

Thus, the expected number μ of y_2' for which $(x_1, x_2, y_1, y_2') \in \mathcal{R}^{\mathsf{close}}_{\mathcal{C},\pi,\delta,\ell}$ satisfies $\mu \leq 2^{\ell-\Omega(k)}$. Applying a concentration bound for t-wise almost-dependent random variables (Appendix A), we see that for any fixed x_1, x_2, y_1 with $x_1 \neq x_2$ it holds that

$$\Pr_\pi\left[\Pr_{y_2\leftarrow\{0,1\}^\ell}\left[(x_1, x_2, y_1, y_2) \in \mathcal{R}^{\mathsf{close}}_{\mathcal{C},\pi,\delta,\ell}\right] \geq \frac{t+1}{2^\ell}\right] \leq O\left(\frac{\mu^t}{(t+1)!}\right) \leq O\left(\mu^t\right).$$

Thus, by a union bound over x_1, x_2, y_1, it holds that, with all but $O\big(2^{2k-\ell} \cdot \mu^t\big)$ probability, for *all* x_1, x_2, y_1,

$$\Pr_{y_2\leftarrow\{0,1\}^\ell}\left[(x_1, x_2, y_1, y_2) \in \mathcal{R}^{\mathsf{close}}_{\mathcal{C},\pi,\delta,\ell}\right] \leq \frac{t}{2^\ell}. \tag{1}$$

By a symmetric argument, it holds with all but $O\big(2^{2k-\ell} \cdot \mu^t\big)$ probability that for all x_1, x_2, y_2,

$$\Pr_{y_1 \leftarrow \{0,1\}^\ell}\left[(x_1, x_2, y_1, y_2) \in \mathcal{R}^{\mathsf{close}}_{\mathcal{C},\pi,\delta,\ell}\right] \leq \frac{t}{2^\ell}. \tag{2}$$

Applying one last union bound, Eqs. (1) and (2) hold simultaneously with probability all but

$$\begin{aligned} O\left(2^{2k-\ell} \cdot \mu^t\right) &\leq 2^{2k-\ell} \cdot 2^{t\big(\ell - \Omega(k)\big)} \\ &\leq 2^{-\Omega(tk)}, \end{aligned}$$

where the last inequality is because $\ell \leq o(k)$ and $t \geq \omega(1)$. □

This concludes the proof of Section 4.1 □

4.2 From Efficient List Decodability to Pseudounique Decodability

We next observe that if $\mathcal{C}$ is *efficiently ρ-list decodable* then so is $\mathcal{C}' = \mathcal{SC}[\mathcal{C}, \Pi, \mathcal{H}]$ (as long as Π and $\mathcal{H}$ are fully explicit). We show that this, combined with the high pseudodistance that we have already established, implies that $\mathcal{C}'$ has a pseudounique decoding algorithm against a large fraction of errors.

We first define the straight-forward adaptation of list decoding for seeded families of codes. We say that Dec is an $\big(L(\cdot), \rho\big)$-list decoding algorithm for a seeded family of codes $(\mathsf{Setup}, \mathsf{Enc})$ if for all pp in the support of $\mathsf{Setup}(1^k)$, all $m \in \{0,1\}^k$, and all $y \approx_\rho \mathsf{Enc}(\mathsf{pp}, m)$, $\mathsf{Dec}(\mathsf{pp}, y)$ is an $L(k)$-sized set that contains m. We say that Dec is simply a ρ-list decoding algorithm if it is an $\big(L(\cdot), \rho\big)$-list decoding algorithm for some $L(k) \leq k^{O(1)}$.

We say that $\mathcal{C} = (\mathsf{Setup}, \mathsf{Enc})$ is efficiently ρ-list decodable if there exists a polynomial-time ρ-list decoding algorithm for $\mathcal{C}$.

If $\mathcal{C} = \{C_k\}$ is *efficiently ρ-list decodable* and Π and $\mathcal{H}$ are fully explicit, then so is $\mathcal{SC}[\mathcal{C}, \Pi, \mathcal{H}]$.

Proof. Given public parameters $(\pi, h) \leftarrow \mathsf{Setup}(1^k)$ and a noisy codeword c', we can list-decode by:

1. Running the list-decoding algorithm for C_k to obtain strings $y_1, \ldots, y_L \in \{0,1\}^k$,
2. Inverting each y_i under π to obtain pairs $(m_1, r_1), \ldots, (m_L, r_L)$,
3. Outputting the set $\{m_i : r_i = h(m_i) \wedge C_k(\pi(m_i, r_i)) \approx_\rho c'\}$. □

If $\mathcal{C} = (\mathsf{Setup}, \mathsf{Enc})$ is a seeded family of codes that:

- is efficiently list-decodable against a ρ fraction of errors; and
- has relative pseudodistance $\tilde{\delta}$,

then $\mathcal{C}$ is efficiently pseudouniquely decodable against a ρ' fraction of errors for any $\rho' < \min(\rho, \frac{\tilde{\delta}}{2})$.

Proof. Let $q = q(k)$ and $n = n(k)$ denote the alphabet and block length of $\mathcal{C}$, respectively. The efficient pseudounique decoding algorithm Dec operates as follows, given public parameters pp and corrupted codeword $y \in [q]^n$ as input:

1. Run the list-decoding algorithm for $\mathcal{C}$ on (pp, y) to obtain a list of messages $m_1, \ldots, m_L$ (and corresponding codewords $c_1, \ldots, c_L$).
2. Output m_i for the $i \in [L]$ minimizing $\Delta(c_i, y)$.

This algorithm clearly runs in polynomial-time, so it suffices to analyze correctness. Suppose we have $(m, y) \leftarrow \mathcal{A}(\mathsf{pp})$, where $\mathcal{A}$ is a polynomial-size adversary and $\Delta\big(y, \mathsf{Enc}(\mathsf{pp}, m)\big) \leq \rho' n$. We first observe that some $m_i = m$ by the list-decodability of $\mathcal{C}$. No other m_j can also have $\Delta\big(y, \mathsf{Enc}(\mathsf{pp}, m)\big) \leq \rho' n$, because otherwise we would have $\Delta(m_i, m_j) \leq 2\rho' n < \tilde{\delta} n$ by the triangle inequality. This contradicts the $\mathcal{C}$'s pseudodistance since the above process for generating $\{m_1, \ldots, m_L\}$ is efficient.

In other words, c_i is the closest codeword to y, and the decoding algorithm outputs $m_i = m$ as desired. □

4.3 Main Theorem

We are now ready to state our main theorem: For any:

- rate-r (efficiently) ρ-list decodable fully explicit ensemble $\mathcal{C}$ of codes $\{C_k : \{0,1\}^k \to [q_k]^{n_k}\}_{k\in\mathbb{Z}^+}$;
- ensemble $\Pi = \{\Pi_k \subseteq S_{\{0,1\}^k}\}_{k\in\mathbb{Z}^+}$ of $\omega(1)$-wise $O(1)$-dependent (in Rényi∞-divergence) permutation families;
- ensemble $\mathcal{H} = \{\mathcal{H}_k\}$ of 2-input correlation intractable hash families, where functions in $\mathcal{H}_k$ map $\{0,1\}^k$ to $\{0,1\}^{k-\ell_k}$ for $\omega(\log k) \le \ell_k \le o(k)$;
- $\rho' < \min\left(\rho, \frac{H_q^{-1}\left(r+H_q(\rho)\right)}{2}\right)$ where $q = \lim_{k\to\infty} q_k$,

$\mathcal{SC}[\mathcal{C},\Pi,\mathcal{H}]$ is efficiently pseudouniquely decodable against a ρ' fraction of errors.

Proof. Follows immediately by combining Sections 4.1 to 4.2. □

4.4 Instantiations with Known Codes

Finally, we apply Section 4.3 with some known codes, first recalling applicable results from coding theory. We focus on large alphabets ($q_k \to \infty$) and binary alphabets ($q_k = 2$). [[GR08]] For all $r, \rho \in (0,1)$ satisfying $r + \rho < 1$, there is a rate-r, efficiently ρ-list decodable, fully explicit ensemble of codes $\{C_k : \{0,1\}^k \to [q_k]^{n_k}\}_{k\in\mathbb{Z}^+}$ with $q_k \le \text{poly}(k)$.

[[GR09]] For all r, ρ satisfying $0 < \rho < 1/2$ and

$$0 < r < R_{\mathsf{BZ}}(\rho) \overset{\mathsf{def}}{=} 1 - H(\rho) - \rho \cdot \int_0^{1-H(\rho)} \frac{dx}{H^{-1}(1-x)}, \tag{3}$$

there is a rate-r, efficiently ρ-list decodable, fully explicit ensemble of codes $\{C_k : \{0,1\}^k \to \{0,1\}^{n_k}\}_{k\in\mathbb{Z}^+}$. The bound of Eq. (3) is called the Blokh-Zyablov bound.

Plugging these codes into Section 4.3, we get For all r, ρ with $r + \rho < 1$, there is a rate-r seeded family of codes (with alphabet size $q_k \le \text{poly}(k)$), that is efficiently pseudouniquely decodable against a ρ fraction of errors. This result should be contrasted with the Singleton bound, which states that if rate-r code is uniquely decodable against a ρ fraction of errors, then $r + 2\rho \le 1$. For all $0 < \rho < 1/2$ and all $0 < r < R_{\mathsf{BZ}}(\rho)$, there is a rate-$r$ seeded family of binary codes that is efficiently pseudouniquely decodable against a $\min\left(\rho, \frac{H^{-1}\left(r+H(\rho)\right)}{2}\right)$ fraction of errors.

Acknowledgments

This work was done (in part) while the authors were visiting the Simons Institute for the Theory of Computing. Eylon Yogev is funded by the ISF grants 484/18, 1789/19, Len Blavatnik and the Blavatnik Foundation, and The Blavatnik Interdisciplinary Cyber Research Center at Tel Aviv University.

A Limited Independence Tail Bound

We rely on the following: [[LL14]] Let $X_1, \ldots, X_N$ be $\{0,1\}$-valued random variables, let $t, \tau \in \mathbb{Z}^+$ satisfy $0 < t < \tau < N$. Then

$$\Pr\left[\sum_{i=1}^{N} X_i \ge \tau\right] \le \frac{1}{\binom{\tau}{t}} \cdot \sum_{A\in\binom{[N]}{t}} \mathbb{E}\left[\prod_{i\in A} X_i\right].$$

We apply this theorem to obtain a concentration bound on t-wise almost-dependent random variables. Let $X_1, \ldots, X_n$ be $\{0,1\}$-valued random variables that are t-wise β-dependent in Rényi∞-divergence with $\mathbb{E}\left[\sum_i X_i\right] = \mu$.

Then for any $\tau \in \mathbb{Z}^+$ with $\tau > \mu$,

$$\Pr\left[\sum_i X_i \geq \tau\right] \leq \beta \cdot \frac{\mu^k}{(\tau)_k},$$

where $k = \min(t, \lfloor \tau - \mu \rfloor)$ and $(\tau)_k = \tau \cdot (\tau - 1) \cdots (\tau - k + 1)$ denotes the k^{th} falling factorial of τ.

Proof. We invoke Appendix A. For any $k < \tau$ and $k \leq t$, we have

$$\begin{aligned}
\Pr\left[\sum_{i=1}^{N} X_i \geq \tau\right] &\leq \binom{\tau}{k}^{-1} \cdot \sum_{A \in \binom{[n]}{k}} \mathbb{E}\left[\prod_{i \in A} X_i\right] \\
&\leq \beta \cdot \binom{\tau}{k}^{-1} \cdot \sum_{A \in \binom{[n]}{k}} \prod_{i \in A} \mathbb{E}\left[X_i\right] && \text{(by k-wise β-dependence)} \\
&= \beta \cdot \binom{\tau}{k}^{-1} \cdot \sum_{1 \leq i_1 < \cdots < i_k \leq [n]} \prod_{j \in [k]} \mathbb{E}\left[X_{i_j}\right] \\
&\leq \frac{\beta}{k!} \cdot \binom{\tau}{k}^{-1} \cdot \sum_{\text{distinct } i_1, \ldots, i_k} \prod_{j \in [k]} \mathbb{E}\left[X_{i_j}\right] \\
&\leq \frac{\beta}{k!} \cdot \binom{\tau}{k}^{-1} \cdot \sum_{i_1, \ldots, i_k} \prod_{j \in [k]} \mathbb{E}\left[X_{i_j}\right] \\
&= \frac{\beta}{k!} \cdot \binom{\tau}{k}^{-1} \cdot \mu^k \\
&= \beta \cdot \frac{\mu^k}{(\tau)_k}.
\end{aligned}$$

This is minimized by picking $k \leq t$ as large as possible subject to $\tau - k + 1 \geq \mu$, i.e. $k = \min(t, \lfloor \tau - \mu + 1 \rfloor)$. □

B Covering Number Bounds

q-ary n-dimensional Hamming space is the metric space $([q]^n, \Delta)$, where $\Delta(x,y) = \left|\{i : x_i \neq y_i\}\right|$. In a metric space (X, d), the ball of radius r centered at x, which we denote by $B_r(x)$, is the set $\{y : d(x,y) \leq r\}$. The sphere of radius r centered at x, which we denote by $S_r(x)$, is $\{y : d(x,y) = r\}$. The q-ary entropy function is $H_q(x) \overset{\mathsf{def}}{=} x \log_q(q-1) - x \log_q(x) - (1-x)\log_q(1-x)$. The following bounds are well-known. In q-ary n-dimensional Hamming space, we have $q^{n \cdot H_q(r/n)} \cdot n^{-O(1)} \leq |B_r(x)| \leq q^{n \cdot H_q(r/n)}$ for all $r \leq n \cdot (1 - 1/q)$.

In q-ary n-dimensional Hamming space, any ball of radius $r_1 \leq n \cdot (1 - 1/q)$ can be covered by $\mathrm{poly}(n) \cdot \ln(q) \cdot q^{n \cdot \left(H_q(r_1/n) - H_q(r_0/n)\right)}$ balls of radius r_0 for any $r_0 \leq r_1$.

References

[BGGZ19] Jeremiah Blocki, Venkata Gandikota, Elena Grigorescu, and Samson Zhou. Relaxed locally correctable codes in computationally bounded channels. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 2414–2418. IEEE, 2019.

[CCH$^+$18] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, and Ron D. Rothblum. Fiat-shamir from simpler assumptions. *IACR Cryptol. ePrint Arch.*, 2018:1004, 2018.

[CCR16] Ran Canetti, Yilei Chen, and Leonid Reyzin. On the correlation intractability of obfuscated pseudorandom functions. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part I*, volume 9562 of *Lecture Notes in Computer Science*, pages 389–415. Springer, 2016.

[CCRR18] Ran Canetti, Yilei Chen, Leonid Reyzin, and Ron D. Rothblum. Fiat-shamir and correlation intractability from strong kdm-secure encryption. In *EUROCRYPT (1)*, volume 10820 of *Lecture Notes in Computer Science*, pages 91–122. Springer, 2018.

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

[DGL04] Yan Zhong Ding, Parikshit Gopalan, and Richard J Lipton. Error correction against computationally bounded adversaries. *Manuscript. Appeared initially as [Lip94]*, 2004.

[Eli57] Peter Elias. List decoding for noisy channels. Technical Report 335, Research Laboratory of Electronics, MIT, 1957.

[GR08] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Trans. Inf. Theory*, 54(1):135–150, 2008.

[GR09] Venkatesan Guruswami and Atri Rudra. Better binary list decodable codes via multilevel concatenation. *IEEE Trans. Inf. Theory*, 55(1):19–26, 2009.

[GS16] Venkatesan Guruswami and Adam Smith. Optimal rate code constructions for computationally simple channels. *Journal of the ACM (JACM)*, 63(4):1–37, 2016.

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

[HO08] Brett Hemenway and Rafail Ostrovsky. Public-key locally-decodable codes. In *Annual International Cryptology Conference*, pages 126–143. Springer, 2008.

[HOSW11] Brett Hemenway, Rafail Ostrovsky, Martin J Strauss, and Mary Wootters. Public key locally decodable codes with short keys. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 605–615. Springer, 2011.

[KNR09] Eyal Kaplan, Moni Naor, and Omer Reingold. Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica*, 55(1):113–133, 2009.

[KRR17] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of fiat-shamir for proofs. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, volume 10402 of *Lecture Notes in Computer Science*, pages 224–251. Springer, 2017.

[Lip94] Richard J. Lipton. A new approach to information theory. In *STACS*, volume 775 of *Lecture Notes in Computer Science*, pages 699–708. Springer, 1994.

[LL14] Nathan Linial and Zur Luria. Chernoff's inequality - a very elementary proof, 2014.

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

[OPS07] Rafail Ostrovsky, Omkant Pandey, and Amit Sahai. Private locally decodable codes. In *International Colloquium on Automata, Languages, and Programming*, pages 387–398. Springer, 2007.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 89–114. Springer, 2019.

[SS16] Ronen Shaltiel and Jad Silbak. Explicit list-decodable codes with optimal rate for computationally bounded channels. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[SS20] Ronen Shaltiel and Jad Silbak. Explicit uniquely decodable codes for space bounded channels that achieve list-decoding capacity. *Electronic Colloquium on Computational Complexity (ECCC)*, 27:47, 2020.

[Woz58] John M Wozencraft. List decoding. *Quarterly Progress Report*, 48:90–95, 1958.