Binary Codes for Error Detection and Correction in a Computationally Bounded World

Jad Silbak [*] Daniel Wichs [†]

February 9, 2025

Abstract

We study error detection and correction in a computationally bounded world, where errors are introduced by an arbitrary *polynomial-time* adversarial channel. Our focus is on *seeded* codes, where the encoding and decoding procedures can share a public random seed, but are otherwise deterministic. We can ask for either *selective* or *adaptive* security, depending on whether the adversary can choose the message being encoded before or after seeing the seed. For large alphabets, a recent construction achieves essentially optimal rate versus error tolerance trade-offs under minimal assumptions, surpassing information-theoretic limits. However, for the binary alphabet, the only prior improvement over information theoretic codes relies on non-standard assumptions justified via the random oracle model. We show the following:

- **Selective Security under LWE:** Under the learning with errors (LWE) assumption, we construct selectively secure codes over the binary alphabet. For error detection, our codes achieve essentially optimal rate $R \approx 1$ and relative error tolerance $p \approx \frac{1}{2}$. For error correction, they can uniquely correct $p < 1/4$ relative errors with a rate R that essentially matches that of the best list-decodable codes with error tolerance p. Both cases provide significant improvements over information-theoretic counterparts. The construction relies on a novel form of 2-input correlation intractable hash functions that we construct from LWE.
- **Adaptive Security via Crypto Dark Matter:** Assuming the exponential security of a natural collision-resistant hash function candidate based on the "crypto dark matter" approach of mixing linear functions over different moduli, we construct adaptively secure codes over the binary alphabet, for both error detection and correction. They achieve essentially the same trade-offs between error tolerance p and rate R as above, with the caveat that for error-correction they only do so for sufficiently small values of p.

[*]Northeastern University, Email: jadsilbak@gmail.com. Research supported by the Khoury College Distinguished Postdoctoral Fellowship.

[†]Northeastern University and NTT Research. Research supported by NSF CNS-2349972 and CNS-2055510.

Contents

1 Introduction **1**
- 1.1 Our Results: Seeded Codes for the Binary Alphabet . . . 2
- 1.2 Our Techniques . . . 3

2 Preliminaries and Generic Ingredients **7**
- 2.1 Notations . . . 7
- 2.2 Universal One-Way Hash Functions . . . 8
- 2.3 Correlation Intractable Hashing . . . 8
- 2.4 Standard Error Correction/Detection Codes . . . 9

3 Seeded Codes: Adaptive and Selective Security **10**

4 Construction for Seeded Codes With Selective Security **12**
- 4.1 Ingredients for Our Construction . . . 12
  - 4.1.1 Target 2-Input Correlation Intractable Hashing . . . 12
  - 4.1.2 Nested List-Decodable Codes Meeting The Blokh-Zyablov Bound . . . 15
- 4.2 Proof of Theorem 4.1. . . . 16
  - 4.2.1 Proof of Lemma 4.8. . . . 18

5 Construction for Seeded Codes With Adaptive Security **19**
- 5.1 Everywhere Collision Resistance . . . 21
  - 5.1.1 Everywhere Collision Resistance $\Leftrightarrow$ Seeded Codes with Adaptive Security . . . 23
- 5.2 A Black-Box Transformation From Detection to Correction . . . 26

A Appendix **30**
- A.1 Proof of Theorem 4.7 . . . 30
  - A.1.1 Adopting the construction of Guruswami and Rudra for list decoding up to the Blokh-Zyablov bound. . . . 31

1 Introduction

We consider the classical problem of communication over a noisy adversarial channel. In this setting, a sender *encodes* a message into a codeword that is transmitted over the channel. The channel may introduce some bounded number of arbitrary errors by modifying a subset of the codeword symbols. A receiver gets the modified codeword and attempts to *decode* the message. The goal of *error-detection* is to recover the original message when no errors have occurred and to detect the presence of errors otherwise. The goal of *error-correction* is to 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.

A large body of work starting with Hamming [Ham50] studies codes for worst-case channels that can introduce arbitrary error patterns, subject to the number of errors being bounded. In this setting, 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.

A more recent line of works [Lip94, OPS07, MPSW10, HOSW11, GS16, SS21a, GHY20, BGGZ21, SS21b, SS22, SS24] studies error detection and correction under the natural additional restriction that the channel is *computationally bounded*. Interestingly, this restriction often allows us to beat the information-theoretic limits. To take advantage of computationally bounded channels, we cannot use a fixed deterministic code, since an efficient (non-uniform) adversary can always have hard-coded advice consisting of a worst-case message and error pattern for which detection/correction fails. Instead, we need some kind of randomized setup. Some prior works [Lip94, MPSW10] consider secret-coin trusted setups, such as shared secret keys or a public-key infrastructure. In contrast, we focus on *seeded codes*, where the only trusted setup consists of a public uniformly random seed that is known to the encoding/decoding algorithm as well as the adversarial channel. Some works [GS16, SS21a, SS21b, SS22, SS24] also consider a setting where the complexity of the channel can be bounded by some a-priori known polynomial and the complexity of the code can grow accordingly to exceed that of the channel. In contrast, we focus on codes that have a fixed polynomial run-time and achieve security against arbitrary polynomial time adversarial channels.

Selective vs Adaptive Security. Elaborating on our setting, we consider a polynomial-time computable family of seeded codes $(\mathrm{Enc}_s, \mathrm{Dec}_s)$ with a common random public seed s, and we want error detection/correction to hold against an arbitrary polynomial-time adversary. We distinguish between *selective* and *adaptive* security, depending on whether the message being encoded can depend on the seed. In the selective setting, the adversary first chooses a message m upfront. Then a random seed s is chosen and the adversary gets s, which defines the codeword $c := \mathrm{Enc}_s(m)$. The adversary then chooses an erroneous codeword c' within relative Hamming distance p from c. In the adaptive setting, the random seed s is chosen first and given to the adversary. The adversary then adaptively chooses the message m, which is encoded to $c := \mathrm{Enc}_s(m)$, along with an erroneous codeword c' within relative Hamming distance p from c. In both cases, the goal of error detection is to ensure that, with overwhelming probability, the erroneous codeword c' decodes to the correct message m when there are no errors $c' = c$, and decodes to $\perp$ when there are errors $c' \neq c$. Error correction ensures that with overwhelming probability c' decodes to m. Although adaptive security is the stronger and more desirable notion, selective security is already meaningful and captures settings where the adversary may not be able to fully control the messages that honest parties are encoding.[1]

[1] There is a generic method to convert selective security to adaptive security via complexity leveraging, but unfortunately it blows up the parameters to a point where the resulting codes are often uninteresting.

Prior Work. The work of Grossman, Holmgren and Yogev [GHY20] introduced the notion of *seeded codes* and gave an adaptively secure 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 assumption. For a large poly-sized alphabet, and for any error-tolerance $p < 1/2$ they achieve optimal error correction with rate $R \approx 1 - p$, where the "$\approx$" hides additive constant factors that can be made arbitrarily small. For the binary alphabet, and for sufficiently small p, they achieve error correction with the same rate R as the best known information-theoretic efficiently list-decodable binary codes with error-tolerance p, which is given by the Blokh-Zyablov bound $R_{BZ}(p)$ [GR08a, BZ76]. While this bound has a complicated expression, it is significantly higher than the corresponding rate for the best known information-theoretic uniquely decodable binary codes with error-tolerance p.

The work of Silbak and Wichs [SW24] considered *randomized / self-seeded* codes, which improve over seeded codes in that the seed s 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 such codes for large constant sized alphabets with essentially optimal parameters under minimal assumptions: just one-way functions in the case of selective security and collision-resistant hash functions in the case of adaptive security. The codes achieve essentially optimal error-detection with error tolerance $p \approx 1$ and rate $R \approx 1$ and, as well as essentially optimal error-correction with error tolerance $p < 1/2$ and rate $R \approx 1 - p$. Unfortunately, their results crucially rely on the use of a large alphabet and do not extend to the binary alphabet case.

In summary, while the large alphabet setting was essentially fully resolved in [SW24], the only prior known result that beats information-theoretic codes in the binary alphabet setting [GHY20] requires strong non-standard cryptographic hardness assumptions that are only justified via the random oracle model.

1.1 Our Results: Seeded Codes for the Binary Alphabet

We revisit the problem of seeded codes for the binary alphabet, and provide two new results.

Selective Security under LWE. Under the learning with errors (LWE) assumption, we construct selectively secure binary codes. For error detection, the codes achieve essentially optimal rate $R \approx 1$ and relative error tolerance $p \approx \frac{1}{2}$. For error correction, they can uniquely correct $p < 1/4$ relative errors with essentially the same rate R as the best currently known information-theoretic binary list-decodable codes with error tolerance p, which is given by the Blokh-Zyablov bound $R_{BZ}(p)$ [BZ76]. The construction relies on a novel specialized form of 2-input correlation intractable hash functions that we construct from LWE. These hash functions offer the following guarantee for any function f of some bounded polynomial size that outputs a list of values. A polynomial-time adversary that selectively chooses a message m and gets a random hash function h from the hash family should be unable to find a message $m' \neq m$ such that $h(m') \in f(m, m', h(m))$ with better than negligible probability.[2]

Adaptive Security via Crypto Dark Matter. Assuming the exponential security of a new collision-resistant hash function candidate described below, we construct adaptively secure codes. For error detection, our codes achieve essentially optimal rate $R \approx 1$ and relative error tolerance $p \approx \frac{1}{2}$. For error correction, they can uniquely correct p relative errors with rate matching the Blokh-Zyablov bound $R_{BZ}(p)$, but only for sufficiently small values of p.

[2]We note that this type of hash function is significantly weaker than the general 2-input correlation-intractability needed in [GHY20], and we still do not know how to achieve the latter from any standard assumptions.

The hash function candidate is based on the "crypto dark matter" [BIP+18] approach of mixing linear functions over different moduli. In particular, the hash function takes an input x and first applies a *random expanding linear transformation over* $\mathbb{F}_3$ followed by a *random shrinking linear transformation over* $\mathbb{F}_2$, where the matrices for these transformations are part of the seed. The assumption is that no polynomial-time attacker can find a collision with probability that is exponentially better than trivial guessing.

Discussion of Parameters. For error detection, it is easy to show that for any constant rate $R > 0$ it is impossible to get error tolerance $p > \frac{1}{2}$ against efficient channels. This follows from the same analysis as the Plotkin bound, which can be used to show that two random codewords have relative Hamming distance $\leq \frac{1}{2}$ with noticable probability; see [MPSW10, Proposition 1]. Therefore, an efficient selective channel that chooses a random m, m' and replaces $c = \mathrm{Enc}_s(m)$ with $c' = \mathrm{Enc}_s(m')$ breaks error detection using $p = \frac{1}{2}$ relative errors with noticeable probability. This shows that the combination of rate $R \approx 1$ and error-tolerance $p \approx \frac{1}{2}$ achieved by our codes is optimal. In contrast, for information-theoretic codes, an extension of the Plotkin bound shows that detection with error-tolerance $p \approx \frac{1}{2}$ requires rate $R \approx 0$.

For error correction, it is similarly easy to see that for any constant rate $R > 0$ it is impossible to get error tolerance $p > \frac{1}{4}$. An efficient selective channel that chooses a random messages m and gets $c = \mathrm{Enc}_s(m)$, can then choose a random independent codeword $c^* = \mathrm{Enc}_s(m^*)$ and, by the preceding argument, c^* is at relative distance at most $1/2$ from c with noticable probability. In that case it is easy to find some c' that lies at distance at most $1/4$ from both c and c^*; by replacing c with c', the decoding algorithm cannot tell whether the original message was m or m^* and must output an incorrect value with noticeable probability.

However, understanding the optimal rate R for different choices of the error-correction tolerance $p < 1/4$ is more complicated. The Shannon bound for the binary symmetric channel (BSC) [Sha48] shows that to correct a p fraction of *random errors*, the optimal rate is at most $R \approx 1 - H(p)$ where H is the binary entropy function. This therefore also gives an upper bound on the achievable rate for efficient channels. Our positive results for error-correction, as well as those of [GHY20], show that for sufficiently small $p < 1/4$ the achievable rate R is the same as that of certain types of efficiently list-decodable codes with error-tolerance p. If we ignore efficiency, then these types of list decodable codes are known to exist with $R \approx 1 - H(p)$ matching the Shannon bound [Sha48], which gives hope that future improvements in efficient list-decoding will allow us to reach to this optimal bound. (In contrast, unique error-correction in the information-theoretic setting is only known to be possible inefficiently with much smaller rate $R \approx 1 - H(2p)$ via the Gilbert-Varshamov bound.) The best currently known efficient list-decodable codes [GR08a], which have the properties needed for out positive results, only go up to the so-called Blokh-Zyablov bound [BZ76] denoted by $R_{BZ}(p)$, which has a complicated expression (see Equation (1) in Section 2). For example, for $p \approx 1/4$, the Shannon bound would imply an optimal rate of $R \approx 0.188$ while the Blokh-Zyablov bound gives $R \approx 0.037$; in contrast, unique error-correction in the information-theoretic setting with $p \approx 1/4$ implies a rate $R \approx 0$. Another factor that differentiates the two positive results in our work from each other and from [GHY20] is the range of p for which they meet the Blokh-Zyablov bound. Our selectively secure construction from LWE does so for the full range of $p < 1/4$. On the other hand, our adaptively secure construction from crypto dark matter and the construction of [GHY20] only do so for sufficiently small p. Concretely, our crypto dark matter construction requires $p < R_{BZ}(p)/4$, while [GHY20] requires $p < H^{-1}(R_{BZ}(p) + H(p))/2$; both of which are satisfied when $p < p_{max}$ for some $.05 \leq p_{max} \leq .1$.

1.2 Our Techniques

We now give an overview of our techniques for the various results.

Selectively Secure Error Detection from LWE. We construct a selectively secure code $(\mathrm{Enc}_s, \mathrm{Dec}_s)$ with error-tolerance $p \approx 1/2$ and rate $R \approx 1$. Let $\mathcal{H} = \{h : \{0,1\}^k \to \{0,1\}^v\}$ be a family of hash functions with small output length $v = o(k)$, whose security properties we will specify later. Let $\mathbf{G} \in \mathbb{F}_2^{v \times n}$ be the generator matrix of a linear efficiently list-decodable code that maps a v-bit message to an $n = (k+v)$-bit codeword. The code has low rate $R = v/(k+v) \approx 0$, but we want it to have high error-tolerance $p \approx 1/2$, meaning that it can list-decode from a p fraction of errors with a polynomial-size list. Such linear codes can be constructed using code concatenation (e.g., [GR08a]). Extend the generator matrix $\mathbf{G}$ into a full-rank matrix $\mathbf{A} = \left[\begin{smallmatrix}\mathbf{B}\\ \mathbf{G}\end{smallmatrix}\right] \in \mathbb{F}_2^{n \times n}$ arbitrarily, by completing the basis. The encoding algorithm $\mathrm{Enc}_s(m)$ takes as input a message $m \in \{0,1\}^k$, computes $h(m)$, and outputs the codeword

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

where the hash function $h \leftarrow \mathcal{H}$ is given in the seed $s = h$ of the code. The decoding algorithm $\mathrm{Dec}_s(c')$ first computes $(m', y') := c'\mathbf{A}^{-1}$ and then checks $h(m') = y'$; if so it outputs m' else $\perp$.

It is easy to see that the rate of the overall code is $R = k/(k+v) \approx 1$ and that decoding gives the correct answer when there are no errors. If there are errors, with $c' = c + e$ for for some error vector $e \neq 0$ having relative Hamming weight $\leq p$ and the decoding algorithm outputs some value $(m', y') \neq \perp$ then it must hold that $m \neq m'$, $h(m') = y'$ and:

$$h(m')\mathbf{G} = (m - m')\mathbf{B} + h(m)\mathbf{G} + e.$$

By list-decoding of the code generated by $\mathbf{G}$ we therefore have

$$h(m') \in \underbrace{\mathsf{ListDec}_{\mathbf{G}}((m - m')\mathbf{B} + h(m)\mathbf{G})}_{f(m,m',h(m))}.$$

An adversary that breaks the selective error-detection property of the code, must therefore also break the following form of *target 2-input correlation-intractability (T2CI)* of the hash family $\mathcal{H}$:

- The adversary chooses a message m and a bounded poly-size circuit f whose output is a list.[3]
- The adversary gets a random hash $h \leftarrow \mathcal{H}$.
- The adversary outputs m' and wins if $m' \neq m$ and $h(m') \in f(m, m', h(m))$.

Constructing a T2CI Hash. We show how to construct a *T2CI secure hash family*, for which no PPT adversary can win the above game with better than negligible probability, from LWE. We rely on the results of [CCH+19, PS19], which construct a more basic form of 1-input correlation intractable (CI) hash functions under LWE, ensuring that no PPT adversary can win the following game with better than negligible probability:

- The adversary chooses a bounded poly-size circuit f' whose output is a list.
- The adversary gets a random seed $h \leftarrow \mathcal{H}$.
- The adversary outputs m' and wins if $h(m') \in f'(m')$.

Perhaps a first thought would be that CI may directly imply T2CI security by simply defining $f'_{m,h(m)}(m') = f(m, m', h(m))$. But this doesn't work since the function f' cannot depend on choice of h and hence

[3] We allow any polynomial bound on the size of f, but the parameters of the hash family can depend on this bound.

cannot depend on $h(m)$. Instead we construct a T2CI secure hash function h by applying different subsets of CI hash functions for each input. Choose a collision-resistant (or just UOWHF) hash function $\hat{h}$ with output length ℓ and choose 2ℓ CI hash functions $\{h_i^b\}_{i\in[\hat{v}],b\in\{0,1\}}$. To compute $h(x)$, first compute $z := \hat{h}(x)$ and then use the bits $z = (z_1, \ldots, z_\ell)$ to select which CI hash functions to apply; output $y := (h_1^{z_1}(x), \ldots, h_\ell^{z_\ell}(x))$. If an adversary breaks T2CI of h, then by the security of the collision-resistant hash, it must be the case that $z \neq z'$ for $z = \hat{h}(m), z' = \hat{h}(m')$, and therefore there is some index j such that $z_j \neq z'_j$. We can guess the index j and, if we're correct, then a break on the T2CI security of h with some input m and function f implies a break on the CI security of $h_j^{1-z_j}$ with the function $f'_{j,m,h(m)}(m')$ that outputs the j'th component of each value in the list $f(m, m', h(m))$. Note that the function f' only depends on $h(m)$ which in turn only depends on the hash functions $\{h_i^{z_i}\}_{i\in[\ell]}$, but does not depend on $h_j^{1-z_j}$.

Selectively Secure Error Correction from LWE. We construct selectively secure seeded error-correction codes $(\mathrm{Enc}_s, \mathrm{Dec}_s)$ for any error tolerance $p < 1/4$ similarly. The only difference is that we now allow for $n > (k + v)$ to be larger, and we require that the component matrix $\mathbf{G} \in \mathbb{F}_2^{v\times n}$ is the generator matrix of a low-rate efficiently list-decodable code with high error tolerance $2p$, while the full matrix $\mathbf{A} = \left[\begin{smallmatrix}\mathbf{B}\\ \mathbf{G}\end{smallmatrix}\right] \in \mathbb{F}_2^{(k+v)\times n}$ is the generator matrix of an efficiently list-decodable code with lower error tolerance p but higher rate. Luckily, a small adaptation of the construction of [GR08a] yields such codes for $p < 1/4$ with rate $(k + v)/n \approx R_{BZ}(p)$ given by the Blokh-Zyablov bound.[4] The encoding algorithm $\mathrm{Enc}_s(m) = (m, h(m))\mathbf{A} = m\mathbf{B} + h(m)\mathbf{G}$ is the same as before. The decoding algorithm $\mathrm{Dec}_s(c^*)$ first applies list-decoding of the overall code generated by $\mathbf{A}$ on the value c^* to recover a list of tuples (m', y'). It checks if there is some tuple in the list for which $y' = h(m')$ and, if so, it outputs m' else it outputs $\perp$.

The overall rate of the code is $k/n \approx (k + v)/n \approx R_{BZ}(p)$. By the correctness of list-decoding for the code generated by $\mathbf{A}$, we know that if c^* is within relative distance p of $c = \mathrm{Enc}_h(m)$ then the tuple $(m, h(m))$ will be in the list generated during decoding. So the only way that $\mathrm{Dec}_h(c^*)$ can fail to output m is if there is some other tuple $(m', h(m'))$ in the list. Moreover, since $c = (m, h(m))\mathbf{A}$ and $c' = (m', h(m'))\mathbf{A}$ are each within relative distance p of c^*, they must be within relative distance $2p$ of each other. Writing $c' = c + e$ for for some error vector e we have:

$$h(m')\mathbf{G} = (m - m')\mathbf{B} + h(m)\mathbf{G} + e.$$

By list-decoding of the code generated by $\mathbf{G}$ we therefore have

$$h(m') \in \underbrace{\mathsf{ListDec}_{\mathbf{G}}((m - m')\mathbf{B} + h(m)\mathbf{G})}_{f(m,m',h(m))}.$$

Therefore, by the same argument as previously, an adversary that breaks the selective error-correction of the code, must therefore also break the T2CI security of the hash.

Adaptively Secure Error Detection from Crypto Dark-Matter. We construct a candidate seeded error-detection code $(\mathrm{Enc}_s, \mathrm{Dec}_s)$ with adaptive security based on the "crypto dark matter" approach of combining linear functions over different moduli [BIP+18]. This approach was previously used to derive new simple candidates for (weak) pseudorandom functions in low complexity classes. To the best of our knowledge, we are the first to use this approach to construct a new cryptographic primitive which is unknown from other standard assumptions. At a high level, we use dark matter crypto to construct an efficiently invertible

[4] Random linear codes give an inefficient instantiation with rate $R \approx 1 - H(p)$ matching the Shannon bound [Sha48].

high-rate function which acts as an (exponentially-secure) collision-resistant hash when restricted to every small subset of output bits, and we show this implies adaptive error-detection security.

Let $\mathsf{map}_{a\to b}(x)$ be the function that maps a vector $x \in \mathbb{Z}_a^n$ to $y \in \mathbb{Z}_b^{\lceil (\log_b a)n \rceil}$ by parsing x as the digits of an integer $< a^n$ in base a and re-writing it in base b to derive y. This function is efficiently invertible. The encoding algorithm $\mathrm{Enc}_s(m)$ takes as input $m \in \{0,1\}^k$ and applies $\tilde{m} := \mathsf{map}_{2\to 3}(m) \in \mathbb{F}_3^{\tilde{k}}$ for $\tilde{k} = \lceil (\log_3 2)k \rceil$. It then applies a random *expanding linear transformation over* $\mathbb{F}_3$ to derive $y := \tilde{m} \cdot \mathbf{B}$ where $\mathbf{B} \leftarrow \mathbb{F}_3^{\tilde{k}\times \ell}$ for $\ell = (1+\delta)\tilde{k}$ with an arbitrary small constant $\delta > 0$. Then it computes $\tilde{y} := \mathsf{map}_{3\to 2}(y) \in \mathbb{F}_2^{\tilde{\ell}}$ for $\tilde{\ell} = \lceil (\log_2 3)\ell \rceil$ and applies a second random *expanding linear transformation over* $\mathbb{F}_2$ to derive the output $c := \tilde{y} \cdot \mathbf{A}$ where $\mathbf{A} \leftarrow \mathbb{F}_2^{\tilde{\ell}\times n}$ for $n = (1+\delta)\tilde{\ell}$. In other words:

$$\mathrm{Enc}_s(m) = \mathsf{map}_{3\to 2}(\mathsf{map}_{2\to 3}(m) \cdot \mathbf{B}) \cdot \mathbf{A},$$

where we think of $\mathbf{A}, \mathbf{B}$ as the seed s. With high probability, both $\mathbf{A}, \mathbf{B}$ are invertible. Therefore, the decoding algorithm $\mathrm{Dec}_s(c')$ simply attempts to compute $m' = \mathrm{Enc}_s^{-1}(c')$ and outputs m' if such an inverse exists and $\perp$ otherwise. The rate of the construction is $k/n \approx 1$ by choosing δ sufficiently small. We conjecture that it is adaptively secure for $p = 1/2 - \varepsilon$ for any constant $\varepsilon > 0$. That is, given the seed s consisting of the two matrices $\mathbf{A}, \mathbf{B}$, no polynomial time adversary can find $m \neq m'$ such that $\mathrm{Enc}_s(m)$ and $\mathrm{Enc}_s(m')$ are within relative distance p of each other.

We connect the above "error-detection conjecture" to a simpler "collision-resistance conjecture" for a simple candidate collision-resistant hash function family. Define the hash family

$$h_s(m) := \mathsf{map}_{3\to 2}(\mathsf{map}_{2\to 3}(m) \cdot \mathbf{B}) \cdot \mathbf{A}$$

the same way as the above encoding procedure, except that we now choose $\mathbf{A} \leftarrow \mathbb{F}_2^{\tilde{\ell}\times t}$ for a small $t \ll k$ (i.e., the security parameter) to be a random *shrinking* linear transformation over $\mathbb{F}_2$. Note that the hash function $h_s(m)$ is the same as taking $\mathrm{Enc}_s(m)$ and outputting some fixed subset of t bits of the codeword. While we are not aware of this hash function being studied previously, it is a simple and natural candidate. The intuition for its security is that, although it is easy to find collisions $\tilde{y}, \tilde{y}'$ such that $\tilde{y} \cdot \mathbf{A} = \tilde{y}' \cdot \mathbf{A}$ using linear algebra over $\mathbb{F}_2$, it appears hard to find such collisions where $y = \mathsf{map}_{3\to 2}^{-1}(\tilde{y}), y' = \mathsf{map}_{3\to 2}^{-1}(\tilde{y}')$ are both also in the row-span of $\mathbf{B}$ over $\mathbb{F}_3$, which is a sparse subset of $\mathbb{F}_3^{\ell}$. This intuition follows the general "crypto dark matter" methodology, which says that mixing linear functions over different moduli introduces cryptographic hardness. Concretely, we conjecture that, for any polynomial-time adversary who is given a random seed s, the probability of finding a collision on h_s is $\leq 2^{o(t)}/2^t$. Note that the trivial attack that guesses a polynomial number of inputs and sees if any collide only succeeds with probability $\mathrm{poly}(t)/2^t$. Hence the above says that no adversary can do exponentially better than trivial.

The collision-resistance conjecture implies the error-detection conjecture. Suppose a polynomial-time adversary, given a random seed s', has a non-negligible probability of finding m, m' such that $\mathrm{Enc}_{s'}(m)$ and $\mathrm{Enc}_{s'}(m')$ are within relative distance $1/2 - \varepsilon$ for some $\varepsilon > 0$. We use such an adversary to break the hash function h_s for a random s with significantly higher probability than trivial. Given a random seed $s = (\mathbf{A}, \mathbf{B})$ for the hash function h_s, we generate a seed $s' = (\mathbf{A}', \mathbf{B})$ for the code by placing the rows of $\mathbf{A}$ in t random positions of $\mathbf{A}'$ and choosing the remaining rows of $\mathbf{A}'$ randomly. Then if the adversary finds m, m' as above and $\mathrm{Enc}_{s'}(m), \mathrm{Enc}_{s'}(m')$ agree in all of the t chosen positions, then m, m' are also a collision on the hash function with $h_s(m) = h_s(m')$. Since the adversary's view is independent of which positions were chosen, this happens with probability $\binom{(1/2+\varepsilon)n}{t} / \binom{n}{t} \geq 2^{\Omega(t)}/2^t$.

Adaptively Secure Error Correction from Crypto Dark-Matter. Lastly, we give a general transformation from error-detection to error-correction. By applying this transformation to the above error-detection

code based on the crypto dark matter assumption, we get a corresponding error-correction code under the same assumption.[5]

We use a *systematic* efficiently list-decodable information-theoretic error-correcting code $(\mathrm{Enc}_{\mathrm{List}}, \mathrm{Dec}_{\mathrm{List}})$ where the codeword $c \in \{0,1\}^n$ contains the message $m \in \{0,1\}^k$ in the first k positions, having relative error-tolerance p and rate R. Note that any linear code can be made systematic while preserving its list decoding properties, and therefore we can use the codes of [GR08a] with $R = R_{BZ}(p)$. We also use an adaptively secure seeded error-detection code $(\mathrm{Enc}'_s, \mathrm{Dec}'_s)$ with error tolerance $p' \approx 1/2$ and rate $R' \approx 1$ as above. We construct a seeded error-correction code $(\mathrm{Enc}_s, \mathrm{Dec}_s)$ by defining $\mathrm{Enc}_s(m) = \mathrm{Enc}_{\mathrm{List}}(\mathrm{Enc}'_s(m))$ and defining $\mathrm{Dec}_s(c')$ to first apply list decoding on c' to recover a list of values $\tilde{c}_i$, then check if any of them satisfy $m'_i = \mathrm{Dec}'(\tilde{c}_i) \neq \perp$, and if so output the first such m'_i, else $\perp$. The above achieves rate $RR' \approx R$. We argue that it achieves adaptively secure error-correction with tolerance p as long as $p < p'R/2 \approx R/4$. In other words, for sufficiently small p, we get the same rate for unique decoding as the best linear list-decodable code for the same value of p.

To argue security, note that if c' is within relative distance p of $c = \mathrm{Enc}_s(m) = \mathrm{Enc}_{\mathrm{List}}(\mathrm{Enc}'_s(m))$ then, by the correctness of list decoding, the value $\mathrm{Enc}'_s(m)$ is in the recovered list. Therefore, the only way that decoding can fail to output m is if some $\mathrm{Enc}'_s(m')$ for $m' \neq m$ also appears in the list. By the error-detection security, it must be that $\mathrm{Enc}'_s(m)$ and $\mathrm{Enc}'_s(m')$ are at relative distance at least p'; else an adversary could break error-detection by changing $\mathrm{Enc}'_s(m)$ to $\mathrm{Enc}'_s(m')$. But because the $\mathrm{Enc}_{\mathrm{List}}$ is systematic, it must then be the case the at $\mathrm{Enc}_s(m) = (\mathrm{Enc}'_s(m), \cdots)$ and $\mathrm{Enc}_s(m') = (\mathrm{Enc}'_s(m'), \cdots)$ are at relative distance at least $p'R$ from each other. By the triangle inequality, one of them must therefore be at a distance at least $p'R/2$ away from c'. But list decoding only outputs values within distance $< p$ of c', which gives a contradiction when $p < p'R/2 \approx R/4$.

Organization of the Paper.

In Section 2 we give general notations and define standard ingredients. In Section 3 we give formal definitions for seeded codes with selective and adaptive security. In Section 4 we present our result seeded codes with selective security. In Section 5 we present our result for seeded codes with adaptive security.

2 Preliminaries and Generic Ingredients

2.1 Notations

All logarithms that are not specified are taken in base 2. Let poly stand for the set of all polynomials. Let PPT stand for non-uniform probabilistic polynomial-time. We note that while we rely on the non-uniform model of computation by default, all our results also hold if we were to define PPT in the uniform model. We use sans-serif (e.g., A) for PPT algorithms and bold (e.g, $\mathbf{G}$) for matrices. A function $\nu \colon \mathbb{N} \mapsto [0,1]$ is *negligible*, denoted $\nu(k) = \mathrm{neg}(k)$, if $\nu(k) < 1/p(k)$ for every $p \in \mathrm{poly}$ and large enough k. For a set S, let $x \leftarrow S$ denote that x was drawn uniformly from S. Given a distribution P, we use $x \leftarrow P$ to denote that x is chosen according to P.

Hamming distance. The Hamming distance between $x, y \in [q]^n$ is $\Delta(x,y) = |\{i : x_i \neq y_i\}|$. The relative Hamming distance between $x, y \in [q]^n$ is $\delta(x,y) = \frac{\Delta(x,y)}{n}$.

[5] The transformation works for both the selective and adaptive settings, and we could also apply it to our construction of selective error detection from LWE to get an alternate selective error correction code from LWE. However, it only works for more restricted parameter settings than our direct construction of selective error correction from LWE above.

Distributions and random variables. The statistical distance (also known as, variation distance) of two distributions P and Q over a discrete domain X is defined by $\mathsf{SD}(P,Q) = \max_{S \subseteq X} |P(S) - Q(S)|$.

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

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

The best known explicit codes with efficient list decoding algorithms are by Guruswami and Rudra [GR08a]. 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 [GR08a], appears in Appendix A.

2.2 Universal One-Way Hash Functions

Definition 2.1. (Universal one-way hash functions)**.** For a security parameter $k \in \mathbb{N}$. A family of functions $\mathcal{H} = \{h_s \colon \{0,1\}^{n(k)} \to \{0,1\}^{\ell(k)}\}_{s \in \{0,1\}^{\ell(k)}}$ is a family of ***universal one-way hash functions*** (UOWHFs) if it satisfies:

1. Efficiency: Given $s \in \{0,1\}^{\ell(k)}$ and $x \in \{0,1\}^{n(k)}$, $h_s(x)$ can be evaluated in time $\mathrm{poly}(k)$.
2. Shrinking: $\ell(k) < n(k)$.
3. Target Collision Resistance: For every PPT A consider the following randomized experiment:
   - $(x, \mathrm{state}) \leftarrow \mathsf{A}(1^k) \in \{0,1\}^{n(k)} \times \{0,1\}^*$,
   - $s \leftarrow \{0,1\}^{\ell(k)}$,
   - $x' \leftarrow \mathsf{A}(\mathrm{state}, s)$.

   Then $\mathbf{Pr}\left[x \neq x' \wedge h_s(x) = h_s(x')\right] = \mathrm{neg}(k)$.

The notion of universal one-way hash functions (UOWHF) was first introduced by Naor and Yung [NY89]. Rompel [Rom90] gave the first UOWHF construction from arbitrary one-way functions.

Theorem 2.2 (One-way functions imply universal one-way hash function, [NY89, Rom90])**.** *There is a black-box construction of universal one-way hash functions (UOWHFs) from one-way functions (OWFs). In particular, assuming OWFs, for any arbitrarily large constant c and arbitrarily small constant $\delta > 0$ there is a UOWHF with input length $n(k) = k^c$ and output/seed length $\ell(k) = k^\delta$. In particular, this implies that if the LWE assumption holds, there exists a universal one-way hash function since the LWE assumption implies OWFs.*

2.3 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 $\mathcal{H} = \{h_s\}_{s \in \{0,1\}^{\ell(k)}}$ is correlation intractable CI for a spare relationship $R(x,y)$, if given a random seed $s \leftarrow \{0,1\}$ it is computationally hard to find an x, such that $(x, h_s(x)) \in R$. A hash, function is fully correlation intractable if it is correlation intractable for all spare relationships. The only known constructions

for the full notion [KRR17, CCRR18], require non-standard assumptions that are not falsifiable except with exponential time.

Canetti et al. [CCH+19, CLW18], constructed a correlation intractable hash functions for efficiently computable relationships, that run in some fixed time bound t. Specifically, (x, y) belongs to such a relationship, if there exists a circuit f of $size(t)$ such that $f(x) = y$. Their construction relied on fully homomorphic encryption scheme (FHE). Later work of Peikert and Shiehian [PS19] gave the first construction from plain LWE. The above notion, of efficiently computable relationships could easily be extended to function families that output a polynomial list, that is, $y \in f(x)$ where $|f(x)|$ is of polynomial size (such relationships were referred to as "efficiently enumerable" in [CLW18]). This is the notion that we will make use of for our results; we define it formally below.

Definition 2.3. (Correlation intractable hash functions (CI))**.** Let $k \in \mathbb{N}$ be a security parameter and $t \in \text{poly}$ be a poly-time computable function. The hash family $\mathcal{H} = \{h_s \colon \{0,1\}^{n(k)} \to \{0,1\}^{m(k)}\}_{s \in \{0,1\}^{\ell(k)}}$ is ***correlation intractable for t-size circuits*** (denoted t-CI) if for every $size(t(k))$ circuit family $f = \{f_k : \{0,1\}^{n(k)} \to (\{0,1\}^{m(k)})^{L(k)}\}_{k \in \mathbb{N}}$ whose output is interpreted as a set, and every (non-uniform) PPT A:

$$\Pr_{\substack{s \leftarrow \{0,1\}^{\ell(k)}, \\ x \leftarrow \mathsf{A}(s)}} [h_s(x) \in f_k(x)] \leq \text{neg}(k).$$

Theorem 2.4 ([CCH+19, CLW18, PS19], t-CI from LWE)**.** *Under the LWE assumption, there exists an $m \in \text{poly}$ such that for all poly-time computable functions $t, n \in \text{poly}$ there exists a poly-time computable $\ell \in \text{poly}$ and a t-CI hash family $\mathcal{H} = \{h_s \colon \{0,1\}^{n(k)} \to \{0,1\}^{m(k)}\}_{s \in \{0,1\}^{\ell(k)}}$.*

2.4 Standard Error Correction/Detection 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 $m \in \text{Dec}(v)$.
- (α, ℓ, L)-***list-recovers***, if the function Dec is allowed to output a list of size at most L, $m \in \{0,1\}^k$ and for every sequence of sets $S_1, \cdots, S_n$ such that $S_i \subseteq [q]$ and $|S_i| \leq \ell$ for every $i \in [n]$ and $|i \colon \text{Enc}(m)_i \in S_i| \geq \alpha \cdot n$, we have $m \in \text{Dec}(S_1, \cdots, S_n)$.

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 $R = \frac{k}{n \cdot \log q}$.

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

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).

All standard codes considered in our paper are binary and explicit unless stated otherwise.

3 Seeded Codes: Adaptive and Selective Security

Seeded codes for computationally bounded channels were first defined in [GHY20]. Below, we define the syntax for seeded code. Loosely speaking, a seeded code is a pair of deterministic encoding and decoding algorithms $(\mathrm{Enc}, \mathrm{Dec})$. Intuitively, the "seed" s is a random string that is public and is known to the sender and receiver and any possible adversary. We stress however, 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.

Definition 3.1. (Seeded codes)**.** Let $n, \ell : \mathbb{N} \mapsto \mathbb{N}$ be poly-time computable functions such that $\ell \in \mathrm{poly}$. A ***Seeded code*** is a pair $(\mathrm{Enc}, \mathrm{Dec})$ such that

- Enc is a deterministic polytime algorithm that on input $m \in \{0,1\}^k$ and a seed $s \in \{0,1\}^{\ell(k)}$ outputs a string $c \in \{0,1\}^{n(k)}$. We use $\mathrm{Enc}_s(m)$ to denote $\mathrm{Enc}(m, s)$.
- Dec is a deterministic polytime algorithm that on input $z \in \{0,1\}^{n(k)}$ and seed $s \in \{0,1\}^{\ell(k)}$ outputs a string $\widehat{m} \in \{0,1\}^k \cup \{\perp\}$. We use $\mathrm{Dec}_s(z)$ to denote $\mathrm{Dec}(z, s)$.
- It holds that for every $m \in \{0,1\}^k$ and $s \in \{0,1\}^{\ell(n)}$, $\mathrm{Dec}_s(\mathrm{Enc}_s(m)) = m$.

For a fixed $k \in \mathbb{N}$, the rate of the code for k is defined to be $R_k = \frac{k}{n(k)}$. When $R = \lim_{k\to\infty} R_k \in [0,1]$ is well defined we say that R is the rate of the code.[6]

As shown in [GHY20, SW24], seeded codes are useful when dealing with computationally bounded adversaries because they enable us to achieve better error-correcting codes than the 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. Indeed, throughout this paper, when mentioning seeded codes, we implicitly assume that the adversary is computationally bounded.

Below, we define two security notions, selective and adaptive. The difference between the two notions lies in whether the message being encoded can depend on the seed.

Seeded codes with selective security. For our notion of selective security, we require that the message being sent does not depend on the seed. This is formalized in the experiment presented below, in which the message is first chosen, and after that a random seed is sampled, and we require that no efficient adversary A could find an error pattern that will contradict detection and correction.

Definition 3.2. (Seeded codes with selective security)**.** Let $0 \le p < 1/2$ be a constant. A seeded code $(\mathrm{Enc}, \mathrm{Dec})$ is said to have ***selective security*** if the following holds: For a PPT algorithm A and $k \in \mathbb{N}$, consider the following randomized experiment:

[6] We assume that n and ℓ are implicitly given as part of the description of the seeded code $(\mathrm{Enc}, \mathrm{Dec})$.

1. $(m, \text{state}) \leftarrow \mathsf{A}(1^k) \in \{0,1\}^k \times \{0,1\}^*$
2. Sampling $s \leftarrow \{0,1\}^{\ell(k)}$
3. $z \leftarrow \mathsf{A}(m, \text{state}, s)$

We say that the code:

- p-***detects***, if for every PPT algorithm A and every sufficiently large $k \in \mathbb{N}$:
  $\mathbf{Pr}\left[\delta(\text{Enc}_s(m), z) \leq p \wedge \text{Dec}_s(z) \notin \{\perp, m\}\right] \leq \text{neg}(k)$.
- p-***corrects*** if for every PPT algorithm A and every sufficiently large $k \in \mathbb{N}$:
  $\mathbf{Pr}\left[\delta(\text{Enc}_s(m), z) \leq p \wedge \text{Dec}_s(z) \neq m\right] \leq \text{neg}(k)$.

While this notion is weaker than the adaptive notion that we present below, we emphasize that this notion is still meaningful and useful. For example, if the sender wishes to send multiple messages (or a full book) that it decided on, then we are only required to sample a random public seed once (after deciding on the messages) to be able to send all of these messages.

Seeded codes with adaptive security. Below, we define the notion of seeded codes with adaptive security for detection and correction. Unlike our notion of selective security which is new to this paper, adaptive security was already considered in [GHY20] and [SW24]. Intuitively, this notion is stronger, since both the message and the error pattern could be chosen by the adversary depending on the public seed. We make this formal in the following definition:

Definition 3.3. (Seeded codes with adaptive security)**.** Let $0 \leq p < 1/2$ be a constant. A seeded code (Enc, Dec) is said to have ***adaptive security*** if the following holds: For a PPT algorithm A and $k \in \mathbb{N}$, consider the following randomized experiment:

1. $s \leftarrow \{0,1\}^{\ell(k)}$
2. $(m, z) \leftarrow \mathsf{A}(s) \in \{0,1\}^k \times \{0,1\}^{n(k)}$

We say that the code:

- p-***detects***, if for every PPT algorithm A:
  $\mathbf{Pr}\left[\delta(\text{Enc}_s(m), z) \leq p \wedge \text{Dec}_s(z) \notin \{\perp, m\}\right] \leq \text{neg}(k)$.
- p-***corrects***,if for every PPT algorithm A:
  $\mathbf{Pr}\left[\delta(\text{Enc}_s(m), z) \leq p \wedge \text{Dec}_s(z) \neq m\right] \leq \text{neg}(k)$.

Seeded codes with adaptive security, are strictly stronger than codes with selective security. However, they appear to be much harder to construct. Indeed, the result of [GHY20] relied on strong assumptions that we only know how to instantiate in the random oracle, and the results in [SW24] do not extend to the binary case (which is the main focus of this paper).

Remark 3.4. (The way we define error detection)**.** Our definition for error detection (both the selective notion and the adaptive notion) permits also error correction. That is, when errors are induced, the decoding algorithm can either return $\perp$ or successfully decode and return the original message m. By doing so we can achieve simultaneous error detection and correction (that is, the same decoding function can at the same time satisfy the notion of detection and correction defined above). We stress, however, that it is possible to achieve the standard notion of detection presented in Definition 2.5 (that always outputs $\perp$ if it receives a string z that is not a valid codeword). This is easily done, by first decoding $m = \text{Dec}_s(z)$ and outputting $\perp$ if $\text{Enc}_s(m) \neq z$.

4 Construction for Seeded Codes With Selective Security

We now state our main result for seeded codes with selective security. As discussed in the introduction, these codes achieve unique decoding with a rate that matches the Blokh-Zyablov bound R_{BZ} (see, Equation (1) for the exact bound) which is the rate of the best known explicit construction for list decodable codes. For error detection, we can achieve codes with rate that is close to 1, that can detect from almost $1/2 - \varepsilon$ errors.

Theorem 4.1 (Binary Seeded codes with selective security)**.** *If the LWE assumption holds: For every constant $0 \leq p < 1/4$ and every sufficiently small constant $0 < \varepsilon$,*

- *there exist binary seeded codes with selective security with rate $R(p) \geq R_{BZ}(p) - \varepsilon$ that corrects p relative errors.*
- *there exist binary seeded codes with selective security with rate $R \geq 1 - \varepsilon$ that detects $1/2 - \varepsilon$ relative errors.*

In Section 4.1, we define the ingredients that are used to prove Theorem 4.1. In Section 4.2, we present the construction and prove Theorem 4.1.

We mention that improvement in constructing explicit list-decodable codes, could be used to improve Theorem 4.1, as long as these codes have some special property which we refer to as "nested list-decodable codes". We define and explain this notion in Section 4.1.2.

Remark 4.2. If the underlying LWE assumption in our theorem statements have sub-exponential security, then our constructed codes in Theorem 4.1 will also have sub-exponential failure probability for correction against channels that are computable by sub-exponential size circuits. This follows since all our reductions run in polynomial time and have a polynomial loss in advantage. However, for the sake of simplicity, we focus on the case of polynomial/negligible security in our formal statements and proofs.

4.1 Ingredients for Our Construction

In this section, we define two ingredients that are specific to our seeded codes with selective security. In Section 4.1.1, we define and construct a new notion for correlation intractable hash functions. In Section 4.1.2 we define the list-decodable codes that we make use of in our construction.

4.1.1 Target 2-Input Correlation Intractable Hashing

We now define a new notion of 2-input correlation intractable hashing. Intuitively, 2-input (or multi-input for the general notion) correlation intractable (2CI) hash function h aims to provide "intractability" for relations of pairs $R(x, x', y, y')$. That is $s \leftarrow \{0,1\}^\ell$, then $x, x' \leftarrow \mathsf{A}(s)$ and it should hold that $\mathbf{Pr}\left[(x, x', h_s(x), h_s(x')) \in R\right] \leq \mathrm{neg}$. Very little is known about 2CI, even for limited relations. In particular, there are no explicit construction for for 2CI from standard hardness assumptions, even when considering correlations that are computed by circuits that run in a bounded time (as is done in Definition 2.3).

Below we define a variant for 2-input correlation intractable hashing which we call "target 2-input correlation intractable" (t-T2CI).

Definition 4.3. (Target 2-input correlation intractable hash functions (T2CI))**.** Let $k \in \mathbb{N}$ be a security parameter and $t \in \mathrm{poly}$ be a poly-time computable function. The hash family $\mathcal{H} = \{h_s \colon \{0,1\}^{n(k)} \to \{0,1\}^{m(k)}\}_{s \in \{0,1\}^{\ell(k)}}$ is ***target 2-input correlation intractable for t-size*** (denoted t-T2CI) if for every $L \in \mathrm{poly}$, $size(t(k))$ circuit $f = \{f_k : \{0,1\}^{n(k)} \times \{0,1\}^{n(k)} \times \{0,1\}^{m(k)} \to (\{0,1\}^{m(k)})^{L(k)}\}_{k \in \mathbb{N}}$ and every PPT A, if we consider the following randomized experiment:

- $(x, \text{state}) \leftarrow \mathsf{A}(1^k) \in \{0,1\}^{n(k)} \times \{0,1\}^*$,
- $s \leftarrow \{0,1\}^{\ell(k)}$,
- $x' \leftarrow \mathsf{A}(\text{state}, s)$

Then $\mathbf{Pr}\left[h_s(x') \in f_k(x', x, h_s(x)) \wedge x' \neq x\right] \leq \text{neg}(k)$.

It is instructive to compare the above notion of t-T2CI to the notion of t-CI defined in Definition 2.3. Informally, a t-CI h guarantees that for all circuits f of $size(t)$, and every PPT A when considering the following experiment: First sample $s \leftarrow \{0,1\}^\ell$ and then $x \leftarrow \mathsf{A}(s)$, it should hold that $\mathbf{Pr}\left[h_s(x) \in f(x)\right] \leq \text{neg}$. Similar to the above definition, we aim to give intractability against bounded size. Loosely speaking, t-T2CI satisfies the following experiment: First $x \leftarrow \mathsf{A}$ then we sample $s \leftarrow \{0,1\}^\ell$, after which $x' \leftarrow \mathsf{A}(x, s)$ and it holds that:

$$\mathbf{Pr}\left[h_s(x') \in f(x', x, h_s(x)) \wedge x' \neq x\right] \leq \text{neg}$$

Note that the above notion provides intractability for a form of relations that consists of two inputs. Specifically, it depends on x', x and $h_s(x)$. We call this a "target" relation since the first input x is selected independent of the seed s. This notion is weaker than the standard notion of 2-input correlation intractable hashing, however, as we show in this paper, this notion is sufficient for achieving seeded codes with selective security.

It is important to note that although it may initially appear that CI directly implies T2CI by defining $f'_{x,h_s(x)}(x') = f(x, x', h_s(x))$ (i.e., fixing the $x, h_s(x)$ into a circuit f using non-uniform advice), this approach does not work because the function f' cannot depend on the seed s and therefore cannot depend on $h_s(x)$. Still, our next theorem states that it is possible to construct a T2CI hash given a CI hash and a universal one-way hash function.

Theorem 4.4 (T2CI from CI and UOWHF)**.** *For every poly-time computable function $t \in \text{poly}$, given a t-CI family $\mathcal{H} = \{h_s \colon \{0,1\}^{n(k)} \to \{0,1\}^{m(k)}\}_{s \in \{0,1\}^{\ell(k)}}$ and a universal one-way hash function family $\hat{\mathcal{H}} = \{\hat{h}_{\hat{s}} \colon \{0,1\}^{n(k)} \to \{0,1\}^{\hat{m}(k)}\}_{\hat{s} \in \hat{m}(k)}$ there exists an explicit t-T2CI hash family $\mathcal{H}' = \{h'_{s'} \colon \{0,1\}^{n(k)} \to \{0,1\}^{m'(k)}\}_{s' \in \{0,1\}^{\ell'(k)}}$, where $\ell' = \hat{m} + 2\hat{m} \cdot \ell$ and $m' = \hat{m} \cdot m$.*

The following corollary directly follows from the above theorem.

Corollary 4.5. (T2CI from LWE)**.** If the LWE assumption holds, then there exists an $m \in \text{poly}$ such that for all poly-time computable functions $t, n \in \text{poly}$ there exists a poly-time computable $\ell \in \text{poly}$ and a t-T2CI hash family $\mathcal{H} = \{h_s \colon \{0,1\}^{n(k)} \to \{0,1\}^{m(k)}\}_{s \in \{0,1\}^{\ell(k)}}$.

Proof of Corollary 4.5. Follows immediately from Theorems 2.2, 2.4 and 4.4. □

We are now ready to prove Theorem 4.4. The main idea is that by making use of UOWHF we can create a t-CI hash function h that uses a "different" seed when it is queried on two different inputs. Intuitively, if after choosing $x \leftarrow \mathsf{A}$ and $s \leftarrow \{0,1\}^\ell$ and computing $h_s(x)$ when choosing $x' \leftarrow \mathsf{A}(x, s)$, instead of computing h on x' using the seed s, we instead where able to use a different seed s' that is independent from s to compute $h'_{s'}(x')$, then taking h to be just CI would be sufficient to get the security notion of T2CI. Indeed, the construction, described below formalizes this intuition by taking s to be a set of possible seeds for a CI, and then for every message that is being sent, a different subset is chosen using the UOWHF so that for two different messages, there will be at least one seed that is not shared, on which we can apply the above strategy. A formal proof is presented below.

Proving Theorem 4.4. We start by stating our t-T2CI construction. Given an

- t-CI family $\mathcal{H} = \{h_s \colon \{0,1\}^{n(k)} \to \{0,1\}^{m(k)}\}_{s \in \{0,1\}^{\ell(k)}}$, and
- a UOWHF family $\hat{\mathcal{H}} = \{\hat{h}_{\hat{s}} \colon \{0,1\}^{n(k)} \to \{0,1\}^{\hat{m}(k)}\}_{\hat{s} \in \hat{m}(k)}$

we define the t-T2CI hash family $\mathcal{H}' = \{h'_{s'} \colon \{0,1\}^{n(k)} \to \{0,1\}^{m'(k)}\}_{s' \in \{0,1\}^{\ell'(k)}}$, as follows. Take:

$$\ell' = \hat{m} + 2\hat{m} \cdot \ell, \qquad m' = \hat{m} \cdot m \tag{2}$$

Given an $s' \in \{0,1\}^{\ell'}$, we parse $s' = (\hat{s}, s^0_1, \cdots, s^0_{\hat{m}}, s^1_1, \cdots s^1_{\hat{m}})$ where $\hat{s} \in \{0,1\}^{\hat{m}}$ should be seen as a seed for the universal one-way hash function $\hat{h}$, and for every $b \in \{0,1\}$ and $i \in [\hat{m}]$ every $s^b_i \in \{0,1\}^{\ell}$ should be seen as a seed for a the t-CI hash. For every $x \in \{0,1\}^n$ and s', define $h'_{s'}(x)$ as follows:

- Compute $z = \hat{h}_{\hat{s}}(x)$, and recall that $z = (z_1, \ldots, z_{\hat{m}}) \in \{0,1\}^{\hat{m}}$.
- For every $i \in [\hat{m}]$, compute $y_i = h_{s^{z_i}_i}(x)$.
- Output $y = (y_1, \cdots, y_{\hat{m}})$.

We now prove that $\mathcal{H}'$ is an t-T2CI.

Proof of Theorem 4.4. Assume towards contradiction that $\mathcal{H}'$ described above is not a t-T2CI. That is, there exists a non-negligible δ, a $size(t(k))$ circuit $f' = \{f'_k\}_{k \in \mathbb{N}}$ and a PPT A such that, if we consider the following randomized experiment:

- $(x, \text{state}) \leftarrow \mathsf{A}(1^k) \in \{0,1\}^{n(k)} \times \{0,1\}^*$,
- $s' \leftarrow \{0,1\}^{\ell'(k)}$,
- $x' \leftarrow \mathsf{A}(\text{state}, s')$

then $\mathbf{Pr}\left[h'_{s'}(x') \in f'_k(x', x, h'_{s'}(x)) \wedge x \neq x'\right] \geq \delta(k)$. By the security of the UOWHF, we also have:

$$\mathbf{Pr}\left[h'_{s'}(x') \in f'_k(x', x, h'_{s'}(x)) \wedge x \neq x' \wedge \hat{h}_{\hat{s}}(x) \neq \hat{h}_{\hat{s}}(x')\right] \geq \delta(k) - \text{neg}(k).$$

Define the reduction A' that uses A to break t-CI as follows:

- Given a function f' for t-T2CI and a value x from A do the following. Sample $i^* \leftarrow [\hat{m}]$ uniformly at random. Sample $\hat{s} \leftarrow \{0,1\}^{\hat{m}}$. Define $z = \hat{h}_{\hat{s}}(x)$. Sample $\{s^b_i \leftarrow \{0,1\}^{\ell}\}$ for all $(i,b) \in ([\hat{m}] \times \{0,1\}) \setminus \{(i^*, 1 - z_{i^*})\}$. These values fully define $y = h'_{s'}(x)$. Define $f_{x,y,i^*}(x')$ to compute the list $\mathcal{L}' = f'(x', x, y)$ of values in $\{0,1\}^{m'} = (\{0,1\}^m)^{\hat{m}}$ and output the list $\mathcal{L}$ correspnding to the values in the i^* position of each item in the list $\mathcal{L}'$. Choose f_{x,y,i^*} as the function for t-CI.
- Receive a seed $s^* \leftarrow \{0,1\}^{\ell}$ from the challenger. Set $s^{1-z_{i^*}}_{i^*} := s^*$ and give $s' = (\hat{s}, \{s^b_i\}_{(i,b) \in [\hat{m}] \times \{0,1\}})$ to A.
- Receive x' from A and output it to the challenger.

Note that the seed s' is uniformly random just like in the t-T2CI game, and is independent of i^*. Therefore the entire view of A is independent of i^*. Let $Q = \{i \in [\hat{m}] \ : \ z_i \neq z'_i\}$ where $z = \hat{h}_{\hat{s}}(x), z' = \hat{h}_{\hat{s}}(x')$ in the context of the above experiment. Then we have:

$$\begin{aligned}
&\mathbf{Pr}\left[h'_{s'}(x') \in f'_k(x', x, h'_{s'}(x)) \wedge x \neq x' \wedge \hat{h}_{\hat{s}}(x) \neq \hat{h}_{\hat{s}}(x') \wedge i^* \in Q\right] \\
\geq &\mathbf{Pr}\left[h'_{s'}(x') \in f'_k(x', x, h'_{s'}(x)) \wedge x \neq x' \wedge \hat{h}_{\hat{s}}(x) \neq \hat{h}_{\hat{s}}(x')\right] \mathbf{Pr}\left[i^* \in Q \mid \hat{h}_{\hat{s}}(x) \neq \hat{h}_{\hat{s}}(x')\right] \\
\geq &(\delta(k) - \mathrm{neg}(k))/\hat{m}.
\end{aligned}$$

Moreover, whenever

$$h'_{s'}(x') \in f'_k(x', x, h'_{s'}(x)) \wedge x \neq x' \wedge \hat{h}_{\hat{s}}(x) \neq \hat{h}_{\hat{s}}(x') \wedge i^* \in Q$$

then A' wins the t-CI game. This is because when $h'_{s'}(x') \in f'(x', x, h'_{s'}(x))$ then $h_{s^*}(x') \in f_{x,y,i^*}(x')$. Therefore A' wins the t-CI game with non-negligible probability, which is a contradiction. □

4.1.2 Nested List-Decodable Codes Meeting The Blokh-Zyablov Bound

For the seeded code construction with selective security, we require as an ingredient a standard list-decodable code $(\mathrm{Enc}, \mathrm{Dec})$ from p-errors with special properties (which we refer to as nested list-decodable codes). Specifically, we require that this code has a sub-code $(\mathrm{Enc}_{\mathrm{nest}}, \mathrm{Dec}_{\mathrm{nest}})$ with the following property:

- $(\mathrm{Enc}_{\mathrm{nest}}, \mathrm{Dec}_{\mathrm{nest}})$ is also list-decodable form p_{nest} errors,
- encoding and decoding (for both codes) is efficient.

Intuitively, this sub-code when viewed on its own can have a much smaller rate, but as a consequence of this, we could expect p_{nest} to be much larger. Indeed, for our application, we require a code with $p_{\mathrm{nest}} \approx 1/2$, even if the rate of the nested code is also some constant that approaches zero.

Looking ahead, this sub-code will be applied on the hash of the message (using the T2CI hash defined in Section 4.1.1). As described in the introduction, the fact that the sub-code can recover from a higher fraction of errors will be crucial for the proof.

Definition 4.6. For a polytime computable function $v \colon \mathbb{N} \to \mathbb{N}$ and constants $0 \leq p, p_{\mathrm{nest}} \leq 1/2$, we say that a standard linear code family $(\mathrm{Enc}, \mathrm{Dec})$ is $(v, p, p_{\mathrm{nest}})$***-nested list decodable*** if

1. Enc is given as a generating matrix $A \in \mathbb{F}_2^{(k+v)\times n}$ and the code is (p, L)-list decodable for some $L \in \mathrm{poly}$.
2. The subcode $(\mathrm{Enc}_{\mathrm{nest}}, \mathrm{Dec}_{\mathrm{nest}})$ given as a generating matrix $\mathbf{A}_{\mathrm{nest}} \in \mathbb{F}_2^{v\times n}$ defined to be the last v rows in A is $(p_{\mathrm{nest}}, L_{\mathrm{nest}})$-list decodable for some $L_{\mathrm{nest}} \in \mathrm{poly}$.

The code is said to be explicit if both $(\mathrm{Enc}, \mathrm{Dec})$ and $(\mathrm{Enc}_{\mathrm{nest}}, \mathrm{Dec}_{\mathrm{nest}})$ run in polytime. (That is, the encoding and decoding for both the overall code and sub-code run in polytime).

We show that the best known construction for list-decoding by Guruswami and Rudra [GR08a] (that achieve the Blokh-Zyablov bound), when viewed in a particular order, are in fact nested list decodable codes that satisfy Definition 4.6. This is stated formally in the following theorem:

Theorem 4.7 (Nested linear list decodable code)**.** *For every $0 \leq p < 1/4$ and every sufficiently small constants $0 < \varepsilon, \gamma$ and $\beta < 1$, there exists an explicit linear $(v(k) := k^{\beta}, p, p_{\text{nest}} = 1/2 - \gamma)$-nested list decodable code with a rate $R > R_{BZ}(p) - \varepsilon$.*[7]

The proof for Theorem 4.7 appears in Appendix A. We stress that if in the future, better list-decodable codes that have (the nesting property) are constructed, then these codes would immediately yield an improvement in our results. In particular, a random linear code achieves optimal parameters and will have the above nesting property (but it does not have efficient decoding). However, as of now, the best explicit construction for list-decodable remains the construction of Guruswami and Rudra [GR08a] (that achieves the Blokh-Zyablov bound).

In the next section, we show that using codes with nested list decoding and target 2-input correlation intractable hashing we can get a seeded code that is selectively secure.

4.2 Proof of Theorem 4.1.

We describe the construction that we will use to prove Theorem 4.1. In Figure 1 we describe the encoding and decoding algorithms as well as the ingredients required for the construction.

We will use the following technical lemma to prove Theorem 4.1:

Lemma 4.8. *Assume there are ingredients and parameters that satisfy the conditions stated in Figure 1. Then, (Enc, Dec), specified in Figure 1 is a binary seeded code with selective security that simultaneously,*

- *corrects from p errors, and*
- *detects from $p_{\text{nest}} - p$ errors.*

Moreover, the code has rate $R = R' - v/n$.

Remark 4.9. The statement of Lemma 4.8 gives a code with a single decoding algorithm that can simultaneously correct and detect errors. Indeed, this is done so that we are able to give a single unified construction for detection and correction. We emphasize however, that when interested only in error detection, the value of p (the decoding radius of the list-decoding code $(\text{Enc}', \text{Dec}')$) can be taken to be zero, which by Lemma 4.8 implies that we get p_{nest} error detection. Moreover, things become a lot simpler in this setting.

Indeed, when $p = 0$, the decoding algorithm Dec' simply checks if the input is a valid codeword that has no noise, if so, it outputs its corresponding message. That is, this amounts to a simple detection algorithm (and list-decoding is not required). It turns out, that achieving the nesting property only for error detection is a much easier task. This is because we can simply start from any small "subcode" with a generating matrix $\mathbf{A}_{\text{nest}}$, that can list-decode from almost $p_{\text{nest}} \approx 1/2$ errors and has a rate close to zero (such codes are very easy to construct explicitly), and then expand $\mathbf{A}_{\text{nest}}$ to a full matrix $\mathbf{A}$ by adding independent rows to $\mathbf{A}_{\text{nest}}$. The end result $\mathbf{A}$ is a bijective linear map that is slightly expanding (that is, the overall rate of $\mathbf{A}$ is close to 1) and easy to invert. Crucially, the list-decoding property of the nested code $\mathbf{A}_{\text{nest}}$ is not affected and remains $p_{\text{nest}} \approx 1/2$, which by Lemma 4.8 means that the overall new codes that we construct can also detect from almost $1/2$ errors and has rate almost 1. This corresponds to the construction for selectively secure error detection codes presented in Section 1.2.

[7] A careful examination of the proof Theorem 4.7, which we give in Appendix A, shows that that it is possible to achieve this rate for every sufficiently large $k \in \mathbb{N}$ (and not just for a family of codes). However, we state our theorem in a manner that is consistent with the error-correcting codes literature.

Figure 1: Encoding and Decoding algorithms.

Ingredients and parameters: Let $v, \ell, t, n, L : \mathbb{N} \mapsto \mathbb{N}$ be poly-time computable functions, and let $0 \leq p < 1/4$, $0 \leq R' \leq 1$ and $0 < \gamma$ be a constant.

- Nested list decodable code: Let $(\mathrm{Enc}', \mathrm{Dec}')$ be a $(v, p, p_{\mathrm{nest}})$-nested list-decodable code with codeword length n and rate R', where the decoding algorithm $\mathrm{Dec}_{\mathrm{nest}}$ of the subcode $(\mathrm{Enc}_{\mathrm{nest}}, \mathrm{Dec}_{\mathrm{nest}})$ runs in at most t_{nest} time for some $t_{\mathrm{nest}} \in \mathrm{poly}$. We require that $2 \cdot p < p_{\mathrm{nest}}$.
- A target 2-input correlation intractable hash family: t-T2CI $\mathcal{H} = \{h_s \colon \{0,1\}^k \to \{0,1\}^v\}_{s \in \{0,1\}^\ell}$ that is efficiently computable. We require that $t > t_{\mathrm{nest}} + k^4$.

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

Input: A message $m \in \{0,1\}^k$ and a seed $s \in \{0,1\}^\ell$.

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

Operation :

- Hash the message: Compute $y = h_s(m)$.
- Encode the message and the hash: Output $c = (m, y) \cdot A = \mathrm{Enc}'(m, y)$, (where $A \in \mathbb{F}_2^{(k+v)\times n}$ is the be generating matrix of the nested list decodable code).

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

Input: A received word $z \in \{0,1\}^n$ and a seed $s \in \{0,1\}^\ell$.

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

Operation :

- List decoding the received word: Compute a list of message candidates $\mathrm{List} = \mathrm{Dec}'(z)$ (that is, list-decode from p-relative errors).
- Check consistency: Out of all $(\widehat{m}, \widehat{y}) \in \mathrm{List}$ for which $h_s(\widehat{m}) = \widehat{y}$ and $\delta(\mathrm{Enc}'(\widehat{m}, \widehat{y}), z) \leq p$, output $\widehat{m}$ for which $\mathrm{Enc}'(\widehat{m}, \widehat{y})$ is the closest to z in hamming distance (breaking ties arbitrary). If no such message exists output $\perp$.

We prove Lemma 4.8 below in Section 4.2.1. But first, we use the lemma to prove our main theorem for seeded codes with selective security.

Proof of Theorem 4.1. Given a sufficiently small $0 < \varepsilon$, let $\hat{\varepsilon} = \varepsilon/2$, $v := k^{1/2}$ and $p_{\mathrm{nest}} = 1/2 - \gamma$ for a constant $\gamma < min\{1/2 - 2p, \hat{\varepsilon}\}$. By Theorem 4.7 there exists a $(v, p, p_{\mathrm{nest}})$-nested code $(\mathrm{Enc}', \mathrm{Dec}')$ with rate $R'(p) > R_{BZ}(p) - \hat{\varepsilon}$. Moreover, the subcode $(\mathrm{Enc}_{\mathrm{nest}}, \mathrm{Dec}_{\mathrm{nest}})$ runs in at most t_{nest} time.

By Corollary 4.5 if the LWE assumption holds then there exists a t-T2CI hash family $\mathcal{H} = \{h_s \colon \{0,1\}^k \to \{0,1\}^v\}_{s \in \{0,1\}^\ell}$. Where it holds that $t > t_{\mathrm{nest}} + k^4$.

The above ingredients satisfy the conditions stated in Figure 1, thus, by Lemma 4.8 the construction described in Figure 1 is a seeded code with selective security that corrects p relative errors and has rate

$R = R' - v/n \geq R_{BZ}(p) - \hat{\varepsilon} - k^{1/2}/n \geq R_{BZ}(p) - \varepsilon$.

To get our error detection result, that is, detecting $1/2 - \varepsilon$ errors, note that by Lemma 4.8 our seeded code can detect $p_{\text{nest}} - p$ errors. Thus, by taking the rate of $(\text{Enc}', \text{Dec}')$ to be $1 - \hat{\varepsilon}$ which implies that the correction radius p goes to zero (in fact, we can even set $p = 0$, which amounts to only require detection from $(\text{Enc}', \text{Dec}')$, see Remark 4.9) implies that the code can detect from $p_{\text{nest}} = 1/2 - \gamma \geq 1/2 - \varepsilon$ as desired. □

4.2.1 Proof of Lemma 4.8.

We now prove Lemma 4.8. By assumption the nested list decodable code $(\text{Enc}', \text{Dec}')$ has $\text{Enc}' \colon \{0,1\}^k \times \{0,1\}^v \to \{0,1\}^n$ that is given as the generating matrix $\mathbf{A} \in \mathbb{F}_2^{(k+v)\times n}$. Where $\mathbf{A}_{\text{nest}} \in \mathbb{F}_2^{v\times n}$ defined to be the last v rows of $\mathbf{A}$ is the generating matrix for the subcode $(\text{Enc}'_{\text{nest}}, \text{Dec}'_{\text{nest}})$ and $\text{Dec}'_{\text{nest}}$ runs in at most $t_{\text{nest}}(k)$ for some fixed $t_{\text{nest}} \in \text{poly}$. And $\mathcal{H} = \{h_s \colon \{0,1\}^k \to \{0,1\}^v\}_{s\in\{0,1\}^\ell}$ is t-T2CI hash family for $t > t_{\text{nest}} + k^4$.

First note, that by construction, $R = \frac{k}{n} = R' - v/n$ since $R' = \frac{k+v}{n}$, and that for every message $m \in \{0,1\}^k$ and seed $s \in \{0,1\}^\ell$: $\text{Dec}_s(\text{Enc}_s(m)) = m$.

The main idea is that to prove both the detection and correction properties, we will reduce to the correctness of the T2CI hash. We will use the following claim which gives an efficient function f which we will later use to contradict the T2CI hash.

Claim 4.10. There exists a function $f : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^v \to (\{0,1\}^v)^L$ for some $L \in \text{poly}$ that runs in at most $t_{\text{nest}}+k^4$ time, such that, for every (m, y) and $(\hat{m}, \hat{y})$ over $\{0,1\}^{k+v}$ if $\delta(\text{Enc}'(m, y), \text{Enc}'(\hat{m}, \hat{y})) \leq p_{\text{nest}}$ then $\hat{y} \in f(\hat{m}, m, y)$.

Proof of Claim 4.10. Recall that $\text{Enc}'(m, y) = (m, y) \cdot \mathbf{A}$ and $\text{Enc}'(\hat{m}, \hat{y}) = (\hat{m}, \hat{y}) \cdot \mathbf{A}$ where $\mathbf{A} \in \mathbb{F}_2^{(k+v)\times n}$ is the generating matrix of Enc'. Recall also that $(\text{Enc}', \text{Dec}')$ is a nested code with a subcode $(\text{Enc}_{\text{nest}}, \text{Dec}_{\text{nest}})$ given by generating the matrix $\mathbf{A}_{\text{nest}} \in \mathbb{F}_2^{v\times n}$ defined to be the last v rows of $\mathbf{A}$. Define the matrix $\mathbf{B} \in \mathbb{F}_2^{k\times n}$ to be the first k rows in $\mathbf{A}$ (that is, $\mathbf{A} = \begin{bmatrix} \mathbf{B} \\ \mathbf{A}_{\text{nest}} \end{bmatrix}$), and note that by linearity for every $(m, y) \in \{0,1\}^{k+v}$:

$$\begin{aligned} \text{Enc}'(m, y) &= (m, y) \cdot \mathbf{A} \\ &= (m, y) \cdot \begin{bmatrix} \mathbf{B} \\ \mathbf{A}_{\text{nest}} \end{bmatrix} \\ &= m \cdot \mathbf{B} + y \cdot \mathbf{A}_{\text{nest}}. \end{aligned}$$

And again, by linearity if $\delta(\text{Enc}'(m, y), \text{Enc}'(\hat{m}, \hat{y})) \leq p_{\text{nest}}$ then it follows that

$$\begin{aligned} \delta(\text{Enc}'(m, y), \text{Enc}'(\hat{m}, \hat{y})) &= \delta(m \cdot \mathbf{B} + y \cdot \mathbf{A}_{\text{nest}} \,,\, \hat{m} \cdot \mathbf{B} + \hat{y} \cdot \mathbf{A}_{\text{nest}}) \\ &= \delta(\hat{y} \cdot \mathbf{A}_{\text{nest}} \,,\, \hat{m} \cdot \mathbf{B} + m \cdot \mathbf{B} + y \cdot \mathbf{A}_{\text{nest}}) \leq p_{\text{nest}} \end{aligned}$$

Since $\mathbf{A}_{\text{nest}}$ is list-decodable from p_{nest} error and $\hat{y} \cdot \mathbf{A}_{\text{nest}}$ is a codeword of Enc_{nest}, it follows that

$$\hat{y} \in \text{Dec}_{\text{nest}}(\hat{m} \cdot \mathbf{B} + m \cdot \mathbf{B} + y \cdot \mathbf{A}_{\text{nest}}).$$

Thus, setting $f(\hat{m}, m, y) \coloneqq \text{Dec}_{\text{nest}}(\hat{m} \cdot \mathbf{B} + m \cdot \mathbf{B} + y \cdot \mathbf{A}_{\text{nest}})$ concludes the proof since f can be computed in less than $t_{\text{nest}} + k^4$ times. □

We now prove the (selectively secure) decoding/correcting properties, in both cases we reduce to the correctness of the t-T2CI property of the hash. For a PPT adversary A and $k \in \mathbb{N}$ consider the following experiment:

1. $(m, \text{state}) \leftarrow \mathsf{A}(1^k) \in \{0,1\}^k \times \{0,1\}^*$
2. Sampling $s \leftarrow \{0,1\}^\ell$
3. $z \leftarrow \mathsf{A}(m, \text{state}, s)$

Let $y = h_s(m)$, $\hat{m} = \mathrm{Dec}_s(z)$ and $\hat{y} = h_s(\hat{m})$, and let $\mathrm{List} = \mathrm{Dec}'(z)$. Recall also that $\mathrm{Enc}_s(m) = \mathrm{Enc}'(m, y)$ and $\mathrm{Enc}_s(\hat{m}) = \mathrm{Enc}'(\hat{m}, \hat{y})$. We first prove the correction property and then the detection property.

Error correction: To prove the correction property, assume towards contradiction that there exists A and a non-negligible function γ such that:

$$\mathbf{Pr}\left[\delta(\mathrm{Enc}_s(m), z) \leq p \wedge \hat{m} \neq m\right] \geq \gamma.$$

Assuming $\delta(\mathrm{Enc}_s(m), z) \leq p$, it follows by construction that $(m, y) \in \mathrm{List}$ which implies that $\hat{m} \neq \perp$. Since $\hat{m} \neq \perp$, the "check consistency" step in the decoding algorithm implies that $\delta(\mathrm{Enc}'(\hat{m}, \hat{y}), z) \leq p$. Thus, by the triangle inequality:

$$\mathbf{Pr}\left[\delta(\mathrm{Enc}'(m, y), \mathrm{Enc}'(\hat{m}, \hat{y})) \leq 2p \wedge m \neq \hat{m}\right] \geq \gamma$$

Since by our choice of parameters $2p \leq p_{\text{nest}}$, by Claim 4.10 there exists a function f that runs in at most $t_{\text{nest}} + k^4$ time such that:

$$\mathbf{Pr}\left[\hat{y} \in f(\hat{m}, m, y) \wedge m \neq \hat{m}\right] \geq \gamma.$$

Since $y = h_s(m)$ and $\hat{y} = h_s(\hat{m})$, and $t > t_{\text{nest}} + k^4$, the function f contradicts the t-T2CI property of $\mathcal{H}$. This concludes the proof for the correction property.

Error detection: To prove detection property, assume towards contradiction that there exists A and a non-negligible function γ such that:

$$\mathbf{Pr}\left[\delta(\mathrm{Enc}_s(m), z) \leq p_{\text{nest}} - p \wedge \hat{m} \neq \{m, \perp\}\right] \geq \gamma.$$

Similar to the correction case, assuming that $\delta(\mathrm{Enc}_s(m), z) \leq p_{\text{nest}} - p$ and $\hat{m} \neq \perp$, the "check consistency" step in the decoding algorithm implies that $\delta(\mathrm{Enc}'(\hat{m}, \hat{y}), z) \leq p$. Thus, by the triangle inequality:

$$\mathbf{Pr}\left[\delta(\mathrm{Enc}'(m, y), \mathrm{Enc}'(\hat{m}, \hat{y})) \leq p_{\text{nest}} \wedge m \neq \hat{m}\right] \geq \gamma$$

The rest of the proof follows along the same line as the correction property by applying Claim 4.10 and contradicting the t-T2CI property of $\mathcal{H}$.

5 Construction for Seeded Codes With Adaptive Security

In this section, we present our construction for seeded codes with adaptive security. To do so, we propose a new hardness assumption based on the cryptographic dark matter paradigm.

The dark matter hash assumption. Boneh, Ishai, Passelègue, Sahai, and Wu [BIP+18], proposed basing cryptographic hardness on what they referred to as "crypto dark matter". Specifically, they proposed using the idea of mixing linear functions over different moduli as a source of hardness for cryptographic primitives. Inspired by their work, we propose a concrete collision resistance hash function based on this paradigm, which we refer to as "the dark matter hash assumption".

We use the following function $\mathsf{map}_{a\to b}$ that changes the bases of representation of integer and which we will use in defining our concrete hash assumption.

Definition 5.1. ($\mathsf{map}_{a\to b}$)**.** For every $a, b \in \mathbb{N}$, we define the function $\mathsf{map}_{a\to b} : \mathbb{Z}_a^n \to \mathbb{Z}_b^{\lceil (\log_b a)n \rceil}$ to be natural mapping from base a to base b. That is, given $x \in \mathbb{Z}_a^n$, $y = \mathsf{map}_{a\to b}(x)$ where $y \in \mathbb{Z}_b^{\lceil (\log_b a)n \rceil}$ is obtained by interpreting x as an integer of size at most a^n and re-writing it in base b.

Assumption 5.2. (The Dark Matter Hash Assumption)**.** Let $k \in \mathbb{N}$ be a parameter and $0 < \delta$ be a constant and define $\tilde{k} = \lceil (\log_3 2)k \rceil$, $v = (1+\delta)\tilde{k}$ and $\tilde{v} = \lceil (\log_2 3)v \rceil$. For every sufficiently small constant $0 < \beta < \delta$, and every PPT A, when considering the following experiment:

- Sample $\mathbf{B} \leftarrow \mathbb{F}_3^{\tilde{k}\times v}$ and $\mathbf{A} \leftarrow \mathbb{F}_2^{\tilde{v}\times \beta k}$ and set $s \coloneqq (\mathbf{A}, \mathbf{B})$.
- $(x, x') \leftarrow \mathsf{A}(s)$

It holds that

$$\mathbf{Pr}\left[x \neq x' \wedge h_s(x) = h_s(x')\right] \leq \left(\frac{1}{2}\right)^{\beta k(1-o(1))}$$

Where we define $h_s : \{0,1\}^k \to \{0,1\}^{\beta k}$ via $h_s(x) \coloneqq \mathsf{map}_{3\to 2}(\mathsf{map}_{2\to 3}(x)\cdot \mathbf{B})\cdot \mathbf{A}$.

The above assumption says that any polytime adversary will not do much better than a simple guessing attack. On a high level, although it is easy to find "collisions" in $\mathbf{A}$, that is, find $\tilde{y}, \tilde{y}'$ such that $\tilde{y}\cdot\mathbf{A} = \tilde{y}'\cdot\mathbf{A}$ (by using simple linear algebra) it appears to be difficult to find $\tilde{y}, \tilde{y}'$ for which $y = \mathsf{map}_{3\to 2}^{-1}(\tilde{y})$ and $y' = \mathsf{map}_{3\to 2}^{-1}(\tilde{y}')$ both lie in the row-span of $\mathbf{B}$ over $\mathbb{F}_3$. For this intuition to be sound, the row-span of $\mathbf{B}$ needs to be sparse, as otherwise a simple guessing attack could be used to find such a collision. Indeed, by taking $\mathbf{B} \in F_3^{\tilde{k}\times v}$ to have a constant expansion (that is, $v = (1+\delta)\tilde{k}$), the row-span of $\mathbf{B}$ is indeed sparse which mitigates the previous trivial attack.[8] For the sake of consistency with the error-correcting literature, we consider constant shrinking ratio for our hash function. However, it makes sense to consider much bigger shrinkage (say $1/\sqrt{k}$ fraction), and in such a case, the expansion of the matrix $\mathbf{B}$ can also be much smaller (that is, we can take, δ to be smaller).

Note also that our above assumption could be generalized to have several levels of moduli switching. That is, instead of having two matrices we could have 2ℓ matrices $\mathbf{A}_1, \cdots \mathbf{A}_\ell$ and $\mathbf{B}_1 \cdots \mathbf{B}_\ell$ where $\mathbf{A}$'s and $\mathbf{B}$'s are over different fields, where we apply them sequentially alternating between the $\mathbf{A}$'s and $\mathbf{B}$'s such that they are slightly expanding, except the last matrix applied which will be shrinking. The benefit to this generalization is that even if it turns out to be possible to find some collision with non-trivial probability in Assumption 5.2, it could still be hard to do so in the generalized assumption.[9] We mention, that our seeded codes construction with adaptive security, could also be constructed from this generalized assumption, however, for the sake of simplicity we state our results with respect to Assumption 5.2.

[8] Note that since $\beta < \delta$ the probability that a random vector in $\mathbb{F}_3^v$ is in the row-span of $\mathbf{B}$ is smaller than the trivial collision finding probably for h, since the output size of h is at most βk bits.

[9] Indeed, a good intuition to keep in mind is the AES construction, that achieves or amplifies hardness by iteratively applying alternating linear and non-linear operations (which is similar to our generalized assumption since switching the moduli is a non-linear operation).

Seeded codes with adaptive security from dark-matter assumptions. We now state our main theorems for seeded codes with adaptive security. We start with our results separately for detection and correction.

Theorem 5.3. *If Assumption 5.2 holds, then for every constant $0 < \varepsilon$, there exists a seeded code $(\mathrm{Enc}, \mathrm{Dec})$ with adaptive security that has rate $R \geq 1 - \varepsilon$ and detects from $p = 1/2 - \varepsilon$ fraction of errors.*

The proof for Theorem 5.3 is given in Section 5.2 and makes use of the results and notions defined in Section 5.1.

Theorem 5.4. *If Assumption 5.2 holds, then given a standard code that list-decodes from p errors with rate R_{List}, for every constant $0 < \varepsilon$ there exists a seeded code $(\mathrm{Enc}, \mathrm{Dec})$ with adaptive security and $R \geq R_{\mathrm{List}}(1 - \varepsilon)$, that can* **uniquely** *correct from p errors as long as $p < (1/4 - \varepsilon)R_{\mathrm{List}}$.*

The proof for Theorem 5.4 is given in Section 5.1.1.

Applying Theorem 5.4 with Theorem 5.3 and using the Guruswami and Rudra code [GR08a] that achieves the Blokh-Zyablov bound, we get the following corollary:

Corollary 5.5. If Assumption 5.2 holds, for every $0 < \varepsilon$ there exists a seeded code with adaptive security that corrects from p errors and has rate $R_{BZ}(p) - \varepsilon$, as long as $p/R_{BZ}(p) \leq 1/2 - \varepsilon$.

Moreover, we stress that any improvement in constructing list-decodable codes will yield an improved seeded code with adaptive security (for $p < R_{\mathrm{List}}/4$).

5.1 Everywhere Collision Resistance

In this section, we define our notion of everywhere collision resistance functions (ECR). In Section 5.1.1 we show how this notion is related to seeded codes with adaptive security.

Intuitively, a ECR is a family of seeded functions that are bijective and efficiently invertible (in particular, this implies that the output is at least as large as the input). An everywhere collision resistance functions $\mathrm{Enc}_s \colon \{0,1\}^k \to \{0,1\}^n$ provide the following cryptography guarantee: For a parameter $\alpha > 0$, it must hold that for every set $T \geq \alpha n$, it is hard to find to inputs $x \neq x'$ that collide on the output of Enc_s restricting only to the set T (that is, with all but negligible probability $\mathrm{Enc}_s(x)_T \neq \mathrm{Enc}_s(x')_T$). In particular, this means that when $\alpha n < k$, ECR trivially implies standard collision resistance functions. We make this formal in the following definition:

Definition 5.6. (Everywhere Collision Resistance)**.** For a constant $0 < \alpha$, a family of efficiently computable and invertible bijective functions $\mathrm{Enc} = \{\mathrm{Enc}_s \colon \{0,1\}^k \to \{0,1\}^{n(k)}\}_{s \in \ell(k)}$ is an ***everywhere collision resistance for relative set size*** α (denoted (α, μ)-ECR) if for every A and sufficiently large $k \in \mathbb{N}$ and every subset $T \subseteq [n(k)]$, such that $|T| \geq \alpha \cdot n(k)$:

$$\Pr_{\substack{s \leftarrow \{0,1\}^{\ell(k)}, \\ x, x' \leftarrow \mathsf{A}(s)}} \left[x \neq x' \wedge \mathrm{Enc}_s(x)|_T = \mathrm{Enc}_s(x')|_T \right] \leq \mu(k).$$

Where for every $x \in \{0,1\}^k$, $\mathrm{Enc}_s(x)|_T$ is the output of $\mathrm{Enc}_s(x)$ restricted to the set of indexes in T. We use Dec to denote the inverse function to Enc, such that for $s \in \{0,1\}^\ell$, $\mathrm{Dec}_s \colon \{0,1\}^n \to \{0,1\}^k$ is defined as follows: On input $z \in \{0,1\}^n$, output $x \in \{0,1\}^k$ such that $\mathrm{Enc}(x) = z$, if no such x exists outputs $\perp$.

We will show, that ECR functions are closely linked to error correction and detection against computationally bounded adversaries.

However, to the best of our knowledge, everywhere collision resistance functions are a natural new cryptographic primitives, that might be useful for achieving other cryptographic tasks.

Everywhere collision resistance from crypto dark matter. We will now show how to construct an everywhere collision resistance function from our proposed dark matter hash assumption (see, Assumption 5.2). Recall that we define collision resistance hash $h_s(x) \coloneqq \mathsf{map}_{3\to 2}(\mathsf{map}_{2\to 3}(x)\cdot \mathbf{B})\cdot \mathbf{A}$, where the matrix $\mathbf{B}$ is slightly expanding and the matrix $\mathbf{A}$ is shrinking. The main idea is that we can replace the matrix $\mathbf{A}$ with a matrix $\mathbf{G}$ that is slightly expanding so that the new function $f_s(x) \coloneqq \mathsf{map}_{3\to 2}(\mathsf{map}_{2\to 3}(x)\cdot \mathbf{B})\cdot \mathbf{G}$ is no longer shrinking (and in fact, it is invertible with a very high probability). The main observation is that any restriction of the output of f_s to a subset T (that is sufficiently shrinking) is equivalent to the collision resistance hash function h_s (that is $f_s(x)_T = h_s(x)$) when $\mathbf{A}$ and $\mathbf{G}$ are sampled at random. Since this is true for any subset T, it implies that f_s is an everywhere collision resistance function as desired. One subtlety that we need to account for is that since $\mathbf{G}$ and $\mathbf{B}$ are sampled at random, it might be the case $\mathbf{G}$ or $\mathbf{B}$ are not invertible (which would contradict the definition of ECR). But since this could be easily checked and it happens with a very small probability, we can define the output of the ECR to be any efficiently invertible function whenever $\mathbf{G}$ or $\mathbf{B}$ don't have full rank. We make this formal in the following theorem.

Theorem 5.7 (ECR from crypto dark matter)**.** *If Assumption 5.2 holds, then for every constants $0 < \gamma, \varepsilon$, and every sufficiently small constant $0 < \alpha$, there exists an $(\alpha, (1/2)^{\alpha n(1-\varepsilon)})$-ECR family Enc with output size n and rate $k/n \geq 1-\gamma$.*

Proving Theorem 5.7. We start by describing the construction for the ECR family. Let $0 < \gamma, \varepsilon$ be constants that are given, and define the constant $0 < \delta$ so that $\frac{1}{(1+\delta)^2} > 1-\gamma$ (that is, $\delta < -1 + \frac{1}{\sqrt{1-\gamma}}$). We define $\mathrm{Enc} = \{\mathrm{Enc}_s \colon \{0,1\}^k \to \{0,1\}^n\}_{s\in\ell(k)}$ as follows: For $k \in \mathbb{N}$, set $\tilde{k} \coloneqq \lceil (\log_3 2)k \rceil$, $v \coloneqq (1+\delta)\tilde{k}$ and $\tilde{v} \coloneqq \lceil (\log_2 3)v \rceil$ and output size $n \coloneqq (1+\delta)\cdot\tilde{v} \leq \frac{k}{1-\gamma}$.

Given a string $s \in \{0,1\}^\ell$, parse $s = (\mathbf{G}, \mathbf{B})$ where $\mathbf{B} \in \mathbb{F}_3^{\tilde{k}\times v}$ and $\mathbf{G} \in \mathbb{F}_2^{\tilde{v}\times n}$. For every $x \in \{0,1\}^k$ and s, we define $\mathrm{Enc}_s(x)$ as follows:

- Given $s = (\mathbf{G}, \mathbf{B})$ check that $\mathbf{G}$ and $\mathbf{B}$ have full rank. If not, define $\mathrm{Enc}_s(x) = (x, 0^{n-k})$.
- Otherwise, define $\mathrm{Enc}_s(x) = f_s(x)$, for $f_s \coloneqq \mathsf{map}_{3\to 2}(\mathsf{map}_{2\to 3}(x)\cdot \mathbf{B})\cdot \mathbf{G}$.

We will now prove that for every sufficiently small $0 < \alpha$, Enc is an $(\alpha, (1/2)^{\alpha n(1-\varepsilon)})$-ECR.

Proof of Theorem 5.7. We will show that under our "dark matter hash" assume (Assumption 5.2), for every sufficiently small constant $0 < \alpha$, it holds that Enc is a $(\alpha, (1/2)^{\alpha n(1-\varepsilon)})$-ECR.

First now that by definition, for every $s = (\mathbf{G}, \mathbf{B})$, Enc_s is an efficiently invertible function. To see why this holds consider two cases: If $\mathbf{G}$ and $\mathbf{B}$ don't have full rank, then $\mathrm{Enc}_s(x) = (x, 0^{n-k})$ which is efficiently invertible. Otherwise, $\mathrm{Enc}_s(x) = f(x)$, and $f(x) = \mathsf{map}_{3\to 2}(\mathsf{map}_{2\to 3}(x)\cdot \mathbf{B})\cdot \mathbf{G}$ is also invertible since $\mathbf{G}$ and $\mathbf{B}$ both have full rank. Specifically, it holds that given $c \in \{0,1\}^n$ we can efficiently find $\tilde{y} \in \mathbb{F}_2^{\tilde{v}}$ such that $c = \tilde{y}\cdot\mathbf{G}$ if such a $\tilde{y}$ exists and output $\perp$ otherwise, and similarly for the matrix $\mathbf{B}$. Moreover, $\mathsf{map}_{2\to 3}$ and $\mathsf{map}_{3\to 2}$ are clearly efficiently invertible. Thus, the overall construction of Enc is efficiently invertible as desired.

We now prove the everywhere collision resistance property by directly reducing to Assumption 5.2.

For every sufficiently small constant $0 < \alpha < \delta$ (which we specify later), every sufficiently large $k \in \mathbb{N}$ and every $|T| \geq \alpha n$, we will show that:

$$\Pr_{\substack{s\leftarrow\{0,1\}^{\ell(k)},\\ x,x'\leftarrow \mathsf{A}(s)}} \left[x \neq x' \wedge \mathrm{Enc}_s(x)|_T = \mathrm{Enc}_s(x')|_T\right] \leq (1/2)^{\alpha n(1-\varepsilon)}.$$

Let E denote the event that $\mathbf{G}$ and $\mathbf{B}$ have full rank. And note that by definition, if E holds then $f_s(x) = \mathrm{Enc}_s(x)$ for every x. The main observation now is that for every x,

$$f_s(x)_T \coloneqq h_s(x) = \mathsf{map}_{3\to 2}(\mathsf{map}_{2\to 3}(x)\cdot \mathbf{B})\cdot \mathbf{A}$$

where $\mathbf{A} \in \mathbb{F}_2^{\tilde{v}\times|T|}$ is a sub matrix of $\mathbf{G}$ obtained by restricting $\mathbf{G}$ only to the rows in T. Moreover, this is the exact setup considered in Assumption 5.2 for the function h_s.

It thus follows that,

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{\substack{s\leftarrow\{0,1\}^{\ell(k)},\\ x,x'\leftarrow \mathsf{A}(s)}}\left[x\neq x' \wedge \mathrm{Enc}_s(x)|_T \leq \mathrm{Enc}_s(x')|_T\right] &\leq \mathop{\mathbf{Pr}}_{\substack{s\leftarrow\{0,1\}^{\ell(k)},\\ x,x'\leftarrow \mathsf{A}(s)}}\left[x\neq x' \wedge f_s(x)|_T = f_s(x')|_T \wedge E\right] \\
&\leq \mathop{\mathbf{Pr}}_{\substack{s\leftarrow\{0,1\}^{\ell(k)},\\ x,x'\leftarrow \mathsf{A}(s)}}\left[x\neq x' \wedge h_s(x) = h_s(x')\right] - 2^{0.9k\delta} \quad (3)\\
&\leq (1/2)^{\alpha n(1-o(1))} - 2^{0.9k\delta} \quad (4)\\
&\leq (1/2)^{\alpha n(1-\varepsilon)} \quad (5)
\end{aligned}$$

Where equation Equation (3) follows since $\mathbf{Pr}[E] \geq 1 - 2^{-0.9\delta\cdot k}$ (which we prove below in Claim 5.8), and Equation (4) and Equation (5) follow by applying Assumption 5.2 and taking a sufficiently small α. □

Proof of Claim 5.8.

Claim 5.8. $\mathbf{Pr}[E] \geq 1 - 2^{-0.9\delta\cdot k}$

Proof of Claim 5.8. Consider first the matrix $\mathbf{G} \leftarrow \mathbb{F}_2^{\tilde{v}\times n}$ and recall that $n = (1+\delta)\cdot\tilde{v}$. By union bound,

$$\begin{aligned}
\mathbf{Pr}[\mathbf{G} \text{ has full rank}] &\geq 1 - \sum_{i=1}^{\tilde{v}} \mathbf{Pr}\left[\text{The } i^{th} \text{ row of } \mathbf{G} \text{ is linearly dependent on the first } i-1 \text{ rows}\right] \\
&\geq 1 - \sum_{i=1}^{\tilde{v}} \frac{2^{i-1}}{2^{(1+\delta)\tilde{v}}} \geq 1 - \frac{1}{2^{\delta\tilde{v}}}.
\end{aligned}$$

Similarly, for $\mathbf{B} \leftarrow \mathbb{F}_3^{\tilde{k}\times v}$, since $v = \tilde{k}(1+\delta)$ and $\tilde{k} \coloneqq \lceil(\log_3 2)k\rceil$

$$\mathbf{Pr}[\mathbf{B} \text{ has full rank}] \geq 1 - \frac{1}{3^{\delta\tilde{k}}} \geq 1 - \frac{1}{2^{\delta k}}$$

By union bound, it follows that $\mathbf{Pr}[E] \geq 1 - \frac{1}{2^{\delta k}} - \frac{1}{2^{\delta\tilde{v}}} \geq 1 - 2^{-0.9\delta k}$. □

5.1.1 Everywhere Collision Resistance ⇔ Seeded Codes with Adaptive Security

As mentioned above, ECR functions and seeded codes are closely linked. This connection is particularly apparent when considering error detection. Loosely speaking, it is easy to see that a code that detects p-fraction of errors is also an ECR that is collision resistance for every subset of relative size at least $\alpha = 1-p$. Intuitively, if it is hard to find two inputs $x \neq x'$ such that their encoding differs in at most p locations, it follows that every set of indexes of size at least $\alpha = 1-p$ contains at least one index that is different between the encoding of x and x' (which is exactly the definition of ECR).

The other direction, that is, showing that ECR implies error detection is not as immediate. To see this, assume for contradiction that a code does not p-detect. We now need to show that the code is not a ECR. Since the code does not p-detect then it follows that it is possible to find x, x' such that $\delta(\mathrm{Enc}_s(x), \mathrm{Enc}_s(x')) \leq p$, which means that $\mathrm{Enc}_s(x)$ and $\mathrm{Enc}_s(x')$ agree on at least $1-p$ indexes. Intuitively, we would want to argue that Enc is not a ECR on this set of agreement indexes (or even a subset of the set of agreements). However, this set could be different for different x, x'. That is, it could be the case that given $x, x' \leftarrow \mathsf{A}(s)$ it is easy to find agreeing indexes in ECR, but there exists no fixed set T for which it is "easy" to find x and x' that agree on it with noticeable probability. Still, we show that if ECR has very strong (exponential) security, then there exists a fixed set T (in fact a random set will do) that will have a collision. This is made formal in the following lemma:

Lemma 5.9. *The following holds:*

- *If* $(\mathrm{Enc}, \mathrm{Dec})$ *is a seeded code with adaptive security that detects* $0 \leq p < 1/2$ *relative errors then it is also a* $(1-p, \mathrm{neg})$-ECR.
- *If* $(\mathrm{Enc}, \mathrm{Dec})$ *is a* $(\alpha, \mu := (1/2)^{\alpha n \cdot (1-2\varepsilon)})$-ECR *for constants* $0 < \alpha, \varepsilon < 1/2$ *then it is also a seeded code with adaptive security that* p*-detect for any constant* $0 \leq p < 1/2 - \alpha - \varepsilon$.

We prove Lemma 5.9 below. But first we will use it to prove Theorem 5.3.

Proving Theorem 5.3.

Proof of Theorem 5.3. By Theorem 5.7, if Assumption 5.2 holds, then for every $0 < \gamma$ and every sufficiently small $0 < \alpha, \varepsilon$ there exists an $(\alpha, (1/2)^{\alpha n(1-\varepsilon)})$-ECR, $(\mathrm{Enc}, \mathrm{Dec})$ with rate $R \geq 1 - \gamma$. By Lemma 5.9, $(\mathrm{Enc}, \mathrm{Dec})$ is also a seeded code with adaptive security that can detect from $p < 1/2 - \alpha - \varepsilon$ errors and has rate $R \geq 1 - \gamma$, taking γ, α and ε to be sufficiently small concludes the proof. □

Proving Lemma 5.9.

Proof of Lemma 5.9. We start by proving the first item. Assume for contradiction that Enc is not $(1 - pn, \mathrm{neg})$-ECR, this implies that there exists a PPT A and a non negligible function γ such that for infinitely many $k \in \mathbb{N}$, there exists a set $T \subseteq [n]$ of size at least $(1-p)n$ for which:

$$\Pr_{\substack{s \leftarrow \{0,1\}^{\ell(k)}, \\ x, x' \leftarrow \mathsf{A}(s)}} \left[x \neq x' \wedge \mathrm{Enc}_s(x)|_T = \mathrm{Enc}_s(x')|_T \right] > \gamma(k).$$

However, the above contradicts the p-detection property of Enc_s since $\mathrm{Enc}_s(x)|_T = \mathrm{Enc}_s(x')|_T$ implies that $\delta(\mathrm{Enc}_s(x), \mathrm{Enc}_s(x')) \leq p$.

We now prove the second item. Assume for contradiction that $(\mathrm{Enc}, \mathrm{Dec})$ is not a seeded code that detects from p-relative errors where $p < 1/2 - \alpha - \varepsilon$. That is, there exists a PPT A and a non negligible function γ such that,

$$\Pr_{\substack{s \leftarrow \{0,1\}^{\ell(k)}, \\ x, z \leftarrow \mathsf{A}(s)}} \left[\delta(\mathrm{Enc}_s(x), z) \leq p \wedge \mathrm{Dec}_s(z) \notin \{\perp, x\} \right] > \gamma.$$

Define the reduction A' that uses A to break the (α, μ)-ECR as follows:

- Sample at random a subset $T \subseteq [n]$, such that $|T| = \alpha n$.
- Receive a seed $s \leftarrow \{0,1\}^{\ell}$ from the challenger, and give s to A.

- Receive (x, z) from A, and compute $x' = \mathrm{Dec}_s(z)$.
- Output (x, x')

Note that the view of A is independent of T. Let $Q = \{i \in [n] \colon \mathrm{Enc}_s(x)_i = \mathrm{Enc}_s(x')_i\}$. It follows that:

$$\begin{aligned}
&\mathbf{Pr}\left[\delta(\mathrm{Enc}_s(x), \mathrm{Enc}_s(x')) \le p \wedge x' \notin \{\perp, x\} \wedge T \subseteq Q\right] \\
&= \mathbf{Pr}\left[\delta(\mathrm{Enc}_s(x), \mathrm{Enc}_s(x')) \le p \wedge x' \notin \{\perp, x\}\right] \cdot \mathbf{Pr}\left[T \subseteq Q | \delta(\mathrm{Enc}_s(x), \mathrm{Enc}_s(x')) \le p \wedge x' \notin \{\perp, x\}\right] \\
&\ge \gamma \cdot \mathbf{Pr}\left[T \subseteq Q | (1-p)n < |Q|\right]
\end{aligned}$$

Moreover whenever,

$$\delta(\mathrm{Enc}_s(x), \mathrm{Enc}_s(x')) \le p \wedge x' \notin \{\perp, x\} \wedge T \subseteq Q$$

then A' wins the $(\alpha, \cdot)$-ECR game. This is because when $T \subseteq Q$ then $\mathrm{Enc}_s(x)|_T = \mathrm{Enc}_s(x')|_T$. Therefore, A' wins the (α, μ)-ECR game whenever $\mu \le \gamma \cdot \mathbf{Pr}\left[T \subseteq Q | (1-p)n < |Q|\right]$. Let $\beta \coloneqq 1 - p$ and note that:

$$\mathbf{Pr}\left[T \subseteq Q | \beta n < |Q|\right] \ge \frac{\binom{\beta n}{\alpha n}}{\binom{n}{\alpha n}} \ge (\beta - \alpha)^{\alpha n}$$

where inequality follows by the Claim 5.10 stated below. This concludes the proof since by our choice of parameters $\beta > 1/2 + \alpha + \varepsilon$, implying that

$$\begin{aligned}
\gamma \cdot (\beta - \alpha)^{\alpha n} &> (1/2 + \varepsilon)^{\alpha n} \\
&= (1/2)^{\alpha n}(1 + 2\varepsilon)^{\alpha n} \\
&\ge (1/2)^{\alpha n}(1/2)^{-2\varepsilon \cdot \alpha n} \\
&\ge (1/2)^{\alpha n(1-2\varepsilon)} \\
&\ge \mu
\end{aligned}$$

as desired. □

Claim 5.10. For positive integers n and constant α and β such that $\alpha < \beta$, for every sufficiently large n:

$$\frac{\binom{\beta n}{\alpha n}}{\binom{n}{\alpha n}} \ge (\beta - \alpha)^{\alpha n}$$

Proof of Claim 5.10. We start by expressing the ratio of the binomial coefficients as a product:

$$\frac{\binom{\beta n}{\alpha n}}{\binom{n}{\alpha n}} = \prod_{k=0}^{\alpha n - 1} \frac{\beta n - k}{n - k} = \prod_{k=0}^{\alpha n - 1} \frac{\beta - \frac{k}{n}}{1 - \frac{k}{n}}$$

To find a lower bound, we observe that for $0 \le k \le \alpha n - 1$:

$$\beta - \frac{k}{n} \ge \beta - \frac{\alpha n - 1}{n} \qquad 1 - \frac{k}{n} \le 1 \tag{6}$$

Therefore, each term in the product satisfies: $(\beta - \frac{k}{n})/(1 - \frac{k}{n}) \geq \beta - \frac{\alpha n - 1}{n} \geq \beta - \alpha$. Since this lower bound is constant across all terms in the product, it holds that: $\binom{\beta n}{\alpha n} / \binom{n}{\alpha n} \geq (\beta - \alpha)^{\alpha n}$. □

5.2 A Black-Box Transformation From Detection to Correction

In this section, we give a generic transformation that given:

- A systematic standard code $(\mathrm{Enc}_{\mathrm{List}}, \mathrm{Dec}_{\mathrm{List}})$ that list-decodes from p-relative errors with and a seeded code that detects from p'-relative errors with rate R_{List}.
- A seeded code $(\mathrm{Enc}', \mathrm{Dec}')$ that detects from p'-relative errors with rate R'.

Outputs a seeded code $(\mathrm{Enc}, \mathrm{Dec})$ that has rate $R = R_{\mathrm{List}} \cdot R'$ that uniquely decodes from p relative errors as long as $p' \geq 2p/R_{\mathrm{List}}$. The construction is very simple, namely, we define $\mathrm{Enc}_s(m) = \mathrm{Enc}_{\mathrm{List}}(\mathrm{Enc}'_s(m))$. Observe that if we had a code that at the same time list-decodes from p-errors and also detects from $2p$-detects then it must hold that this code in fact uniquely decodes. This holds since when performing list decoding, the codewords in the list are of distant at most $2p$ from each other. However, by the $2p$-detection property it should be hard to find two codewords that are this close, which implies that the list contains only one possible message.

Intuitively we would like to use the fact that we can have error-detection codes with a rate that is close to 1 which can detect almost 1/2 errors, and couple them together with good list-decoding property (to make the above idea go through). The key idea is that by using systematic codes we can apply the above argument on the embedded detecting code inside the overall codeword. Since the relative size of the error detection code embedded inside the list-decodable code is R_{List}, the overall detection we need is $2 \cdot p/R_{\mathrm{List}}$ (as opposed to just $2p$). We make this formal in the following lemma:

Lemma 5.11 (A Black-Box Transformation From Detection to Correction)**.** *Given a seeded code* $(\mathrm{Enc}', \mathrm{Dec}')$ *with adaptive [resp., selective] security that detects* p' *relative errors and has rate* R'*, and a binary* **systematic** *standard code* $(\mathrm{Enc}_{\mathrm{List}}, \mathrm{Dec}_{\mathrm{List}})$ *that can list decode from* p *relative errors and has rate* R_{List}*. If* $p' \geq 2 \cdot \frac{p}{R_{\mathrm{List}}}$ *then there exists a seeded code* $(\mathrm{Enc}, \mathrm{Dec})$ *with adaptive [resp., selective] security that* **uniquely** *corrects* p *relative errors and has rate* $R = R_{\mathrm{List}} \cdot R'$*.*

We prove Lemma 5.11 below, but first we will use it to prove Theorem 5.4.

Proving Theorem 5.4.

Proof of Theorem 5.4. If Assumption 5.2 holds, then by Theorem 5.3, for every constant $0 < \varepsilon'$ there exists a seeded code $(\mathrm{Enc}', \mathrm{Dec}')$ with adaptive security that can detect from $p < 12/\varepsilon'$ errors and has rate $R' \geq 1 - \varepsilon$. By Lemma 5.11, it follows that given a standard code $(\mathrm{Enc}_{\mathrm{List}}, \mathrm{Dec}_{\mathrm{List}})$ that can list-decode from p error with rate R_{List}, there exists a seeded code with adaptive security $(\mathrm{Enc}, \mathrm{Dec})$ that can uniquely decode from p errors as long as $p \leq (1/4 - \varepsilon)R_{\mathrm{List}}$, with a rate $R > R_{\mathrm{List}}(1 - \varepsilon)$. □

Proof of Lemma 5.11

Proof of Lemma 5.11. Let $\ell, n \in \mathrm{poly}$ be the bound on the seed size and codeword length of the seeded code $(\mathrm{Enc}', \mathrm{Dec}')$. We define the encoding and decoding functions of the new seeded code $(\mathrm{Enc}, \mathrm{Dec})$ as follows:

- On input $m \in \{0,1\}^k$ and seed $s \in \{0,1\}^{\ell(k)}$: $\mathrm{Enc}_s(m) = \mathrm{Enc}_{\mathrm{List}}(\mathrm{Enc}'_s(m))$.

- On input $z \in \{0,1\}^{(1/R_{\mathrm{List}})\cdot n(k)}$: $\mathrm{Dec}_s(z)$

  1. computes $\mathrm{List} = \mathrm{Dec}_{\mathrm{List}}(z)$
  2. Outputs the first $\hat{m} = \mathrm{Dec}'_s(\hat{c})$, such that $\hat{c} \in \mathrm{List}$, $\mathrm{Enc}'_s(\hat{m}) = \hat{c}$ and $\delta(\mathrm{Enc}_{\mathrm{List}}(\hat{c}), z) \leq p$. If no such message exists output $\perp$.

By construction the code $(\mathrm{Enc}, \mathrm{Dec})$ has rate $R = R_{\mathrm{List}} \cdot R'$. We now prove that the code can correct from p relative errors. For the sake of simplicity, we will prove the case that the code has adaptive security, the proof for the selective security is essentially the same.

Assume towards contradiction that the code $(\mathrm{Enc}, \mathrm{Dec})$ does not correct p relative errors. That is, there exists a PPT A such that for a non negligible γ:

1. Sampling $s \leftarrow \{0,1\}^{\ell(k)}$
2. $(m, z) \leftarrow \mathsf{A}(s)$

It holds that $\mathbf{Pr}\left[\delta(\mathrm{Enc}_s(m), z) \leq p \wedge \mathrm{Dec}_s(z) \neq m\right] \geq \gamma$. Let $c' := \mathrm{Enc}'_s(m)$ and $c =: \mathrm{Enc}_{\mathrm{List}}(c')$ and note that by the list-decoding properties of $(\mathrm{Enc}_{\mathrm{List}}, \mathrm{Dec}_{\mathrm{List}})$ if $\delta(c, z) \leq p$ then $\hat{m} \neq \perp$, which implies that:

$$\mathbf{Pr}\left[\delta(c, z) \leq p \wedge \hat{m} \notin \{\perp, m\}\right] \geq \gamma. \tag{7}$$

This holds since if $\delta(c, z) \leq p$ it follows that $c' \in \mathrm{List}$, and since $\mathrm{Dec}'_s(c') = m \neq \perp$ we conclude that $\hat{m} \neq \perp$.

We define the reduction A' that uses A to break the p'-detection property of $(\mathrm{Enc}', \mathrm{Dec}')$ as follows:

- Receive a seed $s \leftarrow \{0,1\}^{\ell}$ from the challenger and give it to A
- Receive (m, z) from A, and compute $\hat{m} = \mathrm{Dec}_s(z)$ and $\hat{c} = \mathrm{Enc}'_s(\hat{m})$
- Output $(m, \hat{c})$.

Let $z = (z_1, z_2)$ where z_1 is the first $n \cdot R_{\mathrm{List}}$ bits and z_2 is the last remaining bits. Since the standard code $(\mathrm{Enc}_{\mathrm{List}}, \mathrm{Dec}_{\mathrm{List}})$ is systemic if $\delta(c, z) \leq p$ it follows that

$$\delta(c', z_1) \leq \frac{p}{R_{\mathrm{List}}}.$$

Moreover, by construction if $\hat{m} \neq \perp$, it must hold that $\delta(\mathrm{Enc}_{\mathrm{List}}(\hat{c}), z) \leq p$, which similarly implies that:

$$\delta(\hat{c}, z_1) \leq \frac{p}{R_{\mathrm{List}}}.$$

Thus by the triangle inequality, since $\mathrm{Enc}'_s(m) = c'$ it holds that:

$$\mathbf{Pr}\left[\delta(\mathrm{Enc}'_s(m), \hat{c}) \leq 2 \cdot \frac{p}{R_{\mathrm{List}}} \wedge \hat{m} \notin \{\perp, m\}\right] \geq \gamma \tag{8}$$

Since by our choice of parameters $p' > 2 \cdot \frac{p}{R_{\mathrm{List}}}$, Equation (8) contradicts the p' detection properly of $(\mathrm{Enc}', \mathrm{Dec}')$. This concludes the proof. $\square$

References

[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.

[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.

[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.

[CLW18] Ran Canetti, Alex Lombardi, and Daniel Wichs. Fiat-shamir: From practice to theory, part ii (nizk and correlation intractability from circular-secure fhe). *Cryptology ePrint Archive*, 2018.

[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.

[GR08a] Venkatesan Guruswami and Atri Rudra. Better binary list decodable codes via multilevel concatenation. *IEEE Transactions on Information Theory*, 55(1):19–26, 2008.

[GR08b] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on information theory*, 54(1):135–150, 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.

[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.

[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.

[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.

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 33–43. ACM, 1989.

[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.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 387–394. ACM, 1990.

[Sha48] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[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.

[SW24] Jad Silbak and Daniel Wichs. Detecting and correcting computationally bounded errors: A simple construction under minimal assumptions. Cryptology ePrint Archive, Paper 2024/1461, 2024.

A Appendix

A.1 Proof of Theorem 4.7

In this section we prove Theorem 4.7. Guruswami and Rudra [GR08a], showed an explicit construction for a family of linear codes that achieves list decoding up to the Blokh-Zyablov bound. In this section, we show how we can interpret their construction so that it has the nested list decoding property that is stated in Theorem 4.7.

We start by describing the construction of Guruswami and Rudra [GR08a] (using their notation). The code is constructed using the so-called "multilevel concatenation" which is a generalization of standard code concatenations. Let the integer $s \geq 1$ be a parameter to be chosen later. The main idea is that instead of using a code concatenation with a single outer code, we will use s outer codes $C^0_{\text{out}}, \cdots, C^{s-1}_{\text{out}}$ and a single inner code C_{in}. For every $0 \leq i \leq s-1$, C^i_{out} is a list-recoverable code with the same block length N and with rate R_i over the field of $\mathbb{F}_{2^a}$ (for some integer $a > 1$) and C_{in} is a list-decodable code with rate r_0 and block length n that maps elements in $\mathbb{F}^s_{2^a}$ to elements in $\mathbb{F}^n_2$. That is, $C^i_{\text{out}} \colon (\mathbb{F}_{2^a})^{R_i \cdot N} \to (\mathbb{F}_{2^a})^N$ and $C_{\text{in}} \colon (\mathbb{F}_{2^a})^s \to (\mathbb{F}_2)^n$.

Given a message $(m_0, \cdots, m_{s-1}) \in \mathbb{F}_{2^a}^{R_0 \cdot N} \times \cdots \times \mathbb{F}_{2^a}^{R_{s-1} \cdot N}$, we first compute $C^i_{\text{out}}(m_i) = M_i$ for every $0 \leq i \leq s-1$. For every $j \in [N]$, let M^j_i denote the j^{th} symbol in M_i and for every $j \in [N]$, let M^j denote the ordered collection of the j^{th} symbols of $M_0, \cdots M_{s-1}$, that is, $M^j = (M^j_0, \cdots, M^j_{s-1})$. The inner code is then applied on $C_{\text{in}}(M^j)$ for every $j \in [N]$. Formally, the multilevel concatenation denoted as $C \coloneqq (C^0_{\text{out}} \times \cdots \times C^{s-1}_{\text{out}}) \circ C_{\text{in}}$ is defined to be

$$C(m_0, \cdots, m_{s-1}) = (C_{\text{in}}(M^1), \cdots, C_{\text{in}}(M^N)).$$

Where the rate of the overall construction is

$$R = r_0 \cdot \frac{1}{s} \sum_{i=0}^{s-1} R_i.$$

Multi-nested list-decodable inner code: The construction of Guruswami and Rudra [GR08a], requires that the linear inner code C_{in} has an s-level nested list-decodable property (which we refer to as multi-nested). Specifically, let $\mathbf{G} \in \mathbb{F}_2^{r_0 \cdot n \times n}$ be the generating matrix of C_{in} and note that $r_0 = as/n$. For every $0 \leq i \leq s-1$, let $r_j = r(1 - j/s)$, and let C^j_{in} be the subcode given by the generating matrix $\mathbf{G}^j \in \mathbb{F}_2^{r_j \cdot n \times n}$ defined to be the last $r_j \cdot n$ rows of $\mathbf{G}$. Guruswami and Rudra showed how to explicitly construct an inner code such that each of the subcodes C^j_{in} is also efficiently list-decodable and where each of the subcodes (and the whole inner code) fall on the Gilbert Varshamov bound (that is, the relative distance of the subcode C^i_{in} is $\delta_i \geq H^{-1}(1 - r_i)$)).

The Blokh-Zyablov bound describes the trade-off between rate and relative distance between when the inner code (and all its subcodes) meets the Gilbert Varshamov bound and the outer codes fall on the Singletone bound (that is, C^i_{out} has relative distance $1 - R_i$). This implies that the multilevel concatenated code then has relative distance at least $\delta = \min_{0 \leq i \leq s-1}(1 - R_i) \cdot H^{-1}(1 - r_i)$. Putting everything together

as function of the overall distance δ implies that there exists a multilevel concatenation construction with rate

$$R^s_{BZ}(\delta) = \max_{0<r_0<1-H(\delta)} r - \frac{r_0}{s}\sum_{i=0}^{s-1}\frac{\delta}{H^{-1}(1-r_0+r_0 i/s)} \tag{9}$$

Finally, note that for an increasing s, $R^s_{BZ}(\delta)$ approaches $R_{BZ} \coloneqq 1 - H(\delta) - \delta \cdot \int_0^{1-H(\delta)} \frac{dx}{H^{-1}(1-x)}$.

A.1.1 Adopting the construction of Guruswami and Rudra for list decoding up to the Blokh-Zyablov bound.

We now explain why the construction of Guruswami and Rudra satisfies Theorem 4.7. Their construction makes use of folded-Reed Solomon codes as the outer codes $C^0_{\text{out}}, \cdots, C^{s-1}_{\text{out}}$ where for each $0 \le i \le s-1$, C^i_{out} has rate $R_i = 1 - \frac{p}{H^{-1}(1-r_i)}$. Moreover, they show how to explicitly construct a generating matrix $\mathbf{G}$ that satisfies the multi-nested list-decoding properties described above such that the rate of $\mathbf{G}^i$ is $r_i = r_0(1-i/s)$. Note also that the overall multilevel concatenated code $C \coloneqq (C^0_{\text{out}} \times \cdots \times C^{s-1}_{\text{out}}) \circ C_{\text{in}}$ is also a linear code and let $\mathbf{A}$ denote the generating matrix of C. The proof of Theorem 4.7 follows directly from the lemma stated below:

Lemma A.1. *Let $\ell = o(n \cdot N)$ be an integer let $0 < \gamma$ be a sufficiently small constant as in Theorem 4.7, and let $\mathbf{A}$ be the generating matrix for the code $C \coloneqq (C^0_{\text{out}} \times \cdots \times C^{s-1}_{\text{out}}) \circ C_{\text{in}}$. It holds that matrix $\mathbf{A}_{\text{nest}}$ defined to be the last ℓ rows of $\mathbf{A}$, is a generating matrix for an $(1/2-\gamma, L_{\text{nest}})$-list decodable code with efficient decoding algorithm for some $L_{\text{nest}} \in \text{poly}$.*

Proof of Lemma A.1. The rest of this section is dedicated to proving Lemma A.1. We make use of the following claims:

Claim A.2. The matrix $\mathbf{B}$ defined to be the last $z \coloneqq \frac{1}{s} \cdot R_{s-1} \cdot r_{s-1} \cdot (N \cdot n)$ rows of $\mathbf{A}$, is a generating matrix for the (normal code concatenation) $C^{s-1}_{\text{out}} \circ C^{s-1}_{in}$.

Proof of Claim A.2. The claim follows by linearity and by the definition of multilevel concatenation. Recall that the code C takes as inputs messages $m = (m_0, \cdots, m_{s-1})$, and consider the subcode C' of C induced by zeroing out the first $s-1$ parts of the messages and leaving m_{s-1} (that is, fixing $m_i = 0^{R_i}$ for every $0 \le i \le s-2$). Note that binary representation of m_{s-1} is of size z. By definition it holds that,

$$\begin{aligned}
C'(m_{s-1}) = (0, \cdots, 0, m_{s-1}) \cdot \mathbf{A} &= C(0, \cdots, 0, m_{s-1}) \\
&= \Big(C_{\text{in}}\big(0, \cdots, 0, C^{s-1}_{\text{out}}(m_{s-1})_1\big), \cdots, C_{\text{in}}\big(0, \cdots, 0, C^{s-1}_{\text{out}}(m_{s-1})_n\big)\Big) \\
&= \big(C^{s-1}_{\text{in}}(C^{s-1}_{\text{out}}(m_{s-1})_1), \cdots, C^{s-1}_{\text{in}}(C^{s-1}_{\text{out}}(m_{s-1})_n)\big) \\
&= C^{s-1}_{\text{out}} \circ C^{s-1}_{in}(m_{s-1})
\end{aligned}$$

Thus the matrix $\mathbf{B}$ defined to be the last $z \coloneqq \frac{1}{s} \cdot R_{s-1} \cdot r_{s-1}$ rows of $\mathbf{A}$ is the generating matrix of the subcode $C^{s-1}_{\text{out}} \circ C^{s-1}_{in}(m_{s-1})$, since $(0, \cdots, 0, m_{s-1}) \cdot \mathbf{A} = m_{s-1} \cdot \mathbf{B}$. □

Recall that the outer codes in the construction and in particular C^{s-1}_{out} are folded-Reed Solomon codes. Without lose of generality, we assume that the generating matrix for the folded-Reed Solomon codes is given by the inverse order of the set of monomials (that is, $X^{R_i \cdot N - 1}, \cdots, X, 1$) so that bottom r rows of the generating matrix correspond to polynomials of degree at most r.

We will make use of the following fact about folded Reed Solomon codes.

Claim A.3. Let C_{out} be a folded Reed-Solomon over the field F_{2^a} for some integer $a > 1$ with a constant rate $R > 0$ such that C_{out} is $(1-R-\varepsilon, \ell, L)$-list-recoverable for a sufficiently small constant ε. Let $\mathbf{V}$ be the (standard) generating matrix of C_{out} of dimensions $RN \times N$ (that is, $\mathbf{V}$ is the folding of the Vandermonde matrix). Let $\hat{C}_{\text{out}}$ be a subcode of C_{out} define by the matrix $\mathbf{V}^w$ where $\mathbf{V}^w$ is the last w rows of $\mathbf{V}$. It holds that $\hat{C}_{\text{out}}$ generated by $\mathbf{V}^w$ is also a folded Reed-Solomon that is $(1 - w/n - \varepsilon, \ell, L)$-list-recoverable code.

Proof of Claim A.3. Note that $\mathbf{V}^w$ is a generating matrix that represents the first w (smallest) monomials in the polynomial evaluation (that defines the code) and as such they define the evaluation on a polynomial of a smaller degree (degree at most w). For concreteness consider for example the Vandermonde matrix

$$\mathbf{V} = \begin{pmatrix} \alpha_1^{k-1} & \alpha_2^{k-1} & \alpha_3^{k-1} & \cdots & \alpha_n^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{w} & \alpha_2^{w} & \alpha_3^{w} & \cdots & \alpha_n^{w} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_n \\ 1 & 1 & 1 & \cdots & 1 \end{pmatrix}$$

Thus, the last w rows of $\mathbf{V}$, denoted by $\mathbf{V}^w$, represent the evaluation of a w degree polynomial. Since the folding operation does not affect this property, the above also holds for folded Reed-Solomon codes. This means that $\mathbf{V}^w$ is also a folded Reed-Solomon code with the same folding parameter as $\mathbf{V}$. Since the rate of the subcode is smaller than the original code, it holds that the code subcode is also list-recoverable for the same choice of ℓ and L.[10] □

We are now ready to prove Lemma A.1.

Proof of Lemma A.1. By Claim A.2, the matrix $\mathbf{B}$ is a generating matrix for the code $C_{\text{out}}^{s-1} \circ C_{in}^{s-1}$, and $\mathbf{B}$ is a subcode of $\mathbf{A}$, where $\mathbf{B}$ is defined to be the last z rows of $\mathbf{A}$.

Recall that the subcode C_{nest} is given by the generating $\mathbf{A}_{\text{nest}}$ defined to be the last ℓ rows of $\mathbf{A}$. Note that $\ell = o(z)$ (since $\ell = o(n \cdot N)$). This implies that $\mathbf{A}_{\text{nest}}$ is a subcode of $\mathbf{B}$ which is a subcode of $\mathbf{A}$.

Thus, C_{nest} is a subcode of $C_{\text{out}}^{s-1} \circ C_{in}^{s-1}$ (since $\mathbf{A}_{\text{nest}}$ is the last ℓ rows of $\mathbf{B}$). Finally, define $\hat{C}_{\text{out}}^{s-1}$ to be the subcode of C_{out}^{s-1} defined by taking the last $w \coloneqq \ell/n$ rows of $\mathbf{V}$ the generating matrix of C_{out}^{s-1}. Thus on input $m_{\text{nest}} \in \{0,1\}^{\ell}$, it holds that:

$$\begin{aligned} C_{\text{out}}^{s-1} \circ C_{in}^{s-1}(0^{z-\ell}, m_{\text{nest}}) &= \hat{C}_{\text{out}}^{s-1} \circ C_{in}^{s-1}(m_{\text{nest}}) \\ &= C_{\text{nest}}(m_{\text{nest}}) \\ &= m_{\text{nest}} \cdot \mathbf{A}_{\text{nest}} \end{aligned}$$

By Claim A.3, $\hat{C}_{\text{out}}^{s-1}$ is an $(1 - w/N - \varepsilon, \ell, L)$-list-recoverable code and C_{in}^{s-1} is a $(\delta_{s-1} \coloneqq H^{-1}(1 - 1/s), \ell)$-list-decodable code. It follows that C_{nest} is $(\delta_{s-1} \cdot (1 - o(1) - \varepsilon), L)$-list decodable code with an efficiently decoding algorithm. Thus for every constant $\gamma > 0$, by taking by taking a sufficiently large constant s and a sufficiently small constant ε, C_{nest} is $(1 - \gamma, L)$-list decodable code with an efficiently decoding algorithm for some $L \in \text{poly}$. This concludes the proof. □

[10] This is implicit in the result of [GR08b] for folded Reed-Solomon codes. Note also, that when $w = o(n)$ we could just use the list-recovery algorithm for standard Reed-Solomon code to achieve the same ℓ with essentially the same final list size L.