Code Offset in the Exponent

Luke Demarest[*] Benjamin Fuller[†] Alexander Russell[‡]

May 27, 2020

Abstract

We study the *code offset* construction used extensively in fuzzy extractors. The goal is to derive a key, key, from a value $\mathbf{e}$ and be able to reproduce key only if a nearby value $\mathbf{e}'$ is known. The construction stores the value $\mathbf{e}$ in a one-time pad which is sampled as a codeword, $\mathbf{Ax}$, of a linear error-correcting code: $\mathbf{Ax}+\mathbf{e}$ (Juels and Wattenberg, CCS 1999). The value key is then derived using a randomness extractor on $\mathbf{e}$. The construction is also used in pattern matching obfuscation (Bishop et al., Crypto 2018).

We study *code offset in the exponent*: instead of storing the value directly, store a vector of group elements that are a random group generator to the output of the code offset (instantiated with a random linear code). The construction and adversary are both provided with the code $\mathbf{A}$. If the resulting vector is indistinguishable from random group elements, the construction has important properties: 1) it leaks nothing about $\mathbf{e}$ 2) it allows for multiple instantiations from the same $\mathbf{e}$ and 3) it retains error correction for distances between $\mathbf{e}$ and $\mathbf{e}'$ that are a subconstant fraction of their length (in the Hamming metric).

We ask which error distributions make code offset indistinguishable from random group elements in the generic group model. We show a sufficient condition for hardness of an error distribution: the inner product between the error distribution and all vectors in the null space of the code must be unpredictable.

This is equivalent to error distributions where distinguishing learning with errors is hard for algorithms that work in two stages: 1) arbitrary computation on the code $\mathbf{A}$ followed by 2) a linear computation on the received value $\mathbf{y}$ which is either $\mathbf{Ax}+\mathbf{e}$ or uniform. Prior attacks such as Arora and Ge (ICALP, 2011) fit in this paradigm.

In this model, for any polynomial number of samples, the discretized Gaussian and uniform interval error distributions are hard but not the uniform bit (due to Arora and Ge's attack). Our primary result in this model is that for large enough fields, distinguishing learning with errors is hard for all error distributions with minentropy that is larger than log of the size of the nullspace of the code by any super logarithmic amount.

In the generic group model, the above analysis yields a computational fuzzy extractor that is secure for the distribution families considered in many prior works, while providing reusability and without leaking anything about the value $\mathbf{e}$. The analysis also shows a prior construction of pattern matching obfuscation (Bishop et al., Crypto 2018) is secure for more distributions than previously known.

Assuming hardness of discrete log, we also show hardness of bounded distance decoding of random linear codes with uniform input point, quantitatively improving prior bounds of Peikert (TCC 2006).

Keywords code offset; learning with errors; error-correction; generic group model; fuzzy extractors

[*] Email: `luke.h.demarest@gmail.com`. University of Connecticut.

[†] Email: `benjamin.fuller@uconn.edu`. University of Connecticut.

[‡] Email: `acr@uconn.edu` University of Connecticut.

1 Introduction

In fuzzy extractors [DORS08], the goal is to derive a stable key, key, from a noisy value $\mathbf{e}$. That is, the goal is create a public value pub where the same key can be found given $\mathbf{e}'$ that is close to $\mathbf{e}$. The code-offset construction [JW99] and its variants [DGV$^+$16] represent the majority of constructions. The construction is determined by a linear error-correcting code $\mathbf{A}$ and a secret, uniformly random $\mathbf{x}$; given a sample $\mathbf{e}$ from the noisy source, the construction publishes the pair

$$\mathsf{pub} = (\mathbf{A}, \mathbf{A}\mathbf{x} + \mathbf{e})\,.$$

The salient feature of the construction is that with a second sample $\mathbf{e}'$ from the source—which we assume has small Hamming distance from $\mathbf{e}$[1]—the difference

$$(\mathbf{A}\mathbf{x} + \mathbf{e}) - \mathbf{e}' = \mathbf{A}\mathbf{x} + (\mathbf{e} - \mathbf{e}')$$

is evidently close to the codeword $\mathbf{A}\mathbf{x}$. By decoding the error correcting code one can recover $\mathbf{x}$ (and $\mathbf{e}$).[2]

When $\mathbf{A} \in \mathbb{F}_q^{n \times k}$, so that the construction is carried out over the field with q elements, existing security analyses guarantee that the entropy of $\mathbf{e}$ decreases by at most $(n-k)\log q$ bits when conditioned on pub. This bound is tight for some distributions: for a fixed code $\mathbf{A}$, consider a distribution $\mathbf{e}$ that is uniform over the coset leaders of $\mathbf{A}$, then the coset of pub determines $\mathbf{e}$. The above analysis provides no guarantees for structured but low-entropy distributions [CFP$^+$16]. This construction has two further security weaknesses:

1. *Leakage.* It may leak potentially sensitive attributes of the value $\mathbf{e}$.
2. *One-time.* Users cannot safely "enroll" a single value $\mathbf{e}$ multiple times (by publishing $\mathsf{pub}^1, \mathsf{pub}^2, ...$ from noisy readings $\mathbf{e}^1, \mathbf{e}^2, ...$).

Motivated by these shortcomings, we study the security properties of this construction when carried out *in the exponent.* Specifically, if r is a random generator for a cyclic group $\mathbb{G}$ of prime order q, we consider

$$\mathsf{pub} = (\mathbf{A}, r, r^{\mathbf{A}\mathbf{x}+\mathbf{e}})\,,$$

where we adopt the shorthand notation $r^{\mathbf{v}}$, for a vector $\mathbf{v} = (v_1, \ldots, v_n)^\intercal \in (\mathbb{Z}_q)^n$, to indicate the vector $(r^{v_1}, \ldots, r^{v_n})^\intercal$. We refer to this construction as *code-offset in the exponent* and show that it possesses strong security properties under natural cryptographic assumptions on the group $\mathbb{G}$; in particular, it addresses both of the complaints above.

We focus on code-offset in the exponent with a random linear code and adopt the generic group model [Sho97] to reflect the cryptographic properties of the underlying group. We are then able to establish strong guarantees that address the leakage and one-time concerns mentioned above. In general, security properties of the construction depend on the distribution of $\mathbf{e}$. Since code offset is used when $\mathbf{e}$ is drawn from sources in nature where symbols are correlated (e.g., biometrics), understanding this relationship is fundamental. Thus, the goal is to characterize the distributions on $\mathbf{e}$ for which $r^{\mathbf{A}\mathbf{x}+\mathbf{e}}$ is indistinguishable from a vector of random group elements (conditioned on $\mathbf{A}$ and r).

Peikert [Pei06] showed that when $\mathbf{A}$ is a Reed-Solomon code, decoding "in the exponent" is hard in the generic group model. Specifically, Peikert's result considers a class of distributions $\mathbf{e} \in \mathbb{F}_q^n$ that are

[1] It is also possible to consider other distances between $\mathbf{e}$ and $\mathbf{e}'$. However the error correction techniques required are different. We focus on Hamming error exclusively in this work.

[2] Applying a randomness extractor [NZ93] on either $\mathbf{x}$ or $\mathbf{e}$ yields a uniform key.

determined by placing t uniformly selected elements of $\mathbb{F}_q$ in t randomly selected coordinates, while assigning other coordinates the value 0. Observe that an adversary can use *information set decoding* [Pra62]: repeatedly select subsets of size k and subject them to an "interpolation test:" in case the subset is free of errors, the rest of the codeword can be recovered directly via linear interpolation as this fortuitously involves only linear operations with known scalars [CG99] (so that such operations can be carried out in the exponent). This procedure succeeds when $tk = O(n \log n)$. Peikert's results show that this is tight: no attacker can distinguish $r^{\mathbf{Ax}+\mathbf{e}}$ from uniform elements when $tk = \omega(n \log n)$. The question of hardness has remained open for more general distributions on $\mathbf{e}$.

We remark that this question is equivalent to asking what error distributions make distinguishing LWE (learning with errors) hard in the generic group model, though this is an unusual setting for LWE as it requires a super polynomial size field and the notion of "small" is destroyed by the generic group model—in particular, "rounding" is not possible. It is for these reasons that we refer to the construction as code offset rather than LWE. Despite this, some prior attacks on LWE can indeed be instantiated in the generic group model (with $\mathbf{A}$ provided in the clear). Arora and Ge's attack [AG11] distinguishes LWE samples from uniform when the error distribution has independent symbols which each take a constant number of values. The attack works in two stages, linearizing polynomials whose degree depends on the number of possible errors and then performing Gaussian elimination. Only the Gaussian elimination stage requires $\mathbf{Ax}+\mathbf{e}$ and can be done in a generic group (of known order). For binary errors, as considered by Micciancio and Peikert [MP13], the attack works when $n = \Theta(k^2)$. Thus, the generic group model does capture nontrivial families of LWE attacks.

To the best of our knowledge this is first time this question has been considered.[3] Brakerski and Döttling [BD20] considered the question of distribution flexibility for $\mathbf{x}$: showing hardness when the conditional entropy of $\mathbf{x}$ conditioned on $\mathbf{x}+\mathbf{e}$ is large for Gaussian $\mathbf{e}$.

See Sections 1.2 and 1.3 respectively for details on how to build fuzzy extractors for Hamming errors and pattern matching obfuscation from code offset in the exponent.

1.1 When is code offset in the exponent hard?

In the generic group model, we establish (Theorem 2) that distinguishing code offset in the exponent from a random vector of group elements is hard for any error distribution $\mathbf{e}$ where the following game is hard to win:[4]

```
Experiment $\mathbb{E}^{\mathsf{MIPURS}}_{\mathcal{A},\mathbf{e}}(n,k)$:
$\psi \leftarrow \mathbf{e}; A \xleftarrow{\$} \mathbb{F}_q^{n\times k}$.
$(b,g) \leftarrow \mathcal{A}(A)$.
If $b \in \mathtt{null}(A)$, $b \neq \mathbf{0}$ and $\langle b, \psi\rangle = g$ output 1.
Output 0.
```

Observe that the role of the random matrix A in the game above is merely to define a random subspace of (typical) dimension k.

We call this condition on an error distribution MIPURS or *maximum inner product unpredictable over random subspace*. Specifically, a random variable $\mathbf{e}$ over $\mathbb{F}_q^n$ is (k,β)–MIPURS if for all $\mathcal{A}$ (which knows the distribution of $\mathbf{e}$), $\Pr[\mathbb{E}^{\mathsf{MIPURS}}_{\mathcal{A},\mathbf{e}}(n,k) = 1] \leq \beta$. Clearly, MIPURS is necessary. If the adversary $\mathcal{A}$ is

[3] Dagdelan et al. [DGG15] consider a version of this problem where $\mathbf{A}$ is only provided in the group and show this problem is hard assuming DDH. It is crucial in our applications that $\mathbf{A}$ is provided in the clear.

[4] We use boldface to represent random variables, capitals to represent random variables over matrices, and plain letters to represent samples. We use ψ to represent samples from $\mathbf{e}$ to avoid conflict with Euler's number.

information theoretic, for all distributions $\mathbf{e}$ that are not MIPURS one can find a nonzero vector b in the null space of A whose inner product with $\mathbf{e}$ is predictable, thus predicting $\langle b, \mathbf{Ax}+\mathbf{e}\rangle = \langle b, \mathbf{e}\rangle \overset{?}{=} g$. This is not the case for a uniform distribution, $\mathbf{U}$, the value $\langle b, \mathbf{U}\rangle$ is uniform (and thus is $\langle b, \mathbf{U}\rangle = g$ with small probability if the size of q is super polynomial). Thus b serves as a way to distinguish $\mathbf{Ax}+\mathbf{e}$ from $\mathbf{U}$.

For any $d = \mathtt{poly}(n)$ there is a efficiently constructible distribution $\mathbf{e}$ whose entropy is $\log(dq^{n-k-1})$ where the MIPURS game is winnable by an efficient adversary with noticeable probability: For $1 \le i \le d$, sample some d random linear spaces $\mathbf{B}_i$ of dimension $n-k-1$ and define $\mathbf{E}_i$ to be all points in a random coset g_i of $\mathbf{B}_i$. Consider the following distribution $\mathbf{e}$:

1. Pick $i \leftarrow \{1, ..., d\}$ for some polynomial size d.
2. Output a random element of $\mathbf{E}_i$.

The support size of this distribution is approximately dq^{n-k-1}. For a random $n-k$ dimensional $\mathtt{null}(\mathbf{A})$, with high probability $\exists b_i \neq \mathbf{0}$ such that $b_i \in \mathtt{null}(\mathbf{A}) \cap \mathtt{null}(\mathbf{B}_i)$ (since $\dim(\mathtt{null}(\mathbf{A})) + \dim(\mathtt{null}(\mathbf{B}_i)) > n$). The adversary can calculate these b_i's. Then the adversary just picks a random i and predicts (b_i, g_i). Note that if $\mathbf{A}$ is some fixed code (chosen before adversary specifies $\mathbf{e}$), then $\mathbf{E}_i$ can directly be a coset of $\mathbf{A}$ and one can increase the size of $\mathbf{E}$ to dq^{n-k}.

The main technical contribution of this work is a characterization of the MIPURS distributions. Our main theorem is that all distributions whose entropy is greater than $\log(\mathtt{poly}(n)q^{n-k})$ are MIPURS. (Note this is a factor of q away from matching the size of our counterexample for a random code.) Informally (see Corollary 13):

Theorem 1 (Informal)**.** *Let $n, k \in \mathbb{Z}$ be parameters. Let $q = q(n)$ be a large enough prime. For all $\mathbf{e} \in \mathbb{Z}_q^n$ whose minentropy is at least $\omega(\log n) + \log(q^{n-k})$, there exists some $\beta = \mathtt{ngl}(n)$ for which $\mathbf{e}$ is (k, β)–MIPURS.*

Information theoretic analysis of code offset provides a key of length $\omega(\log n)$ when the initial entropy of $\mathbf{e}$ is at least $\omega(\log n) + \log(q^{n-k})$. However, information theoretic analysis of code offset reduces the entropy of $\mathbf{e}$ which may allow prediction of sensitive attributes. In the generic group analysis no function of $\mathbf{e}$ is leaked. The generic group analysis allows the construction to be safely reused multiple times (with independent generators).

Proof Intuition. Suppose in the above game the adversary generated $\mathbf{e}$ as the span of a linear space $\mathbf{E}$ with the goal that $\mathtt{null}(\mathbf{A}) \cap \mathtt{null}(\mathbf{E}) \supset \{\mathbf{0}\}$. For a random, independent $\mathbf{B} \overset{\text{def}}{=} \mathtt{null}(\mathbf{A})$, the probability of $\mathbf{B}$ and $\mathtt{null}(\mathbf{E})$ overlapping is noticeable only if the sum of the dimensions is more than n (Claim 7). This creates an upper bound on the dimension of $\mathbf{E}$ of $n-k$ (ignoring the unlikely case when $\mathbf{A}$ is not full rank).

There are three ways our setting differs from this classic problem. First, the distribution $\mathbf{e}$ is not linear, second the adversary doesn't have to "nullify" the entire space $\mathbf{B}$—only a single vector, and third, the adversary can predict any inner product, not just 0.

Our proof is dedicated to removing these three obstacles in turn. First we upper bound the size of a set E where each vector is predictable in the MIPURS game. We show for a random sample from E to have a large intersection with a low dimensional space requires E to have size at least that of the low dimensional space (Lemma 6). In Lemma 8, we switch from measuring the size of intersection of a sample of E with respect to the worst case subspace to how "linear" E is with respect to the worst vector in an average case subspace. This result thus controls an "approximate" algebraic structure in the sense of additive combinatorics. We show the adversary can't do much better on a single vector b as long as it is chosen from a random $\mathbf{B}$.

The above argument considers the event that the adversary correctly predicts an inner product of 0; this can be transformed to an arbitrary inner product by a compactness argument which introduces a modest loss in parameters (Theorem 10). Once we have a bound on how large a predictable set E can be, another superlogarithmic factor guarantees that all distributions $\mathbf{e}$ with enough minentropy are not predictable.

We also show other distributions are MIPURS which are relevant for LWE and the code offset construction; the proofs are straightforward.

1. **Independent.** Distributions where symbols are independent and contribute a super logarithmic amount of entropy, including the discretized Gaussian [Reg05] and uniform interval [DMQ13]. It follows that most previously considered error distributions for LWE yield hardness in the generic group model. These results hold for an arbitrary polynomial number of samples.

   We do not consider hardness for the uniform bit error [MP13]. This error distribution cannot be secure for an arbitrary polynomial number of samples due to Arora and Ge's attack [AG11].

2. **Location.** Distributions where errors are either zero or random. The location of zero errors may be correlated as long as it is unlikely for a subset (of appropriate size) to have no errors. This setting is closer to decoding random linear codes [BMvT78] than traditional LWE. Peikert's result considered decoding random linear codes in the exponent where the position of errors is uniformly distributed [Pei06]. We show a sufficient condition for security is that each subset of size k has an overwhelming probability of including a nonzero error.

As future work, it may be possible to show hardness for MIPURS in polynomial size fields, however such a result requires a change in modeling. One can no longer give the adversary "handles" but instead the adversary would specify linear combinations without seeing $\mathbf{Ax}+\mathbf{e}$ and learn if the resulting inner product is zero. However, such a change would require careful analysis. Our results use the size of q multiple times to provide guarantees on rank and collision probability.

1.2 Application to Fuzzy Extractors

In a fuzzy extractor, the goal is to derive a stable key, key, from a noisy value $\mathbf{e}$. That is, the goal is create a public value pub where the same key can be found given $\mathbf{e}'$ that is close to $\mathbf{e}$. When $\mathbf{e}$ is a $(k-\Theta(1),\beta)$–MIPURS distribution for a code with dimension k and $\beta = \mathtt{ngl}(n)$ then code-offset in the exponent yields a fuzzy extractor in the generic group model (Theorem 14). Showing this requires one additional step of key extraction; we use a result of Akavia, Goldwasser, and Vaikuntanathan [AGV09, Lemma 2] which states that dimensions of $\mathbf{x}$ become hardcore once there are enough dimensions for LWE to be indistinguishable. This reduction is entirely linear and holds in the generic group setting. If one uses a random generator in each invocation of Gen the construction allows multiple (noisy) *enrollments* of $\mathbf{e}$, known as a reusable fuzzy extractor [Boy04].

Canetti and Goldwasser's result [CG99] says linear decoding is efficient when $tk = O(n\log n)$. This implies if k is just $\omega(\log n)$ one can achieve decoding for $t = o(n)$. The adversary's decoding is efficient if $k = O(\log n)$ so $k = \omega(\log n)$ is smallest safe setting for k.[5] This uses information set decoding: repeatedly

[5] Importantly, the adversary's view of errors is different than the construction, the adversary sees $\mathbf{Ax}+\mathbf{e}$, the construction sees $\mathbf{Ax}+(\mathbf{e}-\mathbf{e}')$.

selecting random subsets of coordinates and hoping to find a subset with no errors. For each such subset one can compute a candidate $g^{\mathbf{x}'}$ and test correctness by checking the weight of $g^{(\mathbf{A}\mathbf{x}+\mathbf{e}-\mathbf{e}')-\mathbf{A}\mathbf{x}'}$.

Concurrent work of Galbraith and Zobernig [GZ19] introduces a new subset sum computational assumption to build a secure sketch that is able to handle $t = \Theta(n)$ errors, they conjecture hardness for all securable distributions. A secure sketch is the error correction component in most fuzzy extractors. Their assumption is security of the cryptographic object and deserves continued study.

A unifying construction. Our construction unifies multiple computationally secure fuzzy extractor constructions that are used for different distributions. (We omit discussion of recent interesting line of works [WL18, WLG19] that use information-theoretic tools for error correction and computational tools to achieve additional properties. Those constructions embed a variant of the code offset.) Code offset in the exponent mitigates security weaknesses in the information-theoretic analysis of the code-offset construction. For all supported distributions, the construction does not leak information about $\mathbf{e}$ and is reusable. Since we only require an additional super logarithmic amount of entropy we are secure whenever the information-theoretic code offset is secure (for large enough q).

Since we support **Independent** distributions, our construction is secure for common LWE admissible distributions, and thus is secure when the LWE based fuzzy extractor of Fuller et al. [FMR13] is known to be secure. Indeed, Fuller et al.'s construction is code offset instantiated with a random code, the only change we make is moving to a "hard" group. We note that their "error-correction" was entirely linear so there is no downside to moving to the exponent.

Canetti et al. [CFP$^+$16] presented a fuzzy extractor that explicitly places subsets of $\mathbf{e}$ in a digital locker [CD08]. To achieve meaningful error tolerance for an actual biometric, millions of these lockers are required [SSF19]. Canetti et al.'s construction is secure when a random subset of bits is hard to predict (Definition 6). A binary $\mathbf{e} \in \{0,1\}^n$ where subsets are hard to predict can be amplified into a **Location** source, whose zero error positions may be correlated. If $\mathbf{e}$ has low weight, one can multiply $\mathbf{e}$ by a uniform random vector $\mathbf{r}$ to yield a MIPURS distribution that is the component-wise product of $\mathbf{e}$ and $\mathbf{r}$. However, if $\mathbf{e}$ often has high weight this transform requires modification; see the discussion in Section 4.

The code-offset in the exponent construction allows more flexibility in subset testing. Our analysis requires all subsets of have entropy see Definition 2.[6] The construction of Canetti et al. required an average subset to have entropy, see Definition 6. The motivation for not explicitly specifying subsets in Gen is that many physical sources are sampled along with correlated side information that is called *confidence*. Confidence information is a secondary probability distribution $\mathbf{z}$ (correlated with the reading $\mathbf{e}$) that can predict the error rate in a symbol $\mathbf{e}_i$. When $\mathbf{z}_i$ is large this means a bit of $\mathbf{e}_i$ is less likely to differ. Examples include the magnitude of a convolution in the iris [SSF19] and the magnitude of the difference between two circuit delays in ring oscillator PUFs [HRvD$^+$16].

Herder et al. [HRvD$^+$16] report that by considering bits with high confidence it is possible to reduce the effective error rate from $t = .10 \cdot n$ to $t = 3 \cdot 10^{-6} \cdot n$. Note that for a subset size of 128 and $t = .1n$ unlocking with 95% probability requires testing approximately $2 \cdot 10^6$ subsets while $t = 3 \cdot 10^{-6} \cdot n$ requires testing a single subset. This confidence information could not be used in Canetti et al's work to guide subset selection as it is correlated with $\mathbf{E}$. Our construction can securely use confidence information since it is only needed at reproduction time.

[6] Code offset in the exponent is secure if there are some low entropy subsets, the condition is that they should have negligible probability of being in a null space vector that has nonzero coordinates at just the location in the subset. However, the fraction of allowable subsets is small so we keep our condition as all subsets having entropy.

1.3 Application to Pattern Matching Obfuscation

Bishop et al. [BKM+18] show how to obfuscate a pattern $\mathbf{v}$ where each $\mathbf{v}_i \in \{0, 1, \perp\}$ indicates that the bit $\mathbf{v}_i$ should match 0, 1 or either value. The goal is to allow a user to check for input string $\mathbf{y}$, if $\mathbf{y}$ and $\mathbf{v}$ are the same on all non-wildcard positions. Their construction was stated for Reed-Solomon codes but works for any linear code. We state the construction for a random linear code: Let $|\mathbf{v}| = n$ and assume $\mathbf{A} \leftarrow (\mathbb{F}_q)^{2n\times n}$. Then for a random $\mathbf{x}$ the construction outputs the following obfuscation (for a group $\mathbb{G}_q$ of prime order q):[7]

$$\mathcal{O}_w = \left\{ o_i = \begin{cases} (g^{\mathbf{A}_{2i}\mathbf{x}}, r_{2i+1}), r_{2i+1} \leftarrow \mathbb{G}_q & \mathbf{v}_i = 1 \\ (r_{2i}, g^{\mathbf{A}_{2i+1}\mathbf{x}}), r_{2i} \leftarrow \mathbb{G}_q & \mathbf{v}_i = 0 \\ (g^{\mathbf{A}_{2i}\mathbf{x}}, g^{\mathbf{A}_{2i+1}\mathbf{x}}) & \mathbf{v}_i = \perp \end{cases} \right\}_{i=0}^{|\mathbf{v}|-1}.$$

In the above $\mathbf{A}_j$ is the jth row of $\mathbf{A}$. Bishop et al. prove security of the scheme in the generic group model. Their analysis focuses on allowing a large number of randomly placed wildcards with the uniform distribution for nonwildcard bits of $\mathbf{v}$. Most applications of string matching are on nonuniform and correlated values such as human language. We show the same construction is secure for more distributions over $\mathbf{v}$. First, we define an auxiliary variable $\mathbf{s}$ of length $2n$ that describes the placement of errors as follows:

$$s_i = \begin{cases} 10 & \text{if } v_i = 1, \\ 01 & \text{if } v_i = 0, \\ 00 & \text{if } v_i = \perp. \end{cases}$$

It is sufficient for the probability distribution over $\mathbf{s}$ to have entropy in all subsets of size n (see Definition 2). In human language, it seems subsets of bits do have this property [Sha51, BPM+92, MZ11].

In concurrent work Bartusek, Lepoint, Ma, and Zhandry [BLMZ19] present two contributions of interest to this work. They consider the pattern matching obfuscation application. Their first contribution raises the upper bound on the number of wildcards in [BKM+18] from $0.774n$ to $n - \omega(\log n)$ using a new dual form of analysis. Their analysis still considers the uniform distribution over nonwildcard positions. Thus, our analysis expands the provably secure distributions over $\mathbf{v}$. Their second contribution considers random linear codes not in the exponent, they use a modified version of the Random Linear Code (RLC) assumption defined in [IPS09]. They prove for some structured error distributions hardness of both search and decision problems. Importantly, their analysis relies on the adversary receiving only $2n$ dimensions and would not apply for our fuzzy extractor application.

1.4 Decoding in the Standard Model

Lastly, we show hardness of decoding linear codes with independent, random errors in the exponent assuming the hardness of discrete log. Prior work by Peikert showed such a result for Reed-Solomon codes [Pei06, Theorem 3.1]. We quantitatively improve Peikert's result for Reed-Solomon codes (Theorem 25) and present a similar result for random linear codes (Theorem 21).[8] Both results slightly improve parameters over Peikert's result [Pei06, Theorem 3.1]. These arguments require that a random point lies close to a codeword with noticeable probability. As q increases this probability decreases but discrete log becomes harder, creating a tension between these parameters. Peikert's result requires that $q \leq \binom{n}{k+1}/n^2$.

[7] Bishop et al. state their construction where $\mathbf{x}_0 = 0$ to allow the user to check whether they matched the pattern. In this description, we allow the user to get out a key contained in $g^{\mathbf{x}_0}$ when they are correct.

[8] Both results require the error $\mathbf{e}$ to have independent symbols, with $\mathbf{e}$ possessing t randomly chosen nonzero positions.

In an application to the fuzzy extractors reducing k leads to improved efficiency, meaning the goal is to have small k for which $k = \omega(\log n)$ see Section 1.2). This means that the upper bound on q may be just superpolynomial. Our results allow q to grow more quickly, improving the bound by a modest factor of n^2 (requiring that $q \leq \binom{n}{k+1}$).

Theorems 21 and 25 consider an adversary that performs error correction: given $g^{\mathbf{y}}$ it returns $g^{\mathbf{z}}$, where $\mathbf{z}$ is a codeword and $\mathsf{dis}(\mathbf{y}, \mathbf{z}) \leq t$. Recently, Fuchsbauer et al. [FKL18] introduced the algebraic group model which is weaker than the generic group model. From an input $g^{\mathbf{y}}$, an *algebraic* adversary produces a solution $g^{\mathbf{z}}$ along with a matrix Γ such that $g^{\mathbf{z}} = g^{\Gamma\mathbf{y}}$. The model is weaker than the generic group model as the adversary is allowed to see the elements $g^{\mathbf{y}}$ before creating Γ. A standard model adversary that decodes a linear code implies an algebraic adversary. One can find k indices where $g^{\mathbf{z}_i} = g^{\mathbf{y}_i}$. One then uses the *linear* decoding (from these indices) and encoding procedures of the code to find the coefficients such that $g^{\mathbf{z}} = g^{\Gamma\mathbf{y}}$. Thus, decoding is a problem where the algebraic model appears weaker than the generic group model. Recent work provides evidence of groups where all adversaries are algebraic assuming the existence of indistinguishability obfuscation [AHK20].

We note the wide gap between error distributions we can show in the generic group model and assuming discrete log. The main open question from this work is how much of a gap is necessary?

Organization. The remainder of the paper is organized as follows, Section 2 covers definitions and preliminaries, Section 3 presents the MIPURS condition and characterizes distributions that satisfy this condition. Sections 4 and 5 describe our applications to fuzzy extractors and pattern matching obfuscation respectively. Section 6 shows hardness of decoding high entropy errors in the standard model.

2 Preliminaries

We use boldface to represent random variables, capitals to represent random variables over matrices or sets, and corresponding plain letters to represent samples. As one notable exception, we use ψ to represent samples from $\mathbf{e}$ to avoid conflict with Euler's number. For random variables $\mathbf{x}_i$ over some alphabet $\mathcal{Z}$ we denote the tuple by $\mathbf{x} = (\mathbf{x}_1, ..., \mathbf{x}_n)$. For a vector $\mathbf{v}$ we denote the ith entry as $\mathbf{v}_i$. For a set of indices J, $\mathbf{x}_J$ denotes the restriction of $\mathbf{x}$ to the indices in J. For $m \in \mathbb{N}$, we let $[m] = \{1, \ldots, m\}$, so that $[0] = \emptyset$. We use the notation $\mathrm{span}(S)$ to denote the linear span of a set S of vectors and apply the notation to sequences of vectors without any special indication: if $F = (f_1, \ldots, f_m)$ is a sequence of vectors, $\mathrm{span}(F) = \mathrm{span}(\{f_i \mid i \in [m]\})$.

The *min-entropy* of a random variable $\mathbf{x}$ is $\mathrm{H}_\infty(\mathbf{x}) = -\log(\max_x \Pr[\mathbf{x} = x])$. The *average (conditional)* min-entropy [DORS08, Section 2.4] of $\mathbf{x}$ given $\mathbf{y}$ is

$$\tilde{\mathrm{H}}_\infty(\mathbf{x} \mid \mathbf{y}) = -\log\left(\mathop{\mathbb{E}}_{y \in \mathbf{y}} \max_{x} \Pr[\mathbf{x} = x \mid \mathbf{y} = y]\right).$$

For a metric space $(\mathcal{M}, \mathsf{dis})$, the *(closed) ball of radius t around x* is the set of all points within radius t, that is, $B_t(x) = \{y \mid \mathsf{dis}(x, y) \leq t\}$. If the size of a ball in a metric space does not depend on x, we denote by $\mathsf{Vol}(t)$ the size of a ball of radius t. We consider the Hamming metric. Let $\mathcal{Z}$ be a finite set and consider vectors in $\mathcal{Z}^n$, then $\mathsf{dis}(x, y) = |\{i \mid x_i \neq y_i\}|$. For this metric, we denote volume as $\mathsf{Vol}(n, t, |\mathcal{Z}|)$ and $\mathsf{Vol}(n, t, \mathcal{Z}) = \sum_{i=0}^{t} \binom{n}{i}(|\mathcal{Z}| - 1)^i$. For a vector in $x \in \mathbb{Z}_q^n$ let $\mathsf{wt}(x) = |\{i | x_i \neq 0\}|$. U_n denotes the uniformly distributed random variable on $\{0, 1\}^n$. Logarithms are base 2. We denote the vector of all zero elements as 0. We let $\cdot_c$ denote component-wise multiplication. In our theorems we consider a security parameter γ, when we use the term negligible and super polynomial, we assume other parameters are functions of γ. We elide this notation the dependence of other parameters on γ.

3 When is code offset in the exponent hard?

In this section, we introduce the *Maximum Inner Product Unpredictable over Random Subspace* (MIPURS) condition, show that code offset in the exponent is secure in the generic group model given this condition, and show distributions of interest that satisfy MIPURS.

Definition 1. *Let* $\mathbf{e}$ *be a random variable taking values in* $\mathbb{F}_q^n$ *and let* $\mathbf{A}$ *be uniformly distributed over* $\mathbb{F}_q^{n\times k}$ *and independent of* $\mathbf{e}$*. We say that* $\mathbf{e}$ *is a* $(k,\beta)-\mathsf{MIPURS}$ *distribution if for all random variables* $\mathbf{b}\in\mathbb{F}_q^n, \mathbf{g}\in\mathbb{F}_q$ *independent of* $\mathbf{e}$ *(but depending arbitrarily on* $\mathbf{A}$ *and each other)*

$$\mathop{\mathbb{E}}_{\mathbf{A}}\left[\Pr\left[\langle \mathbf{b},\mathbf{e}\rangle=\mathbf{g} \textit{ and } \mathbf{b}\in\mathtt{null}(\mathbf{A})\setminus\mathbf{0}\right]\right]\leq\beta\,.$$

To see the equivalence between this definition and the game presented in the introduction, $\mathbf{b}$ can be seen as encoding the "adversary" and quantifying over all $\mathbf{b}$ is equivalent to considering all information-theoretic adversaries. Before showing sufficiency of MIPURS, we review some notation from the generic group model. The model is reviewed in detail in Appendix A. Let $\mathbb{G}$ be a group of prime order q. For each element $r\in\mathbb{G}$ in the standard game, rather than receiving r, the adversary receives a handle $\sigma(r)$ where σ is a random function with a large range. The adversary is given access to an oracle, which we denote as $\mathcal{O}_{\mathbb{G}}^{\sigma}$, which given $x=\sigma(r_1), y=\sigma(r_2)$ computes $\sigma(\sigma^{-1}(x)+\sigma^{-1}(y))$; when σ can be inferred from context, we write $\mathcal{O}_{\mathbb{G}}$. Since the adversary receives random handles they cannot infer anything about the underlying group elements except using the group operation and testing equality. Below, the adversary is provided the code directly in the group, not its image in the handle space.

Theorem 2. *Let* γ *be a security parameter. Let* q *be a prime and* $n,k\in\mathbb{Z}^+$ *with* $k\leq n\leq q$*. Let* $\mathbf{A}\in\mathbb{F}_q^{n\times k}$ *and* $\mathbf{x}\in\mathbb{F}_q^k$ *be uniformly distributed. Let* $\mathbf{e}$ *be a* $(k,\beta)-\mathsf{MIPURS}$ *distribution. Let* $\mathbf{u}\in(\mathbb{F}_q)^n$ *be uniformly distributed. Let* Σ *be the set of random functions with domain of size* q *and range of size* q^3*. Then for all adversaries* $\mathcal{D}$ *making at most* m *queries*

$$\left|\Pr_{\sigma\overset{\$}{\leftarrow}\Sigma}\left[\mathcal{D}^{\mathcal{O}_{\mathbb{G}}}(\mathbf{A},\sigma(\mathbf{A}\mathbf{x}+\mathbf{e}))=1\right]-\Pr[\mathcal{D}^{\mathcal{O}_{\mathbb{G}}}(\mathbf{A},\sigma(\mathbf{u}))=1]\right|<\mu\left(\frac{2}{q}+\beta\right)+\frac{1}{q}\leq\mu\left(\frac{3}{q}+\beta\right)$$

for $\mu=\left((m+n+2)(m+n+1)\right)^2/2$*. In particular, if* $1/q=\mathtt{ngl}(\gamma), n,m=\mathtt{poly}(\gamma)$*, and* $\beta=\mathtt{ngl}(\gamma)$ *then the statistical distance between the two cases is* $\mathtt{ngl}(\gamma)$*.*

In the above, the final $1/q$ term represents the small probability of σ not being $1-1$. The proof of Theorem 2 is relatively straightforward and in Appendix A. Our proof uses the simultaneous oracle game introduced by Bishop et al. [BKM+18, Section 4]. The rest of this section is dedicated to understanding what types of distributions are MIPURS.

3.1 Characterizing MIPURS

Definition 1 of MIPURS is admittedly unwieldily. It considers a property of a distribution $\mathbf{e}\in\mathbb{F}_q^n$ with respect to a random matrix. We turn to characterizing distributions that satisfy MIPURS. We begin with distributions whose analysis is straightforward (the general entropy case is in Section 3.2). Throughout, we consider a prime order group $\mathbb{G}$ of prime size q, a random linear code $\mathbf{A}\in\mathbb{F}_q^{n\times k}$ and the null space $\mathbf{B}\overset{\text{def}}{=}\mathtt{null}(\mathbf{A})$.

Independent Sources. In most versions of LWE, each error coordinate is independently distributed and contributes some entropy. Examples include the discretized Gaussian introduced by Regev [Reg05, Reg10], and a uniform interval introduced by Döttling and Müller-Quade [DMQ13]. We show that these distributions fit within our MIPURS characterization.

Lemma 3. *Let* $\mathbf{e} = \mathbf{e}_1, \dots, \mathbf{e}_n \in \mathbb{F}_q^n$ *be a distribution where each* $\mathbf{e}_i$ *is independently sampled. Let* $\alpha = \min_{1\le i\le n} \mathrm{H}_\infty(\mathbf{e}_i)$*. For any* $k \le n$*,* $\mathbf{e}$ *is a* $(k,\beta) - \mathsf{MIPURS}$ *distribution for* $\beta = 2^{-\alpha}$*.*

Proof. Consider a fixed element $b \neq 0$ in $\mathbf{B}$. Since the components of $\mathbf{e}$ are independent, predicting $\langle b, \mathbf{e}\rangle$ is at least as hard as predicting $\mathbf{e}_i$ for each i such that $\mathbf{b}_i \neq 0$. This can be seen by fixing b and $\mathbf{e}_j$ for $j \neq i$ and noting that the value of $\mathbf{e}_i$ then uniquely determines $\langle b, \mathbf{e}\rangle$. Since $b \neq 0$ there exists at least one such i. Thus,

$$\Pr_{\mathbf{B}}\left[\max_g \max_{b\in\mathbf{B}\setminus 0} \Pr_{\mathbf{e}}[\langle b, \mathbf{e}\rangle = g]\right] \le 2^{-\alpha} \overset{\text{def}}{=} \beta.$$

□

Location Sources. The second family of error distributions we consider are $\mathbf{e}'$ given by the coordinatewise product of a uniform vector $\mathbf{r} \in \mathbb{F}_q^n$ and a "selection vector" $\mathbf{e} \in \{0,1\}^n$: that is, $\mathbf{e}'_i = \mathbf{r}_i \cdot_c \mathbf{e}_i$ where $\mathbf{e}$ is assumed to be unpredictable on all large enough subsets ($\cdot_c$ is component-wise multiplication). Distributions with entropic subsets are important for applications as discussed in the Introduction and Sections 4 and 5. More formally, we introduce a notion called subset entropy:

Definition 2. *Let a source* $\mathbf{e} = \mathbf{e}_1, \dots, \mathbf{e}_n$ *consist of* n*-bit binary strings. For some parameters* k*,* α *we say that the source* $\mathbf{e}$ *is has* (α, k)***-entropy subsets*** *if* $\mathrm{H}_\infty(\mathbf{e}_{j_1}, \dots, \mathbf{e}_{j_k}) \ge \alpha$ *for any* $1 \le j_1, \dots, j_k \le n$*.*

Lemma 4. *Let* $\ell \in \mathbb{N}$ *and* $k \in \mathbb{Z}^+$*. Let* $\mathbf{e} \in \{0,1\}^n$ *be a distribution with* $(\alpha, k-\ell)$ *entropy subsets. Define the distribution* $\mathbf{e}'$ *as the coordinatewise product of a uniform vector* $\mathbf{r} \in \mathbb{F}_q^n$ *and* $\mathbf{e}$*: that is,* $\mathbf{e}'_i = \mathbf{e}_i \cdot_c \mathbf{r}_i$*. Then the distribution* $\mathbf{e}'$ *is a* MIPURS *distribution for* $(k-\ell, \beta)$ *for*

$$\beta = 2^{-\alpha} + \left(\frac{(k-\ell)\binom{n}{k-\ell-1}}{q^{\ell+1}}\right).$$

Proof. We start by bounding the "minimum distance" of $\mathbf{B}$, that is, the minimum weight of a non-zero element of $\mathbf{B} = \mathtt{null}(\mathbf{A})$. Observe that the number of vectors in $\mathbb{F}_q^n$ of weight less than $k-\ell$ is

$$\sum_{j=0}^{k-\ell-1} \binom{n}{j} q^j \le (k-\ell)\binom{n}{k-\ell-1} q^{k-\ell-1}.$$

The probability that any fixed, nonzero vector lies x in $\mathtt{null}(\mathbf{A})$ is q^{-k}, as it must annihilate k independent, uniform linear equations. (That is, $\sum_i x_i \mathbf{A}_{is} = 0$ for each $1 \le s \le k$.) Thus

$$\mathbb{E}[|\{w \neq 0 \in \mathtt{null}(\mathbf{A}) \mid \mathsf{wt}(w) < k-\ell\}|] \le (k-\ell)\binom{n}{k-\ell-1} q^{-\ell-1}. \tag{1}$$

By Markov's inequality, the probability that there is at least one such large weight vector in $\mathtt{null}(\mathbf{A})$ is no more than the expected number of such vectors. Hence

$$\Pr[\exists w \in \mathtt{null}(\mathbf{A}) \setminus 0, \mathsf{wt}(w) < k-\ell] \le (k-\ell)\binom{n}{k-\ell-1} q^{-\ell-1}.$$

For some $\mathbf{b}$ in the span of $\mathbf{B}$ with weight at least $k-\ell$, consider the product $\langle \mathbf{b}, \mathbf{e}' \rangle = \sum_{i=1}^{n} \mathbf{b}_i \cdot \mathbf{e}_i \cdot \mathbf{r}_i$. Define $\mathcal{I}$ as the set of nonzero coordinates in $\mathbf{b}$. With probability at least $1-2^{-\alpha}$ there is some nonzero coordinate in $\mathbf{e}_{\mathcal{I}}$. Conditioned on this fact this means that at least one value $\mathbf{r}_i$ is included in the inner product. Thus, the inner product acts as a one time pad. The argument concludes by assuming perfect predictability when there exists $\mathbf{b}$ in $\mathbf{B}$ with weight of at most $k-\ell-1$. □

3.2 Primary Technical Contribution - MIPURS is hard for high entropy

We now turn to the general entropy condition: MIPURS is hard for all distribution where the min-entropy exceeds $\log q^{n-k}$ (by a super logarithmic amount). As described in the Introduction, requiring min-entropy greater than q^{n-k} is necessary. For conciseness, we introduce $\kappa \overset{\text{def}}{=} n-k$.

The adversary is given a generating matrix of the code, $\mathbf{A}$; this determines $\mathbf{B} = \texttt{null}(\mathbf{A})$. Our proof is divided into three parts. Denote by E a set of possible error vectors.

1. Theorem 5: We show that the number of vectors $\psi \in E$ that are likely to have 0 inner product with an adversarially chosen vector in $\mathbf{B}$ is small. Intuitively, we show that this set is "not much larger than a κ-dimensional subspace."
2. Theorem 10: We then show it is difficult to predict the value of the inner product: even if the adversary may select arbitrarily coupled $\mathbf{b}$ and $\mathbf{g}$, it is difficult to achieve $\langle \mathbf{b}, \psi \rangle = \mathbf{g}$.
3. Lemma 12: We show that any distribution $\mathbf{e}$ with sufficient entropy cannot lie in the set of *predictable* error vectors E with high probability.

We codify the set of possible adversarial strategies by introducing a notion of *κ-induced random variables.* For the moment, we assume that $\mathbf{B}$ is a uniformly selected subspace of dimension exactly κ; at the end of the proof we remove this restriction to apply these results when $\mathbf{B}$ has the distribution given by $\texttt{null}(\mathbf{A})$ (Corollary 13).

Definition 3. *Let $\mathbf{b}$ be a random variable taking values in $\mathbb{F}_q^n$. Let $\mathbf{B}$ be a (typically dependent) random variable that is uniform on the collection of κ-dimensional subspaces of $\mathbb{F}_q^n$. We say that $\mathbf{b}$ is κ-*induced *if $\mathbf{b} \in \mathbf{B}$ and $\mathbf{b} \neq \mathbf{0}$ with certainty:* $\Pr[\mathbf{b} \in \mathbf{B} \wedge \mathbf{b} \neq \mathbf{0}] = 1$*. Note that the random variables $\mathbf{B}$ and $\mathbf{b}$ are necessarily dependent (unless $n = \kappa$).*

It suffices to consider the maximum probability in Definition 1 with respect to κ-induced random variables. This is because for any $\mathbf{b}$ that is not κ-induced we can find another $\mathbf{b}$ that is κ induced that does no worse in the game in Definition 1. For example when $\mathbf{b}$ is not in $\mathbf{B}$ or is the zero vector, one can replace $\mathbf{b}$ with a random element in the span of $\mathbf{B}$.

We now show that if the set E is large enough there is no strategy for $\mathbf{b}$ that guarantees $\langle \mathbf{b}, \psi \rangle = 0$ with significant probability. The next theorem (Thm. 10) will, more generally, consider prediction of the inner product itself. For a κ induced random variable $\mathbf{b}$, define

$$E_\epsilon^{(\mathbf{b},0)} = \left\{ f \in \mathbb{F}_p^n \;\middle|\; \Pr_{\mathbf{b}}[\langle \mathbf{b}, f \rangle = 0] \geq \epsilon \right\}.$$

When $\mathbf{b}$ can be inferred from context, we simply refer to this set as E_ϵ. Then define $P_{\kappa,\epsilon} = \max_{\mathbf{b}} |E_\epsilon^{(\mathbf{b},0)}|$ where the maximum is over all κ-induced random variables in $\mathbb{F}_q^n$.

Theorem 5. *Let q be a prime and let $d > 1$, $\kappa, m, \eta \in \mathbb{Z}^+$ be parameters for which $\kappa \leq n$. Then assuming $P_{\kappa,\epsilon} > d \cdot q^\kappa$ we must have*

$$\epsilon \leq \left(\frac{\kappa+\eta}{m}\right) + \binom{m}{\kappa}\left(\binom{m}{\eta}\left(\frac{1}{d}\right)^\eta + \left(\frac{2}{q}\right)\right).$$

Before proving Theorem 5, we introduce and prove two combinatorial lemmas (6 and 8). We then proceed with the proof of Theorem 5. The major challenge is that the set E_ϵ (for a particular $\mathbf{b}$) is typically not a linear subspace; these results show that is has reasonable "approximate linear" structure. We begin with the notion of *linear density* to measure, intuitively, how close the set is to linear.

Definition 4. *The* ℓ-linear density *of a sequence of vectors $F = (f^1, \ldots, f^m)$, with each $f^i \in \mathbb{F}_q^n$, is the maximum number of entries that are covered by a subspace of dimension ℓ. Formally,*

$$\Delta^\ell(F) = \max_{V, \dim(V)=\ell} |\{i \mid f^i \in V\}|.$$

Lemma 6. *Let q be a prime and let $n, \ell \in \mathbb{Z}^+$ satisfy $\ell \leq n$. Let $E \subset \mathbb{F}_q^n$ satisfy $|E| \geq q^\ell$ and let $\mathbf{F} = (\mathbf{f}^1, \ldots, \mathbf{f}^m)$ be a sequence of uniformly and independently chosen elements of E. Define d so that $|E| = dq^\ell$; then for any $\eta \geq 0$,*

$$\Pr_{\mathbf{F}}[\Delta^\ell(\mathbf{F}) \geq \ell + \eta] \leq \binom{m}{\ell}\binom{m-\ell}{\eta}\left(\frac{1}{d}\right)^\eta.$$

Proof. By the definition of linear density, if $\Delta^\ell(\mathbf{F}) \geq \ell + \eta$ there must be at least one subset of $\ell + \eta$ indices $I \subset [m]$ so that $\{\mathbf{f}^i \mid i \in I\}$ is contained in a subspace of dimension ℓ. In order for a subset I to have this property, there must be a partition of I into a disjoint union $S \cup L$, where S has cardinality ℓ and T indexes the remaining η "lucky" vectors that lie in the span of the vectors given by S. Formally, $\forall t \in T, \mathbf{f}^t \in \operatorname{span}(\{\mathbf{f}^s \mid s \in S\})$.

Fix, for the moment, ℓ indices of $\mathbf{F}$ to identify a candidate subset of vectors to play the role of S and η indices of $\mathbf{F}$ to identify a candidate set T. The probability that each of the η vectors indexed by T lie in the space spanned by S is clearly no more than $(p^\ell/|E|)^\eta \leq (1/d)^\eta$. Taking the union bound over these choices of indices completes the argument: the probability of a sequence is no more than

$$\binom{m}{\ell}\binom{m-\ell}{\eta}d^{-\eta},$$

as desired. □

Before introducing our second combinatorial lemma (Lem 8), we need a Claim bounding the probability of a fixed subspace having a nontrivial intersection with a random subspace.

Claim 7. *Let q be a prime and $\kappa, n \in \mathbb{N}$ with $\kappa \leq n$. Let $\mathbf{V}$ be a random variable uniform on the set of all κ-dimensional subspaces of $\mathbb{F}_q^n$. Let W be a fixed subspace of dimension ℓ. Then*

$$\Pr[\mathbf{V} \cap W \neq \{0\}] \leq q^{\kappa+\ell-(n+1)} \cdot \left(\frac{q}{q-1}\right).$$

Proof. Let $\mathcal{L}$ denote the set of all 1-dimensional subspaces in W. Each 1-dimensional subspace is described by an equivalence class of $q-1$ vectors under the relation $x \sim y \Leftrightarrow \exists \lambda \in \mathbb{F}_q^*, \lambda x = y$. Thus $|\mathcal{L}| = (q^\ell - 1)/(q-1) \leq q^{\ell-1}(q/(q-1))$. Then

$$\begin{aligned}\Pr[\mathbf{V} \cap W \neq \{\mathbf{0}\}] = \Pr[\exists L \in \mathcal{L}, L \subset \mathbf{V}] &\leq \sum_{L \in \mathcal{L}} \Pr[L \subset \mathbf{V}] \\ &\leq |\mathcal{L}| \max_{v \in \mathbb{F}_q^n \setminus \{\mathbf{0}\}} \Pr[v \in \mathbf{V}] \leq q^{\kappa+\ell-(n+1)} \left(\frac{q}{q-1}\right),\end{aligned}$$

where we recall the fact that for any particular fixed nonzero vector v,

$$\Pr[v \in \mathbf{V}] = \frac{q^\kappa - 1}{q^n - 1} \leq q^{\kappa - n}. \qquad \square$$

Lemma 8. *Let q be a prime, let $\ell, \kappa, n \in \mathbb{Z}^+$ satisfy $\ell, \kappa \leq n$. Let $F = (f^1, \ldots, f^m)$ be a sequence of elements of $\mathbb{F}_q^n$ with $\dim(\operatorname{span}(F)) \geq \ell$. Then, for any κ-induced random variable $\mathbf{b}$ taking values in $\mathbb{F}_q^n$,*

$$\Pr_{\mathbf{b}}\big[|\{i \mid \langle \mathbf{b}, f^i \rangle = 0\}| \geq \Delta^\ell(F)\big] \leq \binom{m}{\ell} q^{\kappa-\ell-1} \left(\frac{q}{q-1}\right) \leq 2\binom{m}{\ell} q^{\kappa-\ell-1}.$$

Proof. Let $\mathcal{V}_F$ denote the collection of all ℓ-dimensional subspaces of $\mathbb{F}_q^n$ spanned by subsets of elements in the sequence F. That is,

$$\mathcal{V}_F = \{V \mid V = \operatorname{span}(\{f^i \mid i \in I\}), I \subset [m], \dim(V) = \ell\}.$$

Then $|\mathcal{V}_F| \leq \binom{m}{\ell}$, as each such subspace is spanned by at least one subset of F of size ℓ. As $\dim(\operatorname{span}(F)) \geq \ell$, the set $\mathcal{V}_F$ is nonempty.

Observe that if $I \subset [m]$ has cardinality at least $\Delta^\ell(F)$ then, by definition, $\dim(\operatorname{span}(\{f^i \mid i \in I\})) \geq \ell$; otherwise, an additional element of F could be added to the set indexed by I to yield a set of size exceeding $\Delta^\ell(F)$ which still lies in a subspace of dimension ℓ (contradicting the definition of Δ^ℓ). Note in the case that $m = \ell$ (and there is no element to add) then $\Delta^\ell(F) = \ell = \dim(\operatorname{span}(\{f^i \mid i \in I\}))$. Thus, if $I \subset [m]$ has cardinality at least $\Delta^\ell(F)$, there must be some $V \in \mathcal{V}_F$ for which $V \subset \operatorname{span}(\{f^i \mid i \in I\})$. In particular

$$\begin{aligned}\Pr_{\mathbf{b}}\big[|\{f^i \in F \mid \langle \mathbf{b}, f^i \rangle = 0\}| \geq \Delta^\ell(F)\big] &\leq \Pr_{\mathbf{b}}\big[\exists V \in \mathcal{V}_F, \forall v \in V, \langle v, \mathbf{b} \rangle = 0\big] \\ &\leq \sum_{V \in \mathcal{V}_F} \Pr_{\mathbf{b}}\big[\forall v \in V, \langle v, \mathbf{b} \rangle = 0\big] = \sum_{V \in \mathcal{V}_F} \Pr_{\mathbf{b}}\big[\mathbf{b} \in V^\perp\big],\end{aligned}$$

where we have adopted the notation $V^\perp = \{w \mid \forall v \in V, \langle v, w \rangle = 0\}$. Recall that when V is a subspace of dimension ℓ, $V^\perp$ is a subspace of dimension $n - \ell$. To complete the proof, we recall that $\mathbf{b}$ is κ-induced, so that there is an associated random variable $\mathbf{B}$, uniform on dimension κ subspaces, for which $\mathbf{b} \in \mathbf{B}$ with certainty; applying Claim 7 we may then conclude

$$\begin{aligned}\sum_{V \in \mathcal{V}_F} \Pr_{\mathbf{b}}\big[\mathbf{b} \in V^\perp\big] \leq \sum_{V \in \mathcal{V}_F} \Pr_{\mathbf{B}}\big[\mathbf{B} \cap V^\perp \neq \{\mathbf{0}\}\big] &\leq \binom{m}{\ell} q^{\kappa+(n-\ell)-(n+1)} \frac{q}{q-1} \\ &= \binom{m}{\ell} q^{\kappa-\ell-1} \left(\frac{q}{q-1}\right).\end{aligned}$$

$\square$

Proof of Theorem 5. Now we analyze the relationship between our two paramenters of interest: ϵ and d. Fix some $\epsilon > 0$. Let $\mathbf{b}$ be a κ-induced random variable for which $|E_\epsilon^{(\mathbf{b},0)}| = P_{\kappa,\epsilon}$ and let $\mathbf{B}$ be the coupled variable, uniform on subspaces, for which $\mathbf{b} \in \mathbf{B}$.

For the purposes of analysis we consider a sequence of m vectors chosen independently and uniformly from $E_\epsilon = E_\epsilon^{(\mathbf{b},0)}$ with replacement; we let $\mathbf{F} = (\mathbf{f}^1, \ldots, \mathbf{f}^m)$ denote the set of vectors so chosen. We study the expectation of the number of vectors in $\mathbf{F}$ that are orthogonal to $\mathbf{b}$. We first give an immediate lower bound by linearity of expectation and the definition of E_ϵ:

$$\mathop{\mathbb{E}}_{\mathbf{b},\mathbf{F}}[|\{\mathbf{f}^i \in F \mid \langle \mathbf{b}, \mathbf{f}^i \rangle = 0\}|] \geq \epsilon \cdot m\,.$$

We now infer an upper bound on this expectation using Lemmas 6 and 8. We say that the samples $\mathbf{F}$ from E_ϵ are *bad* if $\Delta^\kappa(\mathbf{F}) \geq \kappa + \eta$. The probability of this *bad* event is no more than

$$\binom{m}{\kappa}\binom{m-\kappa}{\eta}\left(\frac{1}{d}\right)^\eta$$

by Lemma 6. For *bad* selections, we crudely upper bound the expectation by m; for *good* selections we further split the expectation based on the random variable $\mathbf{B}$. We say that $\mathbf{B}$ is *terrible* (for a fixed $F = (f^1, ..., f^m)$) if there exists some $b \in \mathbf{B}$ such that $|\{f^i \in F \mid \langle b, f^i \rangle = 0\}| \geq \Delta^\kappa(F)$. Otherwise, $\mathbf{B}$ is *great*. The probability of a *terrible* selection of $\mathbf{B}$ is bounded above by $(2/q)\binom{m}{\kappa}$ in light of Lemma 8 (applied with $\ell = \kappa$). In this pessimistic case (that $\mathbf{B}$ is *terrible*), we again upper bound the expectation by m. Then if the experiment is neither *bad* nor *terrible*, we may clearly upper bound the expectation by $\kappa + \eta$. So, for any $\eta > 0$ we conclude that

$$\mathop{\mathbb{E}}_{\mathbf{b},\mathbf{B},F}[|\{f_i \in F \mid \langle \mathbf{b}, f_i \rangle = 0\}|] \leq (\kappa + \eta) + m\left(\binom{m}{\kappa}\binom{m-\kappa}{\eta}\left(\frac{1}{d}\right)^\eta + \frac{2}{q}\binom{m}{\kappa}\right)$$

and hence that

$$\epsilon \leq \left(\frac{\kappa+\eta}{m}\right) + \binom{m}{\kappa}\left(\binom{m}{\eta}\left(\frac{1}{d}\right)^\eta + \frac{2}{q}\right)\,. \qquad \square$$

Corollary 9. *Let κ and n be parameters satisfying $1 \leq \kappa < n$ and let q be a prime such that $q \geq 2^{4\kappa}$. Then for $\epsilon \geq 5eq^{-1/(2(\kappa+1))}$ we have $P_{\kappa,\epsilon} \leq 5eq^\kappa/\epsilon$. In particular, for such ϵ and any κ-induced $\mathbf{b}$, the set $|E_\epsilon^{(\mathbf{b},0)}| \leq 5eq^\kappa/\epsilon$.*

Proof. Consider parameters for Theorem 5 that satisfy the following:

$$1 < d \leq q^{1/(2(\kappa+1))}, \qquad m = \frac{d\eta}{2e}, \qquad \text{and} \qquad \eta = \log q\,.$$

First note that $\kappa < 4\kappa \leq \log q = \eta$ (as $q \geq 2^{4\kappa}$). Then, consider a set $E_\epsilon^{(\mathbf{b},0)}$ for some $\mathbf{b}$. We have

$$\begin{aligned}\epsilon \leq \left(\frac{\kappa+\eta}{m}\right) + \binom{m}{\kappa}\left(\left(\frac{me}{\eta d}\right)^\eta + \frac{2}{q}\right) &\leq \left(\frac{2\eta}{m}\right) + 3\binom{m}{\kappa}q^{-1} \\ &\leq \left(\frac{4e}{d}\right) + 3\binom{d\eta/2e}{\kappa}q^{-1} \leq \underbrace{\left(\frac{4e}{d}\right) + 3\left(\frac{d\eta}{2\kappa}\right)^\kappa q^{-1}}_{(\dagger)}\,.\end{aligned}$$

Since $q \geq 2^{4\kappa}$, we may write $q = 2^{2\alpha\kappa}$ for some $\alpha \geq 2$ and it follows that

$$\left(\frac{\log q}{\kappa}\right)^{\kappa} = (2\alpha)^{\kappa} \leq (2^{\alpha})^{\kappa} = \sqrt{q}$$

because $2\alpha \leq 2^{\alpha}$ for all $\alpha \geq 2$. In light of this, consider the second term in the expression (†) above:

$$3\left(\frac{d\eta}{2\kappa}\right)^{\kappa} q^{-1} \leq \frac{3}{2}\left(\frac{d\eta}{\kappa}\right)^{\kappa} q^{-1} \leq \frac{3}{2}\left(\frac{d^{\kappa}}{\sqrt{q}}\right) \cdot \left(\left(\frac{\log q}{\kappa}\right)^{\kappa} \frac{1}{\sqrt{q}}\right) \leq \frac{3}{2d} \leq \frac{e}{d}\,.$$

We conclude that for any $1 < d \leq q^{1/(2(\kappa+1))}$, $P_{\kappa,\epsilon} \geq dq^{k} \implies \epsilon \leq 5e/d$. Observe then that for any $\epsilon > 5e/q^{1/(2(\kappa+1))}$ we may apply the argument above to $P_{\kappa,\epsilon}$ with $d = 5e/\epsilon$ and conclude that $P_{\kappa,\epsilon} \leq 5eq^{\kappa}/\epsilon$. □

Predicting Arbitrary Values. We now show that the adversary cannot due much better than Theorem 5 even if the task is predicting the value $\langle \mathbf{b}, \cdot \rangle$.

Theorem 10. *Let* $\mathbf{b}$ *be a* κ*-induced random variable in* $\mathbb{F}_q^n$ *and let* $\mathbf{g}$ *be a random variable over* $\mathbb{F}_q$ *(arbitrarily correlated with* $\mathbf{b}$*). For* $\epsilon > 0$ *we generalize the notation above so that*

$$E_{\epsilon}^{(\mathbf{b},\mathbf{g})} = \left\{ f \in \mathbb{F}_q^n \,\middle|\, \Pr_{\mathbf{b},\mathbf{g}}[\langle \mathbf{b}, f \rangle = \mathbf{g}] \geq \epsilon \right\}.$$

Then $|E_{\epsilon^2/8}^{(\mathbf{b},0)}| \geq \frac{\epsilon^2}{8}|E_{\epsilon}^{(\mathbf{b},\mathbf{g})}|$.

Proof. For an element $\psi \in E_{\epsilon}^{(\mathbf{b},\mathbf{g})}$, define $F_{\psi} = \{(f, \langle f, \psi \rangle) \mid f \in \mathbb{F}_p^n\}$. Note that $\Pr_{\mathbf{b},\mathbf{g}}[(\mathbf{b},\mathbf{g}) \in F_{\psi}] \geq \epsilon$ by assumption. For any $\delta < \epsilon$, there is a subset $F^* \subset E_{\epsilon}^{(\mathbf{b},\mathbf{g})}$ for which (i.) $|F^*| \leq 1/\delta$, and (ii.) for any $\psi \in E_{\epsilon}^{(\mathbf{b},\mathbf{g})}$,

$$\Pr_{\mathbf{b},\mathbf{g}}\left[(\mathbf{b},\mathbf{g}) \in \left(F_{\psi} \cap \left(\bigcup_{f' \in F^*} F_{f'}\right)\right)\right] \geq \epsilon - \delta\,.$$

To see this, consider incrementally adding elements of $E_{\epsilon}^{(\mathbf{b},\mathbf{g})}$ into F^* in so as to greedily increase

$$\Pr_{\mathbf{b},\mathbf{g}}\left[(\mathbf{b},\mathbf{g}) \in \bigcup_{f' \in F^*} F_{f'}\right].$$

If this is process is carried out until no $\psi \in E_{\epsilon}^{(\mathbf{b},\mathbf{g})}$ increases the total probability by more than δ, then it follows that every F_{ψ} intersects with the set with probability mass at least $\epsilon - \delta$, as desired. Note also that this termination condition is achieved after including no more than $1/\delta$ sets. It follows that for any $\psi \in E_{\epsilon}^{(\mathbf{b},\mathbf{g})}$,

$$\mathop{\mathbb{E}}_{f' \in F^*} \Pr_{\mathbf{b}}[\langle \mathbf{b}, \psi \rangle = \langle \mathbf{b}, f' \rangle] \geq (\epsilon - \delta)\delta$$

and hence

$$\mathop{\mathbb{E}}_{f' \in F^*} \mathop{\mathbb{E}}_{\psi \in E_{\epsilon}^{(\mathbf{b},\mathbf{g})}} \Pr_{\mathbf{b}}[\langle \mathbf{b}, \psi \rangle = \langle \mathbf{b}, f' \rangle] \geq (\epsilon - \delta)\delta\,.$$

Then there exists an f^* for which

$$\mathop{\mathbb{E}}_{\psi \in E_\epsilon^{(\mathbf{b},\mathbf{g})}} \Pr[\langle \mathbf{b}, \psi \rangle = \langle \mathbf{b}, f^* \rangle] \geq (\epsilon - \delta)\delta\,.$$

Setting $\delta = \epsilon/2$ and we see that

$$\mathop{\mathbb{E}}_{\psi \in E_\epsilon^{(\mathbf{b},\mathbf{g})}} \Pr[\langle \mathbf{b}, \psi \rangle = \langle \mathbf{b}, f^* \rangle] \geq \frac{\epsilon^2}{4}\,.$$

Using this expectation (of a probability), we bound the probability it is greater than 1/2 its mean. As the inner product is bi-linear,

$$\Pr_{\mathbf{b}} \left[\Pr_{\phi \in E_\epsilon^{(\mathbf{b},\mathbf{g})}} [\langle \mathbf{b}, \psi - f^* \rangle = 0] \geq \frac{\epsilon^2}{8} \right] \geq \frac{\epsilon^2}{8}\,.$$

Thus, by noting that the set $\{\psi - f^* \mid \psi \in E_\epsilon^{(\mathbf{b},\mathbf{g})}\}$ must be a subset of $E_{\epsilon^2/8}^{(\mathbf{b},0)}$, we can directly translate to the claim of the theorem, $|E_{\epsilon^2/8}^{(\mathbf{b},0)}| \geq (\epsilon^2/8)|E_\epsilon^{(\mathbf{b},\mathbf{g})}|\,.$ □

With the language and settings of this last Theorem, applying Corollary 9 to appropriately control $|E_{\epsilon^2/8}^{(\mathbf{b},0)}|$ yields the following bound on $|E_\epsilon^{(\mathbf{b},\mathbf{g})}|$.

Corollary 11. *Let κ and n be parameters satisfying $1 \leq \kappa < n$ and let q be a prime such that $q \geq 2^{4\kappa}$. Let $\mathbf{b}$ be any κ-induced random variable in $\mathbb{F}_q^n$ and $\mathbf{g}$ any random variable in $\mathbb{F}_q$. Then for any $\epsilon \geq 11q^{-1/(4(\kappa+1))}$ it holds that*

$$|E_\epsilon^{(\mathbf{b},\mathbf{g})}| \leq \frac{8}{\epsilon^2}\frac{5eq^\kappa}{\epsilon^2/8} = \frac{320eq^\kappa}{\epsilon^4}.$$

This implies all high min-entropy distributions are not predictable in the above game.

Lemma 12. *Let $\mathbf{b}$ be a κ-induced random variable in $\mathbb{F}_q^n$. Let $\mathbf{g}$ be an arbitrary random variable in $\mathbb{F}_q$. Let $\mathbf{e}$ be a random variable with $\mathrm{H}_\infty(\mathbf{e}) = s$. Let $E_\epsilon^{(\mathbf{b},\mathbf{g})}$ be as defined in Theorem 10. Then for $\epsilon > 0$*

$$\Pr_{\psi \leftarrow \mathbf{e},\mathbf{b},\mathbf{g}} [\langle \mathbf{b}, \psi \rangle = \mathbf{g}] \leq 2^{-s}|E_\epsilon^{(\mathbf{b},\mathbf{g})}| + \epsilon\,.$$

Proof. Our predictable set $E_\epsilon = E_\epsilon^{(\mathbf{b},\mathbf{g})}$ gives us no guarantee on the instability of the inner product. If $\psi \in E_\epsilon$ then we upper bound the probability by 1. Because $\mathbf{e}$ has min-entropy s, we know that no element is selected with probability greater than 2^{-s}, thus the probability of a lying inside a set of size $|E_\epsilon|$ is at most $|E_\epsilon|/2^s$. Outside of our predictable set, we know that the probability of a stable inner product cannot be greater than ϵ by definition of E_ϵ. Therefore if ψ does not fall in the predictable set we bound the probability by ϵ (for simplicity, we ignore the multiplicative term less than 1). □

Corollary 13. *Let k and n be parameters with $n > k$ and let q be a prime such that $q \geq 2^{4(n-k)}$. Let $\epsilon \geq 11q^{-1/(4(n-k+1))}$ be a parameter. Then for all distributions $\mathbf{e} \in \mathbb{F}_q^n$ such that*

$$\mathrm{H}_\infty(\mathbf{e}) \geq \log\left(\frac{320eq^{n-k}}{\epsilon^5}\right),$$

it holds that (for any $\mathbf{b}$ *and* $\mathbf{g}$ *above)*

$$\Pr_{\mathbf{b},\mathbf{g},\mathbf{e}}[\langle \mathbf{b}, \mathbf{e}\rangle = \mathbf{g}] \le 2\epsilon + k/q^{n-k}$$

and thus $\mathbf{e}$ *is* $(k, 2\epsilon + k/q^{n-k}) - \mathsf{MIPURS}$*.*

The additional k/q^{n-k} term is due to the probability that $\mathbf{A}$ may not be full rank, all of the above analysis was conditioned on $\mathbf{A}$ being full rank. The corollary then follows by replacing $\kappa = n - k$.

4 Fuzzy Extractors

Our motivating application is a new fuzzy extractor that performs error correction "in the exponent." A fuzzy extractor is a pair of algorithms designed to extract stable keys from a physical randomness source that has entropy but is noisy. If repeated readings are taken from the source one expects these readings to be close in an appropriate distance metric but not identical. Before introducing the construction we review the definition. We consider a generic group version of security (computational security is defined in [FMR13], information-theoretic security in [DORS08]).

To model the fact that the adversary receives values in the generic group, when the adversary would be provided a group element, they instead receive the output of a truly random function σ whose domain is the cyclic group $\mathbb{G}_q$; we assume throughout that the range of σ is large enough that the probability of a collision is statistically insignificant (that is $\ll 1/q$). Notationally, we use $\sigma(z)$ to denote the values that the adversary receives. When $z \overset{def}{=} z_1, \ldots, z_n$ then $\sigma(z)$ only passes z_i through σ if $z_i \in \mathbb{G}_q$. For example, $z = (r, \mathbf{A}, r^{\mathbf{Ax}+\mathbf{w}})$, then $\sigma(z) = (\sigma(r), \mathbf{A}, \sigma(r^{\mathbf{Ax}+\mathbf{w}}))$.

Definition 5. *Let* $\mathcal{E}$ *be a family of probability distributions over the metric space* $(\mathcal{M}, \mathsf{dis})$*. A pair of procedures* $(\mathsf{Gen} : \mathcal{M} \to \{0,1\}^\kappa \times \{0,1\}^*, \mathsf{Rep} : \mathcal{M} \times \{0,1\}^* \to \{0,1\}^\kappa)$ *is an* $(\mathcal{M}, \mathcal{E}, \kappa, t)$-fuzzy extractor *that is* (ϵ_{sec}, m)*-hard with error* δ *if* Gen *and* Rep *satisfy the following properties:*

- Correctness*: if* $\mathsf{dis}(\psi, \psi') \le t$ *and* $(\mathsf{key}, \mathsf{pub}) \leftarrow \mathsf{Gen}(\psi)$*, then* $\Pr[\mathsf{Rep}(\psi', \mathsf{pub}) = \mathsf{key}] \ge 1 - \delta$*.*
- Security*: for any distribution* $\mathbf{e} \in \mathcal{E}$*, the string* key *is close to random conditioned on* pub *for all* $\mathcal{A}$ *making at most* m *queries to the group oracle* $\mathcal{O}_{\mathbb{G}}$*, that is*

$$\left| \Pr_{\substack{\sigma \overset{\$}{\leftarrow} \Sigma, \\ (\mathsf{Key},\mathsf{Pub}) \leftarrow \mathsf{Gen}(\mathbf{e})}} [\mathcal{A}^{\mathcal{O}_{\mathbb{G}}}(\sigma(\mathsf{Key}, \mathsf{Pub})) = 1] - \Pr[\mathcal{A}^{\mathcal{O}_{\mathbb{G}}}(\sigma(U, \mathsf{Pub})) = 1] \right| \le \epsilon_{sec}.$$

We also assume that the adversary receives $\sigma(1)$. The errors are chosen before Pub: if the error pattern between ψ and ψ' depends on the output of Gen, then there is no guarantee about the probability of correctness.

One can directly build a fuzzy extractor out of any $\mathbf{e}$ that satisfies the MIPURS condition. One instantiates the code-offset "in the exponent" and then uses hardcore elements of $\mathbf{x}$ as the key.

Construction 1. *Let* γ *be a security parameter,* t *be a distance,* $k = \omega(\log \gamma)$*,* $\alpha \in \mathbb{Z}^+$*,* $\ell \in \mathbb{Z}^+$*, let* q *be a prime and let* $\mathbb{G}_q$ *be a cyclic group of order* q*. Let* $\mathbb{F}_q$ *be the field with* q *elements. Suppose that* $\mathbf{e} \in \mathbb{F}_q^n$*, and let* dis *be the Hamming metric. Define* (Gen, Rep) *as follows:*

```
Gen $(\mathbf{e} = \mathbf{e}_1, ..., \mathbf{e}_n)$
  1. Sample generator $r$ of $\mathbb{G}_q$.
  2. Sample $\mathbf{A} \leftarrow (\mathbb{F}_q)^{n \times (k+\alpha)}$,
     $\mathbf{x} \leftarrow (\mathbb{F}_q)^{k+\alpha}$.
  3. For $i = 1, ..., n$: set $\mathsf{c}_i = r^{\mathbf{A}_i \cdot \mathbf{x} + \mathbf{e}_i}$.
  4. Set $\mathsf{key} = r^{\mathbf{x}_0}, ..., r^{\mathbf{x}_{\alpha-1}}$.
  5. Set $\mathsf{pub} = (r, \mathbf{A}, \{\mathsf{c}_i\}_{i=1}^n)$.
  6. Output $(\mathsf{key}, \mathsf{pub})$.
```

```
Rep $(\mathbf{e}', \mathsf{pub} = (r, \mathbf{A}, \mathsf{c}_1 \ldots \mathsf{c}_n))$
  1. For $i = 1, ..., n$, set $\mathsf{c}_i = \mathsf{c}_i / r^{\mathbf{e}'_i}$.
  2. For $i = 1, ..., \ell$:
       (i) Sample $J_i \subseteq \{1, ..., n\}$
           where $|J| = k$.
      (ii) If $\mathbf{A}_{J_i}^{-1}$ does not exist go to 2.
     (iii) Compute $\mathbf{s} = r^{\mathbf{A}_{J_i}^{-1} \mathbf{c}_{J_i}}$.
      (iv) Compute $\mathsf{c}' = r^{\mathbf{A}(\mathbf{A}_{J_i}^{-1} \mathbf{c}_{J_i})}$.
       (v) If $\mathsf{dis}(\mathsf{c}, \mathsf{c}') \leq t$, output $\mathbf{s}_0, ..., \mathbf{s}_\alpha$.
  3. Output $\perp$.
```

Theorem 14. *Let all parameters be as in Construction 1. Let $\mathbf{e}^1, ..., \mathbf{e}^\rho \in \mathbb{F}_q^n$ be (k, β)-*MIPURS *distributions. Then* (Gen, Rep) *is a fuzzy extractor that is (ϵ_{sec}, m)-hard for all adversaries in the generic group model (making at most m queries) where*

$$\epsilon_{sec} = \left(\frac{((m+n+2)(m+n+1))^2}{2} \right) \left(\frac{3}{q} + \beta \right).$$

Proof. To show this we combine Theorem 2 with a lemma that generalizes Akavia, Goldwasser, and Vaikuntanathan's result on hardcore elements of LWE [AGV09, Lemma 2]. Their result is that if the decision version of LWE is hard for k dimensions than any additional α dimensions are hardcore. The core idea of the proof is that if one distinguish these "hardcore" dimensions then an outer adversary could augment their LWE instance by just sampling these α new coordinates of $\mathbf{x}$ and extending $\mathbf{A}$ accordingly. Note that this can all be done linearly. We restate this lemma for the generic group setting here (the proof is identical to that of Akavia, Goldwasser, and Vaikuntanathan):

Lemma 15. *For any integer $n > 0$, prime $q \geq 2$, and let $\mathbb{G}_q$ be a group of order q, error-distribution $\mathbf{e}$ over $\mathbb{Z}_q^n$, if for random $\mathbf{A} \in \mathbb{F}_q^{n \times k}, \mathbf{x} \in \mathbb{F}_q^k, \mathbf{U} \in \mathbb{F}^n$ uniformly distributed one has for all PPT $\mathcal{A}$:*

$$\left| \Pr_{\sigma \xleftarrow{\$} \Sigma} [\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\mathbf{A}, \sigma(\mathbf{A}\mathbf{x} + \mathbf{e})) = 1] - \Pr_{\sigma \xleftarrow{\$} \Sigma} [\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\mathbf{A}, \sigma(\mathbf{U})) = 1] \right| < \epsilon.$$

Then for $\mathbf{A}' \in \mathbb{F}_q^{n \times (k+\alpha)}, \mathbf{x} \in \mathbb{F}_q^{k+\alpha}, \mathbf{V} \in \mathbb{F}_q^\alpha$ uniformly distributed one has that

$$\left| \Pr_{\sigma \xleftarrow{\$} \Sigma} [\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\mathbf{x}_{0 \ldots \alpha-1}, \mathbf{A}', \sigma(\mathbf{A}'\mathbf{x} + \mathbf{e})) = 1] - \Pr_{\sigma \xleftarrow{\$} \Sigma} [\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\mathbf{V}, \mathbf{A}', \sigma(\mathbf{A}'\mathbf{x} + \mathbf{e})) = 1] \right| < \epsilon.$$

This immediately implies that

$$\left| \Pr_{\sigma \xleftarrow{\$} \Sigma} [\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\sigma(\mathbf{x}_{0 \ldots \alpha-1}), \mathbf{A}', \sigma(\mathbf{A}'\mathbf{x} + \mathbf{e})) = 1] - \Pr_{\sigma \xleftarrow{\$} \Sigma} [\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\sigma(\mathbf{V}), \mathbf{A}', \sigma(\mathbf{A}'\mathbf{x} + \mathbf{e})) = 1] \right| < \epsilon.$$

We also note that to apply this Lemma, the distribution $\mathbf{e}$ must be $(k, \beta) - \mathsf{MIPURS}$ while $\mathbf{x}$ is of length $k + \alpha$. □

We defer arguing correctness to the more difficult case when $\mathbf{e}$ is a binary value that must be "amplified" to a MIPURS distribution by component-wise multiplication with a random vector. We consider this case in the next subsection.

4.1 Handling binary e

As shown in Lemma 4 when the input value $\mathbf{e}$ is binary and subsets of $\mathbf{e}$ are hard to predict one can form a MIPURS distribution by multiplying by an auxiliary random and uniform random variable $\mathbf{r} \in \mathbb{F}_q^n$. This has the effect of placing random errors in the locations where $\mathbf{e}_i = 1$. Since decoding finds a subset without errors (it does not rely on the magnitude of errors) we can augment errors into random errors.

However, this creates a problem with decoding. When bits of $\mathbf{e}$ are 1, denoted $\mathbf{e}_j = 1$ we cannot use location j for decoding as it is a random value (even if $\mathbf{e}'_j = 1$ as well). When one amplifies a binary $\mathbf{e}$, we recommending using another uniform random variable $\mathbf{y} \in \{0,1\}^n$ and check when $\mathbf{y}_i \neq \mathbf{e}_i$ to indicate when to include a random error. Then in reproduction the algorithm should restrict to locations where $\mathbf{y}_i = \mathbf{e}_i$. Using Chernoff bounds one can show this subset is big enough and the error rate in this subset is not much higher than the overall error rate (except with negligible probability). If $k + \alpha$ is just barely $\omega(\log n)$ one can support error rates that are just barely $o(n)$. These arguments are more complex than the fuzzy extractor presented in Construction 1 so we show efficiency of only this construction.

Construction 2. *Let γ be a security parameter, t be a distance, $k = \omega(\log \gamma)$, $\alpha \in \mathbb{Z}^+$, q be a prime and let $\mathbb{G}_q$ be some cycle group of order q. Let $\mathbb{F}_q$ be the field with q elements. Let $\mathcal{E} \in \{0,1\}^n$ and let* dis *be the Hamming metric. Let $\tau = \max(0.01, t/n)$. Define* (Gen, Rep) *as follows:*

```
Gen($\mathbf{e} = \mathbf{e}_1, ..., \mathbf{e}_n$)

  1. Sample random generator $r$ of $\mathbb{G}_q$.
  2. Sample $\mathbf{A} \leftarrow (\mathbb{F}_q)^{n \times (k+\alpha)}, \mathbf{x} \leftarrow (\mathbb{F}_q)^{k+\alpha}$.
  3. Sample $\mathbf{y} \xleftarrow{\$} \{0,1\}^n$.
  4. For $i = 1, ..., n$:
       (i) If $\mathbf{e}_i = \mathbf{y}_i$, set $\mathsf{c}_i = r^{\mathbf{A}_i \cdot x}$.
      (ii) Else set $\mathsf{c}_i \xleftarrow{\$} \mathbb{G}_q$.
  5. Set $\mathsf{key} = r^{\mathbf{x}_{0...\alpha-1}}$.
  6. Set $\mathsf{pub} = (r, \mathbf{y}, \mathbf{A}, \{\mathsf{c}_i\}_{i=1}^n)$.
  7. Output $(\mathsf{key}, \mathsf{pub})$.
```

```
Rep($\mathbf{e}', \mathsf{pub} = (r, \mathbf{y}, \mathbf{A}, \mathsf{c}_1 \ldots \mathsf{c}_\ell)$)

  1. Let $\mathcal{I} = \{i | \mathbf{e}'_i = \mathbf{y}_i\}$.
  2. For $i = 1, ..., \ell$:
       (i) Choose random $J_i \subseteq \mathcal{I}$ where $|J| = k$.
      (ii) If $\mathbf{A}_{J_i}^{-1}$ does not exist, output $\perp$.
     (iii) Compute $\mathsf{c}' = r^{\mathbf{A}(\mathbf{A}_{J_i}^{-1} \mathbf{c}_{J_i})}$.
      (iv) If $\mathsf{dis}(\mathsf{c}_{\mathcal{I}}, \mathsf{c}'_{\mathcal{I}}) \leq |\mathsf{c}_{\mathcal{I}}|(1 - 2\tau)$,
           output $r_{0...\alpha-1}^{\mathbf{A}_{J_i}^{-1} \mathbf{c}_{J_i}}$.
  3. Output $\perp$.
```

We now show this construction is correct and efficient. Our correctness argument considers $k' \overset{def}{=} k + \alpha = \Theta(n)$ and $t = \Theta(n)$. For the fuzzy extractor application, one would consider a smaller k' and t. In particular, for $t = o(n)$ the theorem applies with overwhelming probability as long as $k' \leq \frac{(1-\Theta(1))}{3} \cdot n$. We use the q-ary entropy function which is a generalization of the binary entropy function to larger alphabets. $H_q(x)$ is the q-ary entropy function defined as

$$H_q(x) = x\log_q(q-1) - x\log_q(x) - (1-x)\log_q(1-x).$$

Theorem 16. *Let parameters be as in Construction 2. Define $\tau = t/n$. Let $0 < \delta < 1 - H_q(4\tau)$ and suppose that $k' \leq (1/3)\cdot\lceil 1 - H_q(4\tau) - \delta\rceil n$. If* Rep *outputs a value other than $\perp$ it is correct with probability at least $1 - e^{-\Theta(n)}$.*

Proof of Theorem 16. We assume a fixed number of iterations in Rep denoted by ℓ. For any two ψ, ψ' used as inputs to Gen and Rep respectively, we assume that $\mathsf{dis}(\psi, \psi') \leq t$ and that the value $\mathbf{y}$ is independent of both values (by Def 5, any distribution over ψ' does not depend on the public value). We first show conditions for the final check of $\mathsf{dis}(\mathsf{c}_{\mathcal{I}}, \mathsf{c}'_{\mathcal{I}}) \leq |\mathsf{c}_{\mathcal{I}}|(1-2\tau)$ to return correctly if and only if $r^{\mathbf{x}} = r^{\mathbf{A}_{J_i}^{-1}\mathsf{c}_{J_i}}$. We stress that this is property is independent of the chosen subset and only depends on $\mathbf{A}, \mathbf{x}, \psi, \psi'$ and $\mathbf{y}$. We refer to the values in the exponent, but our argument directly applies to the generated group elements. Define the matrix $\mathbf{A}_{\mathcal{I}}$ defined by the set $\mathcal{I}$. By Chernoff bound,

$$\Pr\left[|\mathcal{I}| \leq \left(1 - \frac{1}{3}\right)\mathbb{E}\,|\mathcal{I}|\right] = \Pr\left[|\mathcal{I}| \leq \left(\frac{2}{3}\right)\frac{n}{2}\right] \leq e^{-\frac{n}{36}} \leq e^{-\Theta(n)}.$$

Without loss of generality we assume that the size of $\mathcal{I} = n/3$. Consider some fixed ψ, ψ' such that $\mathsf{dis}(\psi, \psi') \leq t$ and define the random variable Z of length n where a bit i of z that indicates when $\psi_i = \psi'_i$ and when $\psi'_i = y_i$. We consider the setting when $t = \Theta(n)$, if $t = o(n)$ then $\tau \overset{def}{=} t/n \leq .01$ and the condition holds with high probability. Define $S = \{i \mid \psi_i = y_i = \psi'_i\}$. We can lower bound of size of S by a binomial distribution with $n/3$ flips and probability $p \geq 1 - \tau$. That is, $\mathbb{E}[S] \geq (n/3)(1-\tau)$. By an additive Chernoff bound,

$$\Pr\left[S - \mathbb{E}[S] \geq \tau n\right] =\leq 2e^{-2\tau^2 n} \leq e^{-\Theta(n)}.$$

To show correctness it remains to show that $\mathbf{x}$ is unique. We again assume that $\mathcal{I} = n/3$, all arguments proceed similarly when $\mathcal{I} > n/3$. To show uniqueness of $\mathbf{x}$ suppose that there exists two $\mathbf{x}_1, \mathbf{x}_2$ such that $\mathsf{dis}(\mathbf{A}_{\mathcal{I}}\mathbf{x}_1, \mathsf{c}_{\mathcal{I}}) \leq |\mathsf{c}_{\mathcal{I}}|(1-2\tau)$ and $\mathsf{dis}(\mathbf{A}_{\mathcal{I}}\mathbf{x}_2, \mathsf{c}_{\mathcal{I}}) \leq |\mathsf{c}_{\mathcal{I}}|(1-2\tau)$. This means that $\mathbf{A}_{\mathcal{I}}(\mathbf{x}_1 - \mathbf{x}_2)$ contains at most $4t/3$ nonzero components. To complete the proof we use the following standard theorem:

Lemma 17. *[Gur10, Theorem 8] For prime $q, \delta \in [0, 1 - 1/q), 0 < \epsilon < 1 - H_q(\delta)$ and sufficiently large n, the following holds for $k' = \lceil(1 - H_q(\delta) - \epsilon)n\rceil$. If $\mathbf{A} \in \mathbb{Z}_q^{n\times k'}$ is drawn uniformly at random, then the linear code with $\mathbf{A}$ as a generator matrix has rate at least $(1 - H_q(\delta) - \epsilon)$ and relative distance at least δ with probability at least $1 - e^{-\Omega(n)}$.*

Application of Lemma 17 completes the proof of Theorem 16. □

Recovery. Our analysis of running time is similar in spirit to that of Canetti et al. [CFP+16]. For any given i, the probability that $\psi'_{J_i} = \psi_{J_i}$ is at least

$$\left(1 - \frac{2t}{n - 3k'}\right)^{k'}.$$

This follows since $d(\psi_{\mathcal{I}}, \psi'_{\mathcal{I}}) \leq 2\tau\cdot|\mathcal{I}|$ and since we are sampling sets without replacement the number of error less positions remains at least $n/3 - k'$. We bound the probability of an error for each sample (without replacement) by the probability of the last sample which is at most $\frac{2t/3}{n/3-k'} = \frac{2t}{n-3k'}$. The probability that no iteration matches is at most

$$\left(1 - \left(1 - \frac{2t}{n-3k'}\right)^{k'}\right)^{\ell}.$$

We can use the approximation $\exp(x) \approx 1 + x$ to get

$$\left(1 - \left(1 - \frac{2t}{n-3k'}\right)^{k'}\right)^{\ell} \approx \left(1 - \exp\left(-\frac{2tk'}{n-3k'}\right)\right)^{\ell} \approx \exp\left(-\ell \cdot \exp\left(-\frac{2tk'}{n-3k'}\right)\right).$$

Suppose that correctness $1 - \delta \geq 1 - (\delta' + \exp(-\Theta(n)))$ is desired. (Here, the $\exp(-\Theta(n))$ term is due to sampling of a bad matrix $\mathbf{A}$ and failures of Chernoff bounds above.) Then if $k' = o(n)$ with $tk' = cn \ln n$ for some constant c, setting $\ell \approx n^{2c+\Theta(1)} \log \frac{1}{\delta'}$ suffices as:

$$\begin{aligned}
\exp\left(-\ell \cdot \exp\left(-\frac{tk}{n-3k}\right)\right) &= \exp\left(-n^{2c} \log \frac{1}{\delta'} \exp\left(-\frac{2tk'}{n-3k'}\right)\right) \\
&\leq \exp\left(-n^{2c+\Theta(1)} \cdot \log \frac{1}{\delta'} \cdot \exp\left(-(2c + o(1)) \ln n\right)\right) \\
&= \exp\left(-n^{2c+\Theta(1)} \cdot \log \frac{1}{\delta'} \cdot n^{-(2c+o(1))}\right) \\
&\leq \delta'.
\end{aligned}$$

Thus, for $k = \omega(\ln n)$, one can support error rates $t = o(n)$.

Comparison with sample-then-lock. As mentioned in the introduction, Canetti et al. [CFP$^+$16] proposed a reusable fuzzy extractor based on digital lockers called *sample-then-lock*. Intuitively, a digital locker is a symmetric encryption that is semantically secure even when instantiated with keys that are correlated and only have entropy [CKVW10]. At a high level, their construction took multiple samples $w_{\mathcal{I}_j}$ from the input biometric and use these as keys for different digital lockers, all of which contained the same key. Our construction improves on the storage and use of confidence information over Canetti et al. (see the Introduction). On the other hand the fact that all subsets are available to an adversary does provide them with additional power. As mentioned in the Introduction, our definition can handle a small number of subsets with insufficient entropy, as long as they are unlikely to be in the null space of the code. Canetti et al. were able to show security for all distributions where sampling produced entropy:

Definition 6 ([CFP$^+$16] Sources with High Entropy Samples)**.** *Let the source* $\mathbf{e} = \mathbf{e}_1, \ldots, \mathbf{e}_n$ *consist of strings of length* n *over some arbitrary alphabet* $\mathcal{Z}$*. We say that the source* $\mathbf{e}$ *is a source with a* (k, β)***-entropy-samples*** *if*

$$\mathop{\mathbb{E}}_{j_1,\ldots,j_k \xleftarrow{\$} [1,\ldots,n]} \left(\max_z \{\Pr[(\mathbf{e}_{j_1}, \ldots, \mathbf{e}_{j_k}) = z \mid j_1, \ldots, j_k]\}\right) \leq \beta.$$

Our construction requires all subsets to have high entropy (Definition 2) instead of an average subset.

4.2 Reusability

Reusability is the ability to support multiple independent enrollments of the same value, allowing users to reuse the same biometric or PUF, for example, with multiple noncooperating providers. More precisely, the algorithm Gen may be run multiple times on correlated readings $\mathbf{e}^1, ..., \mathbf{e}^\rho$ of a given source. Each time, Gen will produce a different pair of values $(\mathsf{key}^1, \mathsf{pub}^1), ..., (\mathsf{key}^\rho, \mathsf{pub}^\rho)$. Security for each extracted string key^i should hold even in the presence of all the helper strings $\mathsf{pub}^1, \ldots, \mathsf{pub}^\rho$ (the reproduction procedure Rep at the ith provider still obtains only a single $\mathbf{e}'$ close to $\mathbf{e}^i$ and uses a single helper string pub_i). Because providers may not trust each other key_i should be secure even when all key_j for $j \neq i$ are also given to the adversary.

Definition 7 (Reusable Fuzzy Extractor [CFP+16]). *Let $\mathcal{E}$ be a family of distributions over $\mathcal{M}$. Let $(\mathsf{Gen}, \mathsf{Rep})$ be a $(\mathcal{M}, \mathcal{E}, \kappa, t)$-computational fuzzy extractor that is (ϵ_{sec}, m)-hard with error δ. Let $(\mathbf{e}^1, \mathbf{e}^2, \ldots, \mathbf{e}^\rho)$ be ρ correlated random variables such that each $\mathbf{e}^j \in \mathcal{E}$. Let $\mathcal{A}$ be an adversary. Define the following game for all $j = 1, ..., \rho$:*

- ***Sampling** The challenger samples $u \leftarrow \mathbb{G}_q^\alpha$ and $\sigma \xleftarrow{\$} \Sigma$.*
- ***Generation** For all $1 \leq i \leq \rho$, the challenger computes $(\mathsf{key}^i, \mathsf{pub}^i) \leftarrow \mathsf{Gen}(\mathbf{e}^j)$.*
- ***Distinguishing** The advantage of $\mathcal{A}$ is*

$$\begin{aligned} Adv(\mathcal{A}) \overset{def}{=} &\Pr[\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\sigma(\mathsf{key}^1, ..., \mathsf{key}^{j-1}, \mathsf{key}^j, \mathsf{key}^{j+1}, ..., \mathsf{key}^\rho, \mathsf{pub}^1, ..., \mathsf{pub}^\rho)) = 1] \\ &- \Pr[\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\sigma(\mathsf{key}^1, ..., \mathsf{key}^{j-1}, u, \mathsf{key}^{j+1}, ..., \mathsf{key}^\rho, \mathsf{pub}^1, ..., \mathsf{pub}^\rho)) = 1]. \end{aligned}$$

$(\mathsf{Gen}, \mathsf{Rep})$ is $(\rho, \epsilon_{sec}, m)$-reusable if for all $\mathcal{A}$ making at most m queries to $\mathcal{O}_\mathbb{G}$ and all $j = 1, ..., \rho$, the advantage is at most ϵ_{sec}.

Claim 18. *Let all parameters be as in Construction 1. Let $\mathbf{e}^1, ..., \mathbf{e}^\rho \in \mathbb{F}_q^n$ be (k, β)-MIPURS distributions. Then $(\mathsf{Gen}, \mathsf{Rep})$ is a $(\rho, \epsilon_{sec}, m)$-reusable fuzzy extractor for all adversaries in the generic group model making at most m queries where*

$$\epsilon_{sec} = (2\rho + 1)\left(\frac{((m+n+2)(m+n+1))^2}{2}\right)\left(\frac{2}{q} + \beta\right).$$

We do not formally prove Claim 18 because it requires the using the proof of Theorem 2 (rather than just the statement of the theorem). Here, we present the conceptual argument. Without loss of generality, we assume that the adversary is trying to learn information about the first key. Since we sample generators $r^1, ..., r^\rho$ we can define $s^2, ..., s^\rho$ where s^i is the discrete logarithm between r^1 and r^i (define $s^1 = 1$). For the construction to be reusable for all distinguishers, it must be true that for uniform $\mathbf{V} \in \mathbb{F}_q^\alpha$:

$$\begin{aligned} &\left| \Pr_{\sigma \xleftarrow{\$} \Sigma} [\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\sigma(s_1(\mathbf{x}^1_{0\ldots\alpha-1})), \mathbf{A}^1, \sigma(s_1(\mathbf{A}^1\mathbf{x}^1 + \mathbf{e}^1)), \{\sigma(\mathsf{key}^i, \mathsf{pub}^i)\}_{i=2}^\rho)) = 1] \right. \\ &\left. - \Pr_{\sigma \xleftarrow{\$} \Sigma} [\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\sigma(\mathbf{V}), \mathbf{A}^1, \sigma(s_1(\mathbf{A}^1\mathbf{x}^1 + \mathbf{e}^1)), \{\sigma(\mathsf{key}^i, \mathsf{pub}^i)\}_{i=2}^\rho)) = 1] \right| \leq \epsilon_{sec}. \end{aligned}$$

where $\mathbf{x}^1, \mathbf{A}^1$ are values sampled in the invocation of $\mathsf{Gen}(\mathbf{e}^1)$.

Conditioned on $\sigma(\mathsf{pub}^1, \mathsf{pub}^i)$, the values $\mathbf{x}^1$ and key^i are independent. Then values received in $\mathsf{key}^i, \mathsf{pub}^i$ are multiplied by s^i. As shown in Theorem 2, $\mathcal{A}$ has negligible probability of seeing a 0 response for any nontrivial linear combination. Thus as long as the range of Σ is large enough for collisions between independent draws of Σ to be observed by the adversary, we can independently initialize oracles $\sigma^1, ..., \sigma^\rho$. Each change from a coupled oracle involving elements of $(\mathsf{key}^i, \mathsf{pub}^i)$ to a separate oracle σ^i can be distinguished only if $\mathcal{A}$ can find a meaningful linear combination $\alpha_{j,k}$ involving some other oracles and the values of $(\mathsf{key}^i, \mathsf{pub}^i)$. That is finding some $\sum_{j=1}\sum_k \alpha_{j,k} s^j \mathsf{pub}_k^j = \sum_k \alpha_k s^i \mathsf{pub}_k^i$. Since the value s^i starts as uniformly random from the adversary's perspective this occurs with probability (see Lemma 24) at most

$$\frac{((m+n+2)(m+n+1))^2}{4q}.$$

Thus, we can replace these coupled oracles in a hybrid fashion paying cost at most

$$\rho\frac{((m+n+2)(m+n+1))^2}{4q}.$$

To show hardness in this separate oracle setting we follow a three step process:

1. First we replace all handles for pub^i with uniform values. This again requires using the proof of Theorem 2 which argues that with high probability no response from $\mathcal{O}_\mathcal{G}$ is useful and thus even when correlated values are in independent oracle, the adversary never learns anything useful from any individual oracle. For this step, we need a lemma similar to Lemma 15 that is proved analogously:

   Lemma 19. *For any integer $n > 0$, prime $q \geq 2$, and let $\mathbb{G}_q$ be a group of order q, error-distribution $\mathbf{e}$ over $\mathbb{F}_q^n$, if for random $\mathbf{A} \in \mathbb{F}_q^{n\times k}, \mathbf{x} \in \mathbb{F}_q^k, \mathbf{U} \in \mathbb{F}^n$ uniformly distributed one has for all PPT $\mathcal{A}$ that*

   $$\left|\Pr_{\sigma \xleftarrow{\$} \Sigma}[\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\mathbf{A}, \sigma(\mathbf{A}\mathbf{x}+\mathbf{e})) = 1] - \Pr_{\sigma \xleftarrow{\$} \Sigma}[\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\mathbf{A}, \sigma(\mathbf{U})) = 1]\right| < \epsilon,$$

   then for $\mathbf{A}' \in \mathbb{F}_q^{n\times(k+\alpha)}, \mathbf{x} \in \mathbb{F}_q^{k+\alpha}, \mathbf{U} \in \mathbb{F}_q^n$ uniformly distributed one has that

   $$\left|\Pr_{\sigma \xleftarrow{\$} \Sigma}[\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\mathbf{x}_{0\dots\alpha-1}, \mathbf{A}', \sigma(\mathbf{A}'\mathbf{x}+\mathbf{e})) = 1] - \Pr_{\sigma \xleftarrow{\$} \Sigma}[\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\mathbf{x}_{0\dots\alpha-1}, \mathbf{A}', \sigma(\mathbf{U})) = 1]\right| < \epsilon.$$

   This immediately implies that

   $$\left|\Pr_{\sigma \xleftarrow{\$} \Sigma}[\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\sigma(\mathbf{x}_{0\dots\alpha-1}), \mathbf{A}', \sigma(\mathbf{A}'\mathbf{x}+\mathbf{e})) = 1] - \Pr_{\sigma \xleftarrow{\$} \Sigma}[\mathcal{A}^{\mathcal{O}_\mathbb{G}}(\sigma(\mathbf{x}_{0\dots\alpha-1}), \mathbf{A}', \sigma(\mathbf{U})) = 1]\right| < \epsilon.$$

2. We then replace the key for $\sigma(x^1_{0,\dots,\alpha-1})$ with a uniform value (using Theorem 2 and Lemma 15).

3. Repeating step 1 now that the relevant key has been replaced with uniform.

5 Pattern Matching Obfuscation

In this section we introduce a second application for our main theorem. This application is known as pattern matching obfuscation. The goal is to obfuscate a string v of length n which consists of $(0, 1, \perp)$ where $\perp$ is a wildcard. The obfuscated program on input $x \in \{0,1\}^n$ should output 1 if and only if $\forall i, x_i = v_i \vee v_i = \perp$. Roughly, the wildcard positions are matched automatically. We directly use definitions and the construction from the recent work of Bishop et al. [BKM+18]. Our improvement is in analysis, showing security for more distributions V. We start by introducing a definition of security:

Definition 8. *Let $\mathcal{C}_n$ be a family of circuits that take inputs of length n and let $\mathcal{O}$ be a PPT algorithm taking $n \in \mathbb{N}$ and $C \in \mathcal{C}_n$ outputting a new circuit C'. Let $\mathcal{D}_n$ be an ensemble of distribution families where each $D \in \mathcal{D}_n$ is a distribution over circuits in $\mathcal{C}_n$. $\mathcal{O}$ is a distributional VBB obfuscator for $\mathcal{D}_n$ over $\mathcal{C}_n$ if:*

1. Functionality*: For each $n, C \in \mathcal{C}_n$ and $x \in \{0,1\}^n$,* $\Pr_{\mathcal{O},C'}[C'(x) = C(x)] \geq 1 - \mathtt{ngl}(n)$.
2. Slowdown*: For each $n, C \in \mathcal{C}_n$, the resulting C' can be evaluated in time* $\mathtt{poly}(|C|, n)$.
3. Security*: For each generic adversary $\mathcal{A}$ making at most m queries, there is a polynomial time simulator $\mathcal{S}$ such that $\forall n \in \mathbb{N}$, and each $D \in \mathcal{D}_n$ and each predicate P*

$$\left| \Pr_{\substack{C \leftarrow \mathcal{D}_n,\\ \mathcal{O}^{\mathcal{G}}, \mathcal{A}}} [\mathcal{A}^{\mathcal{G}}(\mathcal{O}^{\mathcal{G}}(C, 1^n)) = P(C)] - \Pr_{C \leftarrow \mathcal{D}_n, \mathcal{S}} [S^C(1^{|C|}, 1^n) = P(C)] \right| \leq \mathtt{ngl}(n).$$

Construction 3. *We now reiterate the construction from Bishop et al. adapted to use a random linear code for some prime $q = q(n)$.*

```
$\mathcal{O}(\mathbf{v} \in \{0,1,\perp\}^n, q, g)$:
where $g$ is a generator of a group $\mathbb{G}_q$.
  1. Sample $\mathbf{A} \in (\mathbb{F}_q)^{2n \times n}$,
     $\mathbf{x}_0 = 0, \mathbf{x}_{1,\ldots,n-1} \leftarrow (\mathbb{F}_q)^{n-1}$.
  2. Sample $\mathbf{e} \in \mathbb{Z}_q^{2n}$ uniformly.
  3. For $i = 0$ to $n - 1$:
      (a) If $v_i = 1$ set $e_{2i} = 0$.
      (b) If $v_i = 0$ set $e_{2i+1} = 0$.
      (c) If $v_i = \perp$ set $e_{2i} = 0, e_{2i+1} = 0$.
  4. Compute $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$.
  5. Output $g^{\mathbf{y}}, \mathbf{A}$.
```

```
Eval($g^{\mathbf{y}}, \mathbf{A}, \psi \in \{0,1\}^n$):
  1. Define $\mathcal{I}$ as
     $\{i \in [1...2n] \mid \psi_{\lfloor i/2 \rfloor} = (i \mod 2)\}$.
  2. Compute $\mathbf{A}_{\mathcal{I}}^{-1}$.
     If none exists output $\perp$.
  3. Output $g^{\mathbf{A}_{\mathcal{I},1}^{-1} \cdot \mathbf{y}} \stackrel{?}{=} g$.
```

To state our security theorem we need to consider the transform from strings v over $\{0, 1, \perp\}$ to binary strings.

$$\mathsf{Bin}(\mathbf{v}) = \mathbf{s} \text{ where } \begin{cases} s_i = 10 & \text{if } v_i = 1, \\ s_i = 01 & \text{if } v_i = 0, \\ s_i = 00 & \text{if } v_i = \perp. \end{cases}$$

Lastly, define the distribution $\mathbf{e}' = \mathbf{r} \cdot_c \mathsf{Bin}(\mathbf{v})_i$ for uniform distributed $\mathbf{r} \in \mathbb{F}_q^{2n}$.

Theorem 20. *Let $\ell \in \mathbb{Z}^+$ be a free parameter. Define $\mathcal{V}$ as the set of all distributions V such that $E' = U_{\mathbb{F}_q}^n \cdot_c \mathsf{Bin}(V)$ is a distribution that is $(n, \beta) - \mathsf{MIPURS}$. Then Construction 3 is VBB secure for generic $\mathcal{D}$ making at most m queries with distinguishing probability at most*

$$\frac{((m+n+2)(m+n+1))^2}{2} \left(\frac{3}{q} + \beta\right).$$

Proof. Like the work of Bishop et al. [BKM+18, Theorem 16] the VBB security of the theorem follows by noting for any adversary $\mathcal{A}$ there exists a simulator S that initializes $\mathcal{A}$, provides them with $2n$ random handles (and simulates the interaction with $\mathcal{O}_r$) and outputs their output. By Theorem 2, the output of this simulator differs from the adversary in the real game by at most the above probability. □

6 Hardness of Decoding in the Standard Model

In this section we ask, "for which $\mathbf{e}$ is code offset in the exponent secure assuming only assuming the hardness of discrete log?" We use this as a comparison to the distributions that aresecure in the generic group model. In this section, we consider hardness of decoding random linear codes in the exponent. In Appendix B we consider Reed-Solomon codes. Both results follow a three part outline:

1. A theorem of Brands [Bra93] which says that if an adversary $\mathcal{A}$ given a uniformly distributed $g^{\mathbf{y}}$ can find $\mathbf{z}$ such that $g^{\langle \mathbf{y}, \mathbf{z} \rangle} = 1$ or equivalently that a vector $\mathbf{z}$ such that $\langle \mathbf{y}, \mathbf{z} \rangle = 0$ then one can solve discrete log with the same probability. For a vector of length n and prime q, this problem is known as the $\mathtt{FIND} - \mathtt{REP}(n, q)$ problem.
2. A combinatorial lemma which shows conditions for a random $g^{\mathbf{y}}$ to be within some distance parameter c of a codeword with noticeable probability. That is, $\exists \mathbf{z} \in \mathbb{C}$ such that $\mathsf{dis}(g^{\mathbf{y}}, g^{\mathbf{z}}) \le c$ (for the codeword space $\mathbb{C}$).
3. Let $\mathcal{O}$ be an oracle for bounded distance decoding. That is, given $g^{\mathbf{y}}$, $\mathcal{O}$ returns some $g^{\mathbf{z}}$ where $\mathsf{dis}(g^{\mathbf{z}}, g^{\mathbf{y}}) \le c$ and $\mathbf{z} \in \mathbb{C}$. Recall that linear codes have known null spaces. Thus, if two vectors $g^{\mathbf{z}}$ and $g^{\mathbf{y}}$ match in more positions than the dimension of the code it is possible to compute a vector γ that is only nonzero in positions where $g^{\mathbf{z}_i} = g^{\mathbf{y}_i}$ and $\langle \gamma, \mathbf{x} \rangle = \langle \gamma, \mathbf{y} \rangle = 0$. If $\mathcal{O}$ works on a random point $g^{\mathbf{y}}$ it is possible to compute a vector γ in the null space of $\mathbf{y}$. This serves as an algorithm to solve the $\mathtt{FIND} - \mathtt{REP}$ and completes the connection to hardness of discrete log.

In this section we focus on a combinatorial lemma to establish point 2. In Appendix B, we present a similar result for Reed-Solomon codes improving prior work of Peikert [Pei06].

Notation and Definitions. We will consider noise vectors $\mathbf{e} \in \mathbb{F}_q$ where the Hamming weight of $\mathbf{e}$ denoted $\mathsf{wt}(\mathbf{e}) = t$ and the nonzero entries of $\mathbf{e}$ are uniformly distributed. That is, we consider $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$. Usually in coding theory the goal is *unique decoding.* That is, given some $\mathbf{y}$, if there exists some $\mathbf{z} \in \mathbb{C}$ such that $\mathsf{dis}(\mathbf{y}, \mathbf{z}) \le t$, the algorithm is guaranteed to return $\mathbf{y}$ and $\mathbf{z}$ is uniquely defined. Our results consider algorithms that perform bounded distance decoding. Bounded distance decoding is a relaxation of unique decoding. For a distance c and a point $\mathbf{y} \in \mathbb{Z}_q^n$ a bounded distance decoding algorithm returns some $\mathbf{z} \in \mathbb{C}$ such that $\mathsf{dis}(\mathbf{y}, \mathbf{z}) \le c$. There is no guarantee that $\mathbf{z}$ is unique or is the point in the code closest to $\mathbf{y}$.

Problem $\texttt{BDDE} - \texttt{RL}(n,k,q,c,g)$, or Bounded Distance Decoding (exponent) of Random Linear Codes.

Instance Known generator g of $\mathbb{F}_q$. Define $\mathbf{e}$ as a random vector of weight c in $\mathbb{F}_q$. Define $g^{\mathbf{y}} = g^{\mathbf{Ax}+\mathbf{e}}$ where $\mathbf{A}, \mathbf{x}$ are uniformly distributed. Input is $g^{\mathbf{y}}, \mathbf{A}$.

Output Any codeword $g^{\mathbf{z}}$ where $\exists \mathbf{x} \in \mathbb{Z}_q^k$ such that $\mathbf{z} = \mathbf{Ax}$ and $\mathsf{dis}(\mathbf{x}, \mathbf{z}) \leq c$.

For a code $\mathbf{C}$ we define the distance between a point $\mathbf{y}$ and the code as the minimum distance between $\mathbf{y}$ and any codeword $\mathbf{c}$ in $\mathbf{C}$. Formally, $\mathsf{dis}(\mathbf{y}, \mathbf{C}) = \min_{\mathbf{c} \in \mathbf{C}} \mathsf{dis}(\mathbf{y}, \mathbf{c})$. Consider some point $\mathbf{y}$ in the codespace and a radius r. The *thickness* of a point is the number of Hamming balls (of radius r) inflated around all codewords that cover $\mathbf{y}$. Specifically, define the set of points contained in a Hamming ball of radius r as $\Phi(r, \mathbf{z})$ for each codeword $\mathbf{z}$ in the code $\mathbf{C}$. Then define random variable $\varphi(r, \mathbf{z}, \mathbf{y})$ for each $\Phi(r, \mathbf{z})$ where $\varphi(r, \mathbf{z}, \mathbf{y}) = 1$ if $\mathbf{y} \in \Phi(r, i)$ and 0 otherwise. Then the thickness of $\mathbf{y}$ is

$$\mathsf{Thick}(r, \mathbf{C}, \mathbf{y}) = \sum_{\mathbf{z} \in \mathbf{C}} \varphi(r, \mathbf{z}, \mathbf{y}).$$

Hardness of Decoding. We now present the theorem of this section and our key technical lemma (Lemma 22), then prove the lemma and finally the theorem.

Theorem 21. *For positive integers n, k, c and prime q where $k < n \leq q$ and let g be a generator of $\mathbb{G}_q$. If an efficient algorithm exists to solve $\texttt{BDDE} - \texttt{RL}(n, k, q, n-k-c, g)$ with probability ϵ, then an efficient randomized algorithm exists to solve the discrete log problem in the same group with probability at least*

$$\epsilon' = \epsilon \left(1 - \left(\frac{q^{n-k}}{\mathsf{Vol}(n, n-k-c, q)} + \frac{k}{q^{n-k}}\right)\right).$$

In particular, using a volume bound $\mathsf{Vol}(n, r, q) \geq \binom{n}{k} q^r (1 - n/q)$,

$$\epsilon' = \epsilon \left(1 - \left(\frac{q^c}{\binom{n}{k+c}(1 - \frac{n}{q})} + \frac{k}{q^{n-k}}\right)\right).$$

Lemma 22. *Let a random code be defined by matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times k}$, then*

$$\Pr_{\mathbf{y} \in \mathbb{F}_q^n, \mathbf{A}}[\mathsf{dis}(\mathbf{y}, \mathbf{A}) > n-k-c] \leq \frac{q^{n-k}}{\mathsf{Vol}(n, n-k-c, q)} + \frac{k}{q^{n-k}}.$$

Proof of Lemma 22. $\mathbf{A}$ has q^k codewords in a q^n sized codespace as long as $\mathbf{A}$ is full rank. The probability of $\mathbf{A}$ being full rank is at least $1 - k/q^{n-k}$ [FMR13, Lemma A.3]. The expected thickness of a code or $\mathbb{E}_{\mathbf{y}} \mathsf{Thick}(r, \mathbf{A}, \mathbf{y})$ is the average thickness over all points in the space. Expected thickness is the ratio of the sum of the volume of the balls and the size of the space itself. Note that this value can be greater than 1. A Hamming ball in this space can only be defined up to radius n. We give denote the expected thickness of the code as follows:

$$\mathbb{E}_{\mathbf{y}}(\mathsf{Thick}(r, \mathbf{A}, \mathbf{y})) = \frac{\mathsf{Vol}(n, r, q) \cdot q^k}{q^n} = \mathsf{Vol}(n, r, q) \cdot q^{k-n}$$

For $r = n - k - c$:

$$\mathbb{E}_{\mathbf{y}}(\mathsf{Thick}(n-k-c, \mathbf{A}, \mathbf{y})) \geq \mathsf{Vol}(n, n-k-c, q) \cdot q^{k-n}$$

For a point to have Hamming distance from our code greater than $n-k-c$, its thickness must be 0. For the thickness of a point to be 0, it must deviate from the expected thickness by the expected thickness. We use this fact to bound the probability that a point is distance at least $n-k-c$. We require that each codeword is pairwise independent (that is, $\Pr_{\mathbf{A}}[c \in \mathbf{A} | c' \in \mathbf{A}] = \Pr_{\mathbf{A}}[c \in \mathbf{A}]$). In random linear codes, only generating matrices with dimension 1 are not pairwise independent. We have already restricted our discussion to full rank $\mathbf{A}$. Define an indicator random variable that is 1 when a point c is in the code. The pairwise independence of the code implies pairwise independence of these indicator random variables. With pairwise independent codewords, we use Chebyshev's Inequality to bound the probability of a random point being remote from a random code. We upper bound the variance of Thick by its expectation (since the random variable is nonnegative). In the below equations we only consider $\mathbf{A}$ where $\mathsf{Rank}(\mathbf{A}) = k$ but do not write this to simplify notation. Let $t = n-k-c$, then

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{A}} \Pr_{\mathbf{y}}[\mathsf{dis}(\mathbf{y}, \mathbf{A}) > t] &= \mathop{\mathbb{E}}_{\mathbf{A}} \Pr_{\mathbf{y}}[\mathsf{Thick}(t, \mathbf{A}, \mathbf{y}) = 0] \\
&\leq \mathop{\mathbb{E}}_{\mathbf{A}} \left(\Pr_{\mathbf{y}} \left[|\mathsf{Thick}(t, \mathbf{A}, \mathbf{y}) - \mathbb{E}(\mathsf{Thick}(t, \mathbf{A}, \mathbf{y}))| > \mathbb{E}(\mathsf{Thick}(t, \mathbf{A}, \mathbf{y})) \right] \right) \\
&\leq \mathop{\mathbb{E}}_{\mathbf{A}} \left(\frac{\mathrm{Var}_{\mathbf{y}}(\mathsf{Thick}(t, \mathbf{A}, \mathbf{y}))}{\mathbb{E}_{\mathbf{y}}(\mathsf{Thick}(t, \mathbf{A}, \mathbf{y}))^2} \right) \leq \mathop{\mathbb{E}}_{\mathbf{A}} \left(\frac{1}{\mathbb{E}_{\mathbf{y}}(\mathsf{Thick}(t, \mathbf{A}, \mathbf{y}))} \right) \\
&= \frac{q^{n-k}}{\mathsf{Vol}(n, n-k-c, q)}.
\end{aligned}$$

□

Proof of Theorem 21. Suppose an algorithm $\mathcal{F}$ solves $\mathtt{BDDE}-\mathtt{RL}(n, k, q, n-k-c, g)$ with probability ϵ. $\mathcal{F}$ can be used to construct an $\mathcal{O}$ that solves $\mathtt{FIND}-\mathtt{REP}$.

$\mathcal{O}$ works as follows:

1. Input $\mathbf{y} = (y_1, \ldots, y_n)$ (where $\mathbf{y}$ is uniform over $\mathbb{Z}_q^n$).
2. Generate $\mathbf{A} \leftarrow \mathbb{Z}_q^{n \times k}$.
3. Run $\mathbf{z} \leftarrow \mathcal{F}(\mathbf{y}, \mathbf{A})$.
4. If $\mathsf{dis}(\mathbf{y}, \mathbf{z}) > n-k-c$ output $\perp$.
5. Let $\mathcal{I} = \{i | \mathbf{y}_i = \mathbf{z}_i\}$.
6. Construct parity check matrix of $\mathbf{A}_{\mathcal{I}}$, denoted $H_{\mathcal{I}}$.
7. Find some nonzero row of H_I, denoted $\mathbf{B} = (b_1, \ldots, b_{k+c})$ with associated indices I.
8. Output γ where $\gamma_i = \mathbf{B}_{i'}$ for $i \in \mathcal{I}$ where i' represents the location of i in a sorted list with the same elements as $\mathcal{I}$ and 0 otherwise.

By Lemma 22, $(\mathbf{y}, \mathbf{A})$ is a uniform instance of $\mathtt{BDDE}-\mathtt{RL}(n, k, q, n-k-c, g)$ with probability at least $1 - (q^{n-k}/\mathsf{Vol}(n, n-k-c, q) + k \cdot q^{-(n-k)})$. This means that $\mathcal{I} \geq k+c$. Note for $\mathbf{z}$ to be a codeword it must be that there exists some $\mathbf{x}$ such that $\mathbf{z} = \mathbf{A}\mathbf{x}$ and thus, the parity check matrix restricted to $\mathcal{I}$ is defined and there is some nonzero row. □

Acknowledgements

The authors give special thanks to reviewer comments and feedback. The authors thank James Bartusek, Ryann Cartor, Fermi Ma, and Mark Zhandry and their helpful discussions of their work. The work of Benjamin Fuller is funded in part by NSF Grants No. 1849904 and 1547399. This material is based upon work supported by the National Science Foundation under Grant No. 1801487.

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19020700008. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *International Colloquium on Automata, Languages, and Programming*, pages 403–415. Springer, 2011.

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *Theory of Cryptography*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer Berlin Heidelberg, 2009.

[AHK20] Thomas Agrikola, Dennis Hofheinz, and Julia Kastner. On instantiating the algebraic group model from falsifiable assumptions. In *Advances in Cryptology –EUROCRYPT*, 2020.

[BD20] Zvika Brakerski and Nico Döttling. Hardness of lwe on general entropic distributions. In *Advances in Cryptology —EUROCRYPT*, 2020. `https://eprint.iacr.org/2020/119`.

[BKM$^+$18] Allison Bishop, Lucas Kowalczyk, Tal Malkin, Valerio Pastro, Mariana Raykova, and Kevin Shi. A simple obfuscation scheme for pattern-matching with wildcards. In *Annual International Cryptology Conference*, pages 731–752. Springer, 2018.

[BLMZ19] James Bartusek, Tancrède Lepoint, Fermi Ma, and Mark Zhandry. New techniques for obfuscating conjunctions. In *Eurocrypt*, pages 636–666, 2019. `https://eprint.iacr.org/2018/936`.

[BMvT78] Elwyn Berlekamp, Robert McEliece, and Henk van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384 – 386, May 1978.

[Boy04] Xavier Boyen. Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM conference on Computer and communications security*, CCS '04, pages 82–91, New York, NY, USA, 2004. ACM.

[BPM$^+$92] Peter F Brown, Vincent J Della Pietra, Robert L Mercer, Stephen A Della Pietra, and Jennifer C Lai. An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 18(1):31–40, 1992.

[Bra93] Stefan Brands. Untraceable off-line cash in wallet with observers. In *Annual International Cryptology Conference*, pages 302–318. Springer, 1993.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Obfuscating point functions with multibit output. In *Advances in Cryptology–EUROCRYPT 2008*, pages 489–508. Springer, 2008.

[CFP$^+$16] Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam Smith. Reusable fuzzy extractors for low-entropy distributions. In *Advances in Cryptology – EUROCRYPT*, pages 117–146. Springer, 2016.

[CG99] Ran Canetti and Shafi Goldwasser. An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 90–106. Springer, 1999.

[CKVW10] Ran Canetti, Yael Tauman Kalai, Mayank Varia, and Daniel Wichs. On symmetric encryption and point obfuscation. In *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, pages 52–71, 2010.

[DGG15] Özgür Dagdelen, Sebastian Gajek, and Florian Göpfert. Learning with errors in the exponent. In *ICISC 2015*, pages 69–84. Springer, 2015.

[DGV$^+$16] Jeroen Delvaux, Dawu Gu, Ingrid Verbauwhede, Matthias Hiller, and Meng-Day Mandel Yu. Efficient fuzzy extraction of puf-induced secrets: Theory and applications. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 412–431. Springer, 2016.

[DMQ13] Nico Döttling and Jörn Müller-Quade. Lossy codes and a new variant of the learning-with-errors problem. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 18–34. Springer, 2013.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

[Eli57] Peter Elias. List decoding for noisy channels. 1957.

[FKL18] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In *Advances in Cryptology – CRYPTO*, pages 33–62. Springer, 2018.

[FMR13] Benjamin Fuller, Xianrui Meng, and Leonid Reyzin. Computational fuzzy extractors. In *Advances in Cryptology-ASIACRYPT 2013*, pages 174–193. Springer, 2013.

[GS98] Venkatesan Guruswami and Madhu Sudan. Improved decoding of reed-solomon and algebraic-geometric codes. In *Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on*, pages 28–37. IEEE, 1998.

[Gur10] Venkatesan Guruswami. Introduction to coding theory - lecture 2: Gilbert-Varshamov bound. University Lecture, 2010.

[GZ19] Steven D. Galbraith and Lukas Zobernig. Obfuscated fuzzy hamming distance and conjunctions from subset product problems. In *Theory of Cryptography*, 2019. `https://eprint.iacr.org/2019/620`.

[HRvD$^+$16] Charles Herder, Ling Ren, Marten van Dijk, Meng-Day Yu, and Srinivas Devadas. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. *IEEE Transactions on Dependable and Secure Computing*, 2016.

[IPS09] Yuval Ishai, Manoj Prabhakaran, and Amit Sahai. Secure arithmetic computation with no honest majority. In *Theory of Cryptography Conference*, pages 294–314. Springer, 2009.

[JW99] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Sixth ACM Conference on Computer and Communication Security*, pages 28–36. ACM, November 1999.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with Small Parameters. In *Advances in Cryptology - CRYPTO 2013*, Lecture Notes in Computer Science. 2013.

[MZ11] Marcelo A Montemurro and Damián H Zanette. Universal entropy of word ordering across linguistic families. *PLoS One*, 6(5):e19875, 2011.

[NZ93] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, pages 43–52, 1993.

[Pei06] Chris Peikert. On error correction in the exponent. In *Theory of Cryptography Conference*, pages 167–183. Springer, 2006.

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the thirty-seventh annual ACM Symposium on Theory of Computing*, pages 84–93, New York, NY, USA, 2005. ACM.

[Reg10] Oded Regev. The learning with errors problem (invited survey). In *Proceedings of the 2010 IEEE 25th Annual Conference on Computational Complexity*, pages 191–204. IEEE Computer Society, 2010.

[RS60] Irving S Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, 8(2):300–304, 1960.

[Sha51] Claude E Shannon. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 256–266. Springer, 1997.

[SSF19] Sailesh Simhadri, James Steel, and Benjamin Fuller. Cryptographic authentication from the iris. In *International Conference on Information Security*, pages 465–485. Springer, 2019.

[WB86] Lloyd R Welch and Elwyn R Berlekamp. Error correction for algebraic block codes, December 30 1986. US Patent 4,633,470.

[WL18] Yunhua Wen and Shengli Liu. Robustly reusable fuzzy extractor from standard assumptions. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 459–489. Springer, 2018.

[WLG19] Yunhua Wen, Shengli Liu, and Dawu Gu. Generic constructions of robustly reusable fuzzy extractor. In *IACR International Workshop on Public Key Cryptography*, pages 349–378. Springer, 2019.

A Generic Group Formalism and Analysis

A.1 The Generic Group Model and the Simultaneous Oracle Game

The focus of this section is on proving Theorem 2. Our proof uses the simultaneous oracle game introduced by Bishop et al. [BKM$^+$18, Section 4]. In this game, the adversary is given two oracles $\mathcal{O}_1$ and a second oracle $\mathcal{O}^*$ that is either $\mathcal{O}_1$ or $\mathcal{O}_2$ with probability 1/2. If $\mathcal{O}^* = \mathcal{O}_1$ it is sampled with independent randomness from the first copy. Bishop et al. show that if an adversary cannot distinguish in this game, they cannot distinguish the two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$. Since the adversary has access to two oracles simultaneously it is easier to formalize when the adversary can distinguish: The adversary's distinguishing ability arises directly from repeated responses. The adversary can only notice inconsistency when (i.) one oracle returns a new response and the other does not or (ii.) if both responses are repeated but not consistent with the same prior query.

Definition 9 (Generic Group Model (GGM) [Sho97])**.** *An application in the generic group model is defined as an interaction between a m-attacker $\mathcal{A}$ and a challenger $\mathcal{C}$. For a cyclic group $\mathbb{G}_N$ of order N with fixed generator g, a uniformly random function $\sigma : [N] \to [M]$ is sampled, mapping group exponents in $\mathbb{Z}_N$ to a set of labels $\mathcal{L}$. Label $\sigma(x)$ for $x \in \mathbb{Z}_N$ corresponds to the group element g^x. We consider M large enough that the probability of a collision between group elements under σ is negligible so we assume that σ is injective.*

Based on internal randomness, $\mathcal{C}$ initializes $\mathcal{A}$ with some set of labels $\mathcal{L} = \{\sigma(x_i)\}_{i=0}^{n}$. It then implements the group operation oracle $\mathcal{O}_G(\cdot,\cdot)$, which on inputs σ_1, $\sigma_2 \in [M]$ does the following:

1. *if either σ_1 or σ_2 are not in $\mathcal{L}$, return $\perp$.*
2. *Otherwise, set $x = \sigma^{-1}(\sigma_1)$ and $y = \sigma^{-1}(\sigma_2)$ compute $x+y \in \mathbb{Z}_N$ and return $\sigma(x+y)$, add $\sigma(x+y)$ to $\mathcal{L}$.*

$\mathcal{A}$ is allowed at most m queries to the oracle, after $\mathcal{A}$ outputs a bit which is sent to C which outputs a bit indicating whether $\mathcal{A}$ was successful.

The above structure captures distinguishing games. Search games can be defined similarly. Bishop et. al. formalized the simultaneous oracle game [BKM$^+$18]. The formal structure is as follows.

Definition 10 (Simultaneous Oracle Game [BKM$^+$18] definition 6)**.** *An adversary is given access to a pair of oracles $(\mathcal{O}_M, \mathcal{O}_*)$ where $\mathcal{O}_*$ is drawn from the same distribution as $\mathcal{O}_M$ with probability* 1/2 *(with independent internal randomness) and is $\mathcal{O}_S$ with probability* 1/2*. In each round, the adversary asks the same query to both oracles. The adversary wins the game if they guess correctly the identity of $\mathcal{O}_*$.*

We note that even if the oracles are drawn from the same distribution their handle mapping functions σ, using their independent internal randomness, will respond with distinct handles with overwhelming probability even if their responses represent the same underlying group element. The distributions that the oracles are drawn from represent any internal randomness used to implement the oracle by the challenger in the definition of the generic group model.

In [BKM$^+$18], Bishop et. al. also define two sets $\mathcal{H}_S^t$ and $\mathcal{H}_M^t$ which are the sets of handles returned by the two oracles after t query rounds. They use these sets to define a function $\Phi : \mathcal{H}_S^t \to \mathcal{H}_M^t$. Initially the adversary sets $\Phi(h_S^{t,i}) = h_M^{t,i}$ for each element indexed by i in the initial sets given by the oracles. The adversary can only distinguish if (i.) one oracle returns a new handle, while the other is repeated or (ii.) the two oracles both return old handles that are not consistent under Φ. Hardness of the simultaneous oracle game is sufficient to show that the two games cannot be distinguished. We state a lemma from Bishop et al.:

Lemma 23 ([BKM$^+$18] Lemma 7)**.** *Suppose there exists an algorithm $\mathcal{A}$ such that*

$$|\Pr[\mathcal{A}^{\mathcal{G}_\mathcal{M}}(\mathcal{O}^{\mathcal{G}_\mathcal{M}}) = 1] - \Pr[\mathcal{A}^{\mathcal{G}_S}(\mathcal{O}^{\mathcal{G}_S}) = 1]| \geq \delta.$$

Then an adversary can win the simultaneous oracle game with probability at least $\frac{1}{2} + \frac{\delta}{2}$ for any pair of oracles $(\mathcal{O}_\mathcal{M}, \mathcal{O}_* = \mathcal{O}_\mathcal{M}/\mathcal{O}_\mathcal{S})$.

In the above $\mathcal{A}^{\mathcal{G}_\mathcal{M}}(\mathcal{O}^{\mathcal{G}_\mathcal{M}})$ corresponds to an adversary being initialized with handles from $\mathcal{G}_\mathcal{M}$ and having an oracle to $\mathcal{G}_\mathcal{M}$. $\mathcal{A}^{\mathcal{G}_S}(\mathcal{O}^{\mathcal{G}_S})$ is defined similarly.

Remark 1. *It is convenient for us to change the query capability of the adversary in the simultaneous oracle game. Rather than single group operation queries we allow the adversary to make queries in the form of a vector representing a linear combination of the initial set of handles given by the pair of oracles. Specifically, a query $\mathcal{X} = (c_0, \ldots, c_n)$ is given to both $\mathcal{O}_M$ and $\mathcal{O}_*$ where they compute and return their responses. Each query to this interface can be simulated using a polynomial number of queries to the traditional group oracle.*

Proof of Theorem 2. We begin by noting that since the output range of σ is q^3 the probability of $\sigma(x) = \sigma(y)$ when $x \neq y$ is at most $1/q^2$ so taking a union bound over all q elements, the probability of some collision existing is at most $1/q$. Thus, for the remainder of the proof we restrict to the case when σ is a 1-1 function.

We begin the proof by describing the two oracles we use in the simultaneous oracle game called the **Code** and **Random** Oracles.

Code Oracle. We define a code oracle that responds to queries faithfully. We denote this oracle $\mathcal{O}_c$ (and particular sampled function as σ_c). This oracle picks a message $\mathbf{x}$, uses the generating matrix $\mathbf{A}$ and the error vector $\mathbf{e}$ which is a $(k, \beta) - \mathsf{MIPURS}$ distribution.

The oracle begins by calculating the noisy codeword $\mathbf{b}_1, ..., \mathbf{b}_n$ as $\mathbf{b} = \mathbf{A}\mathbf{x}+\mathbf{e}$. The oracle prepends $b_0 = 1$ (to allow the adversary constant calculations) and sends $(\sigma_c(b_0), \ldots, \sigma_c(b_n))$ to $\mathcal{D}$. When queried with a vector $\chi = (\chi_0, \chi_1, \ldots, \chi_n) \in \mathbb{Z}_q^{n+1}$ the oracle answers with an encoded group element $\sigma_c(\sum_{i=0}^n \chi_i \cdot b_i)$.

Random Oracle. We also define an oracle $\mathcal{O}_r$ that creates $n + 1$ random initial encodings and responds to all distinct requests for linear combinations with distinct random elements. For a sequence of indeterminates $\mathbf{y} = (y_0, y_1, \ldots, y_n)$, this oracle can be described as a table where the left side is a vector representing a linear combination of the indeterminates and the right side is a handle associated with each vector.

When presented a query, if the vector is in the oracle's table, it responds with the handle on the right side of the table. When the query is a new linear combination, it generates a distinct, random handle. The adversary then stores the vector and the handle in the table and sends the handle to $\mathcal{D}$. We denote the handles τ_i to distinguish them from the encoded group elements of the code oracle.

Lemma 24. *In a simultaneous oracle game, the probability that any adversary $\mathcal{D}$, when interacting with group oracles $(\mathcal{O}_c, \mathcal{O}_* = \mathcal{O}_c/\mathcal{O}_r)$ succeeds after m queries is at most*

$$|\Pr[\mathcal{D}(\mathcal{O}_c) = 1] - \Pr[\mathcal{D}(\mathcal{O}^*) = 1]| \leq \gamma \left(\frac{1}{q} + \beta\right)$$

for $\gamma = ((m+n+2)(m+n+1))^2/4$.

Proof. We examine the simultaneous oracle game that the adversary plays between $\mathcal{O}_c$ and $\mathcal{O}^*$. The adversary maintains its function Φ as it makes queries. We also analyze the underlying structure of $\mathcal{O}_c$. Denote the adversary's linear combination as $\gamma || \chi_1, ..., \chi_n$. We distinguish the first element as it is multiplied by 1 leading to an offset in the resulting product. We do this by noticing that for $i \geq 1$, the group element b_i is $\mathbf{A}_i\mathbf{x} + \mathbf{e}_i$ (we use $\mathbf{A}_i$ to denote the ith row of a matrix $\mathbf{A}$):

$$\sum_{i=1}^{n} \chi_i b_i + \gamma = \sum_{i=1}^{n} \chi_i(\mathbf{A}_i \cdot \mathbf{x}) + \sum_{i=1}^{n} \chi_i(\mathbf{e}_i) + \gamma = \langle \chi, \mathbf{A}\mathbf{x} \rangle + \langle \chi, \mathbf{e} \rangle \ + \gamma.$$

Again, $\mathcal{O}_r$ responds to each distinct query with a new handle. This means that there is exactly one occasion to distinguish when $\mathcal{O}_* = \mathcal{O}_c$ or $\mathcal{O}_r$. This is when the handle returned by $\mathcal{O}_c$ is known and $\mathcal{O}_r$ is new. We divide our cases with respect to the linear combination query χ. If χ is not in the null space of the code $\mathbf{A}$, we call this case 1. If χ is in the null space of $\mathbf{A}$ we call this case 2.

Case 1. Initially, $\mathbf{x}$ is both uniform and private. We can write the product of χ and our noisy code word $\mathbf{b}$ as $\chi(\mathbf{b}) = \chi(\mathbf{A}\mathbf{x} + \mathbf{e}) = (\chi\mathbf{A})\mathbf{x} + \chi(\mathbf{e})$. Since $\chi \notin \mathsf{null}(\mathbf{A})$ then for at least one index i there is a $\chi_i \cdot \mathbf{A}_i \neq 0$. Since x has full entropy, then $(\chi_i \mathbf{A}_i)\mathbf{x}_i$ also has full entropy and the sum of the terms has full entropy. After the first query, $\mathbf{x}$ is no longer uniform. With each query, the adversary learns a predicate about the difference of all previous queries, simply that they do not produce the same element. After m queries (and $n+1$ starting handles) there are $\eta = (m+n+1)(m+n+2)/2$ query differences, giving the same number of these equality predicates. Note that the adversary wins if a single of these predicates is 1 meaning we can consider η total values for the random variable, denoted $\mathtt{EQ}$ representing the equality predicate pattern. Then, using a standard conditional min-entropy argument [DORS08, Lemma 2.2b]. Thus,

$$\forall i, \tilde{\mathrm{H}}_\infty(\mathbf{x}_i \mid \mathtt{EQ}, \mathbf{A}) \geq \log q - \log \eta.$$

Thus, it follows that after m queries,

$$\tilde{\mathrm{H}}_\infty(\chi(\mathbf{A}\mathbf{x}) \mid \mathtt{EQ}, \mathbf{A}) \geq \log q - \log \eta.$$

Thus, the probability that this linear combination represents a known value (on average across $\mathbf{a}$) is:

$$\mathop{\mathbb{E}}_{\mathbf{A}, \mathtt{EQ}} \left[\max_z \Pr[(\chi(\mathbf{A}\mathbf{x}) = z \mid \mathbf{A}, \mathtt{EQ}]\right] \leq \frac{\eta}{q}.$$

Case 2. Decomposing the linear combination of the codeword into $\chi(\mathbf{A}\mathbf{x}+\mathbf{e})$ since χ is in the null space of A then the linear combination is just $\mathbf{0} + \langle \chi, \mathbf{e} \rangle$. Since $\mathbf{e}$ is a $(k, \beta) - \mathsf{MIPURS}$ distribution, then an upper bound for the power of the adversary to predict the outcome of the linear combination (and thus the outcome of $\langle \chi, e \rangle + \gamma$) is β. In this case we also lose entropy due to the linear predicates. After m queries, we pay the same $\log \eta$ bits so the probability is increased to $\eta\beta$.

These two cases are mutually exclusive. Thus, to calculate the probability of either of these cases occurring after m queries (and $n+1$ starting handles) we take the sum. There are only q distinct group elements, and therefore handles. Even a handle with full entropy will collide with a known handle with probability equal to the number of known handles over the size of the group. Since each query can only produce one handle, we have η distinct pairs of handles after m queries. So taking a union bound over each query, we upper bound the distinguishing probability for the adversary by

$$\eta \left(\frac{\eta}{q} + \eta\beta \right) = \eta^2 \left(\frac{1}{q} + \beta \right).$$

This completes the proof of Lemma 24 by setting $\gamma = \eta^2$. □

This lemma gives us the distinguishing power of an adversary interacting with our code oracle and our random oracle. Our random oracle never has collisions because it creates fresh handles every time. We now create an oracle that represents a distribution over uniform elements as claimed in Theorem 2. Note that this oracle is different than $\mathcal{O}_r$ which responded to all distinct queries with distinct handles. This third handle initializes n random elements and faithfully represents the group operation. For a fresh query this oracle has probability $1/q$ of returning a previously seen handle. We call this last oracle the uniform oracle. In this case the adversary only distinguishes by seeing a repeated query handle. This probability is at most η/q. To simplify the final result we know this value is at most γ/q since $\gamma = \eta^2$.

Taking the result of this Lemma 24, we can prove Theorem 2 using Lemma 23 (and the modification to the uniform oracle) where

$$\delta/2 \overset{def}{=} \gamma \left(\frac{2}{q} + \beta \right).$$

Since the probability of an adversary winning the simultaneous oracle game is bounded above by

$$1/2 + \delta/2 = 1/2 + \gamma \left(\frac{2}{q} + \beta \right)$$

then

$$\Pr[A(\mathcal{O}_c) = 1] - \Pr[A(\mathcal{O}_r) = 1] < 2\gamma \left(\frac{2}{q} + \beta \right),$$

for $\gamma = ((m+n+1)(m+n+2)/2)^2$. Because $\mathcal{O}_r$ represents the oracle for uniform randomness and $\mathcal{O}_c$ is the oracle for $A\mathbf{x} + \mathbf{e}$, this gives us the result for generic adversaries. □

B Decoding Reed Solomon Codes in the Exponent

The Reed-Solomon family of error correcting codes [RS60] have extensive applications in cryptography. For the field $\mathbb{F}_q$ of size q, a message length k, and code length n, such that $k \leq n \leq q$, define the Vandermonde matrix $\mathbf{V}$ where the ith row, $\mathbf{V}_i = [i^0, i^1, ..., i^k]$. The Reed Solomon Code $\mathbb{RS}(n, k, q)$ is the set of all points $\mathbf{V}\mathbf{x}$ where $\mathbf{x} \in \mathbb{F}_q^k$. Reed-Solomon Codes have known efficient algorithms for correcting errors. We note that for a particular vector $\mathbf{x}$ the generated vector $\mathbf{V}\mathbf{x}$ is a degree k polynomial with coefficients $\mathbf{x}$ evaluated at points $1, ..., n$.

The Berlekamp-Welch algorithm [WB86] corrects up to $(n - k + 1)/2$ errors in any codeword in the code. List decoding provides a weaker guarantee. The algorithm instead vectors a list containing codewords within a given distance to a point, the algorithm may return 0, 1 or many codewords [Eli57]. The list decoding algorithm of Guruswami and Sudan [GS98] can find all codewords within Hamming distance $n - \sqrt{nk}$ of a given word. Importantly, algorithms to correct errors in Reed-Solomon codes rely on nonlinear operations. Like with Random Linear Codes we consider hardness of constructing an oracle that performs bounded distance decoding.

Problem $\mathtt{BDDE} - \mathtt{RS}(n, k, q, c, g)$, or Bounded Distance Decoding in the exponent of Reed Solomon codes.

Instance A known generator g of $\mathbb{Z}_q^*$. Define $\mathbf{e}$ as a random vector of weight c in $\mathbb{Z}_q^*$. Define $g^{\mathbf{y}} = g^{\mathbf{V}\mathbf{x}+\mathbf{e}}$ where $\mathbf{x}$ is uniformly distributed. Input is $g^{\mathbf{y}}$.

Output Any codeword $g^{\mathbf{z}}$ where $\mathbf{z} \in \mathbb{RS}(n, k, q)$ such that $\mathsf{dis}(g^{\mathbf{y}}, g^{\mathbf{z}}) \le c$.

Theorem 25. *For any positive integers n, k, c, and q such that $q \ge n^2/4$, $c \le n + k$, $k \le n$ and a generator g of the group $\mathbb{G}_q$, if an efficient algorithm exists to solve $\mathtt{BDDE} - \mathtt{RS}(n, q, k, n - k - c, g)$ with probability ϵ (over a uniform instance and the randomness of the algorithm), then an efficient randomized algorithm exists to solve the discrete log problem in $\mathbb{G}_q$ with probability*

$$\epsilon' \ge \begin{cases} \epsilon\left(1 - \frac{2q^c}{\binom{n}{k+c}}\right) & \frac{n^2}{2} \le q \\ \epsilon\left(1 - \frac{cq^c}{\binom{n}{k+c}}\right) & \frac{n^2}{4} \le q < \frac{n^2}{2} \end{cases}.$$

Proof. Like Theorem 21 the core of our theorem is a bound on the probability that a random point is close to a Reed-Solomon code.

Lemma 26. *For any positive integer $c \le n - k$, define $\alpha = \frac{4q}{n^2}$, and any Reed-Solomon Code $\mathbb{RS}(n, k, q)$,*

$$\Pr_{\mathbf{y}}[\mathsf{dis}(\mathbf{y}, \mathbb{RS}(n, k, q)) > n - k - c] \le \frac{q^c}{\binom{n}{k+c}} \alpha^{-c} \sum_{c'=0}^{c} \alpha^{c'}$$

where the probability is taken over the uniform choice of $\mathbf{y}$ from $\mathcal{G}^n$.

Proof of Lemma 26. A vector $\mathbf{y}$ has distance at most $n - k - c$ from a Reed-Solomon code if there is some subset of indices of size $k + c$ whose distance from a polynomial is at most $k - 1$. To codify this notion we define a predicate which we call *low degree.* A set S consisting of ordered pairs $\{\alpha_i, x_i\}_i$ is low degree if the points $\{(\alpha_i, \log_g x_i)\}_{i\in S}$ lie on a polynomial of degree at most $k - 1$. Define $\mathcal{S} = \{S \subseteq [n] : |S| = k + c\}$. For every $S \in \mathcal{S}$, define Y_S to be the indicator random variable for if S satisfies the low degree condition taken over the random choice of $\mathbf{y}$. Let $Y = \sum_{S\in\mathcal{S}} Y_S$.

For all $S \in \mathcal{S}, \Pr[Y_S = 1] = q^{-c}$, because any k points of $\{(\alpha_i, \log_g x_i)\}_{i\in S}$ define a unique polynomial of degree at most k. The remaining c points independently lie on that polynomial with probability $1/q$. The size of $\mathcal{S}$ is $|\mathcal{S}| = \binom{n}{k+c}$. Then by linearity of expectation, $\mathbb{E}[Y] = \binom{n}{k+c}/q^c$. Now we use Chebyshev's inequality,

$$\Pr_{\mathbf{y}}[\mathsf{dis}(\mathbf{y}, \mathbb{RS}(n, k, q)) > n - k - c] = \Pr[Y = 0] \le \Pr[|Y - \mathbb{E}[Y]| \ge \mathbb{E}[Y]] \le \frac{\mathrm{Var}(Y)}{\mathbb{E}[Y]^2}.$$

It remains to analyze $\mathrm{Var}(Y) = \mathbb{E}[Y^2] - \mathbb{E}[Y]^2$. To analyze this variance we split into cases where the intersection of Y_S and $Y_{S'}$ is small and large. Consider two sets S and S' and the corresponding indicator random variables Y_S and $Y_{S'}$. If $|S \cap S'| < k$ then $\mathbb{E}[Y_S|Y_{S'}] = \mathbb{E}[Y_S]$ and $\mathbb{E}[Y_S Y_{S'}] = \mathbb{E}[Y_S]\,\mathbb{E}[Y_{S'}]$. This observation is crucial for security of Shamir's secret sharing [Sha79]. For pairs S, S' where $|S \cap S'| \geq k$, we introduce a variable c' between 0 and c to denote $c' = |S \cap S'| - k$. For such S, S' instead of computing $\mathbb{E}[Y^2] - \mathbb{E}[Y]^2$ we just compute $\mathbb{E}[Y^2]$ and use this as a bound. For each c' we calculate $\mathbb{E}[Y_S Y_{S'}]$ where $|S \cap S| = k + c'$. The number of pairs can be counted as follows: $\binom{n}{k+c}$ choices for S, then $\binom{k+c}{c-c'}$ choices for the elements of S not in S' which determines the $k + c'$ elements that are in both S and S', and finally $\binom{n-k-c}{c-c'}$ to pick the remaining elements of S' that are not in S. So the total number of pairs is $\binom{n}{k+c}\binom{k+c}{c-c'}\binom{n-k-c}{c-c'}$. Using these observations, we can upper bound the variance $\mathrm{Var}(Y)$ for our random variable Y:

$$\begin{aligned}
\mathrm{Var}(Y) &= \sum_{S,S' \in \mathcal{S}} (\mathbb{E}[Y_S Y_{S'}] - \mathbb{E}[Y_S]\,\mathbb{E}[Y_{S'}]) \\
&= \sum_{c'=0}^{c} \sum_{\substack{S,S' \in \mathcal{S} \\ |S \cap S'| = k+c'}} (\mathbb{E}[Y_S Y_{S'}] - \mathbb{E}[Y_S]\,\mathbb{E}[Y_{S'}]) \\
&\leq \sum_{c'=0}^{c} \sum_{\substack{S,S' \in \mathcal{S} \\ |S \cap S'| = k+c'}} (\mathbb{E}[Y_S Y_{S'}]) = \sum_{c'=0}^{c} \sum_{\substack{S,S' \in \mathcal{S} \\ |S \cap S'| = k+c'}} (\frac{1}{q^{2c-c'}})
\end{aligned}$$

Here the last line follows by observing that for both Y_S and $Y_{S'}$ to be 1 they must both define the same polynomial. Since S and S' share $k + c'$ points, there are $(k + c) + (k + c) - (k + c') = k + 2c - c'$ points that must lie on the at most $k - 1$ degree polynomial, and any k points determine the polynomial, and the remaining $2c - c'$ points independently lie on the polynomial with probability $1/q$ then the probability that this occurs is $1/q^{2c-c'}$. Continuing one has that,

$$\begin{aligned}
\mathrm{Var}(Y) &\leq \frac{1}{q^{2c}} \sum_{c'=0}^{c} \sum_{\substack{S,S' \in \mathcal{S} \\ |S \cap S'| = k+c'}} (q^{c'}) \\
&= \frac{1}{q^{2c}} \sum_{c'=0}^{c} \left(q^{c'} \binom{n}{k+c} \binom{k+c}{c-c'} \binom{n-k-c}{c-c'} \right) \\
&= \left[\binom{n}{k+c} \frac{1}{q^c} \right] \frac{1}{q^c} \sum_{c'=0}^{c} \left(q^{c'} \binom{n}{k+c} \binom{k+c}{c-c'} \binom{n-k-c}{c-c'} \right) \\
&= \frac{\mathbb{E}[Y]}{q^c} \sum_{c'=0}^{c} \left(q^{c'} \binom{k+c}{c-c'} \binom{n-k-c}{c-c'} \right)
\end{aligned}$$

We bound the size of $\binom{k+c}{c-c'}\binom{n-k-c}{c-c'}$ by observing that the sum of the top terms of the choose functions is n and the product of two values with a known sum is bounded by the product of their average, in this case $n/2$. We also use the upper bound of the choose function where $\binom{a}{b} \leq a^b$ to arrive at the bound that

$$q^{-c}\sum_{c'=0}^{c}\left(q^{c'}\binom{k+c}{c-c'}\binom{n-k-c}{c-c'}\right) \leq \frac{1}{q^c}\sum_{c'=0}^{c}(q^{c'}(n/2)^{2c-2c'})$$
$$= \left(\frac{(n/2)^2}{q}\right)^c \sum_{c'=0}^{c}\left(\frac{q}{(n/2)^2}\right)^{c'}.$$

The proof then follows using our bound for variance by defining $\alpha = 4q/n^2$. This completes the proof of Lemma 26. □

The remainder of the proof is similar to the proof of Theorem 21. $\mathcal{A}$ works as follows: on input uniform $\mathbf{y}$ run $\mathcal{D}(g, \mathbf{y})$ which is a good list decoder for Reed Solomon (not the code no longer needs to be provided). By Lemma 26, $(g, \mathbf{v})$ is an instance of $\texttt{BDDE} - \texttt{RS}_{q,\mathcal{E},k,n-k-c}$ with probability at least

$$1 - \frac{q^c}{\binom{n}{k+c}}\alpha^{-c}\sum_{c'=0}^{c}\alpha^{c'}.$$

Then conditioned on this event, the instance is uniform, and $\mathcal{D}$ (with probability ϵ) outputs some $\mathbf{z}$ where $\mathsf{dis}(\mathbf{z}, \mathbf{y}) \leq n-k-c$. Define the set $E \subseteq [n]$ as the set of indices i such that $\mathbf{y}_i = \mathbf{z}_i$. Note that $|E| \geq k+1$. From any subset E of size k it is possible given $\{\mathbf{y}_i\}_{i\in\mathcal{I}}$ it is possible to linearly interpolate any $\mathbf{y}_j$ for $1 \leq j \leq n$. Thus for any $k+1$ positions, it is possible to find $\gamma_{i_1}, ..., \gamma_{i_{k+1}}$ such that for any codeword $\mathbf{z}$, $\sum_{i_j} \mathbf{z}_{i_j}\gamma_{i_j} = 0$. Define $\gamma_i = 0$ when $i \notin E$. Then one has that

$$\prod_{i_j \in E} v_i^{\gamma_{i_j}} = g^{\sum_{i_j\in E}\mathbf{y}_{i_j}\gamma_{i_j}} = g^{\sum_{i_j\in E}\mathbf{z}_{i_j}\gamma_{i_j}} = 1.$$

That is, $(\gamma_1, \ldots, \gamma_n)$ is a solution to $\texttt{FIND} - \texttt{REP}$. The parameters in the Theorem follow when $1 \leq \alpha < 2$ by noting that

$$\alpha^{-c}\sum_{c'=0}^{c}\alpha^{c'} \leq \alpha^{-c}(c \cdot \alpha^c) = c.$$

Parameters in Theorem 25 follow in the case when $\alpha = 4q/n^2 \geq 2$ by noting that:

$$\alpha^{-c}\sum_{c'=0}^{c}\alpha^{c'} = \alpha^{-c}\Big(\frac{\alpha^{c+1}-1}{\alpha-1}\Big) = \Big(\frac{\alpha-\alpha^{-c}}{\alpha-1}\Big) \leq 2.$$

□