Code Offset in the Exponent

Luke Demarest* Benjamin Fuller† Alexander Russell‡

October 16, 2019

Abstract

Fuzzy extractors derive stable keys from noisy sources. The standard construction is the code offset: the noisy secret w is stored in a one-time pad which is sampled as a codeword from an error-correcting code (Juels and Wattenberg, CCS 1999). Information-theoretic analysis of this construction has weaknesses: 1) it leaks information about w and 2) does not allow reuse (Boyen, CCS 2004). Fuller, Meng, and Reyzin (Asiacrypt 2013) substitute a random linear code and reduce to learning with errors when the symbols of w are independent and uniform.

We introduce *code offset in the exponent*: a group generator is raised to the output of the code offset (instantiated with a random linear code). If the resulting vector is indistinguishable from random group elements, the construction 1) leaks nothing about w 2) is reusable and 3) corrects up to a subconstant fraction of errors (in the Hamming metric).

We characterize what error distributions make code offset indistinguishable from random group elements in the generic group model. This corresponds to error distributions that make learning with errors hard in the generic group model. The construction and adversary are both provided with the code description.

It suffices for all vectors in the null space of a random linear code to have an unpredictable inner product with the noisy distribution. Our primary result is: the condition is satisfied by all distributions with minentropy that is larger than log of the size of the nullspace of the code by any super logarithmic amount. The condition is also satisfied by structured distributions including:

1. Distributions with independent symbols that have super logarithmic minentropy including the discretized Gaussian and the uniform interval.
2. Binary distributions lifted by multiplying by a random vector. It suffices for subsets of the binary string (whose size is the nullity of the code) to be unlikely to be all zero. Using this modification, we improve decoding over Canetti et al. (Eurocrypt 2016).

Our construction also yields a more flexible construction of pattern matching obfuscation (Bishop et al., Crypto 2018). Lastly, we provide standard model results, showing hardness of bounded distance decoding of random linear codes with uniform input point, quantitatively improving prior bounds of Peikert (TCC 2006).

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

*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

Stable key generation from noisy physical sources enables effective authentication and access control. Despite inherent noise, one needs to derive a stable key. Wyner [Wyn75] first asked when interactive protocols exist for this task (followed by the seminal work of Bennett, Brassard, and Robert [BBR88]). We focus on non-interactive protocols which are appropriate for a user authenticating to a device.

In the non-interactive case, the relevant cryptographic primitive is a fuzzy extractor [JW99, JS06, DORS08]. A fuzzy extractor is two algorithms. The first algorithm generate, or Gen, takes in a value w, producing a cryptographic key key and a helper value pub. The second algorithm reproduce, or Rep, takes in a value $\mathbf{w}'$ and the value pub. Correctness says that Rep should output key whenever $\mathbf{w}$ and $\mathbf{w}'$ are close enough, denoted by t, according to some distance, dis. Security says that key should appear random to an adversary that knows pub.

Most fuzzy extractors use a variant of the code-offset construction [JW99] (and most variants are equivalent [DGV+16]). For some linear error-correcting code $\mathbf{A}$, Gen samples a random $\mathbf{x}$ and outputs $\mathsf{pub} = (\mathbf{A}, \mathbf{A}\mathbf{x} + \mathbf{w})$. In Rep one computes $\mathbf{A}\mathbf{x} + (\mathbf{w} - \mathbf{w}')$. If $\mathbf{w}$ and $\mathbf{w}'$ are within distance t, this value corresponds to a codeword with at most t errors. By decoding the error correcting code one can find $\mathbf{x}$ and $\mathbf{w}$. One can then use a randomness extractor [NZ93] on either $\mathbf{x}$ or $\mathbf{w}$ to get a uniform key. Suppose that $\mathbf{A} \in \mathbb{F}_q^{n \times k}$, the security analysis states that the entropy of $\mathbf{w}$ decreases by at most $(n-k)\log q$ bits.[1] This construction has two major security weaknesses:

1. *Leakage* it may leak sensitive attributes of the value $\mathbf{w}$ and
2. *One-time* a user cannot safely enroll a single value $\mathbf{w}$ multiple times, called a *reusable fuzzy extractor* [Boy04]. Reuse matters in practice as people have a limited number of usable biometrics.

Fuller et al. [FMR13] considered the same construction instantiated with a random code. When $\mathbf{A}$ is random, the value $\mathsf{pub} = \mathbf{A}\mathbf{x} + \mathbf{w}$ is a learning with errors (LWE) instance if $\mathbf{w}$ is drawn from a LWE error distribution [Reg10a]. LWE error distributions include the discretized Gaussian [Reg10b], uniform interval [DMQ13], and a uniform bit [MP13] (when n is not much larger than k). These distributions have independent and identically distributed symbols. Physical sources demonstrate dependence between symbols [SSF18, HRvD+16].

Fuller et al. [FMR13] use a simple guess and check decoding algorithm. They sample many subsets of size k hoping one subset has no errors; if a sampled subset has no errors one can find $\mathbf{x}$ by solving the linear system restricted to the subset. This algorithm is computable using linear operations (assuming $\mathbf{A}$ is public). Thus, correctness is not effected by embedding the vector $\mathbf{A}\mathbf{x} + \mathbf{w}$ into a "hard" group. Let r be a random generator of a prime order group, one could output

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

We call this construction *code-offset in the exponent*. Peikert [Pei06] showed when $\mathbf{A}$ is a Reed-Solomon code decoding is hard in the generic group model [Sho97] when $\mathbf{w}$ is distributed as t-nonzero values that are uniformly and independently distributed. An adversary can repeatedly try to find subsets of size k without any errors and perform a linear operation to recover the original codeword [CG99], succeeding when $tk = \Theta(n \log n)$. Peikert showed this is tight, that no attacker can distinguish when $tk = \omega(n \log n)$. However, this result still requires error symbols to be independent and uniform.

[1] This bound is tight for some distributions: Consider a distribution $\mathbf{W}$ with a single point from each coset of $\mathbf{A}$, then the coset of pub determines $\mathbf{w}$.

In this work, we instantiate code-offset in the exponent with a random linear code and ask for what distributions is this construction secure? We primarily this question in the generic group model [Sho97]. Note this question is equivalent to asking what error distributions make LWE hard in the generic group model. To the best of our knowledge this is first time this question has been considered.[2]

Caveat Achieving meaningful results in the generic group model requires a superpolynomial size group and eliminates the ability to round. So most LWE based constructions cannot be easily instantiated in the exponent. One notable exception is collision-resistant hashing [GGH11] (of course, there are collision resistant hashes from much weaker group theoretic assumptions).

1.1 Our Contributions

In the generic group model, we establish that distinguishing code offset in the exponent from a random vector of group elements (given $\mathbf{A}$ in the clear) is hard for a broad class of distributions $\mathbf{W}$ which we call $(k, \beta) - \mathsf{MIPURS}$ or *maximum inner product unpredictable over random subspace* distributions (Theorem 2). Specifically, a distribution $\mathbf{W}$ (taking values in $\mathbb{F}_q^n$) is $(k, \beta) - \mathsf{MIPURS}$ if—with high probability in selection of a random subspace $\mathbf{B} \in \mathbb{F}_q^{n \times n-k}$, every nonzero $\mathbf{b} \in \mathbf{B}$ it is hard to predict the inner product with $\mathbf{w}$. That is,

$$\forall g, \Pr_{\mathbf{w} \leftarrow \mathbf{W}, \mathbf{B}}[\langle \mathbf{b}, \mathbf{w} \rangle) = g | \mathbf{b} \in \mathbf{B}] \leq \beta.$$

This is formally defined in Definition 1. While the notation is unwieldy, the intuition is natural: the null space of $\mathbf{A}$ is a random subspace, if a distribution is not MIPURS one can find a vector $\mathbf{b}$ in the null space whose inner product with $\mathbf{w}$ is predictable, thus predicting $\langle \mathbf{b}, \mathbf{A}\mathbf{x} + \mathbf{w} \rangle = \langle \mathbf{b}, \mathbf{w} \rangle \stackrel{?}{=} g$. This is not the case for a uniform distribution, the value $\langle \mathbf{b}, \mathbf{U} \rangle$ is uniform. Thus $\mathbf{b}$ serves as a way to distinguish $\mathbf{A}\mathbf{x} + \mathbf{w}$ from U. That is, if the error distribution is not MIPURS there is an information-theoretic distinguisher. The only non-efficient part of the adversary is mapping a subspace $\mathbf{B}$ to the vector $\mathbf{b}$ and point z. Thus, with respect to information-theoretic generic group adversaries the condition is necessary and sufficient.

The proof of Theorem 2 is relatively straightforward. We consider our primary contribution the characterization of the MIPURS condition. Under appropriate parameterization of q the minentropy of all predictable distributions is at most $\log(\mathtt{poly}(n)q^{n-k})$. An informal version of the result follows (Corollary 3):

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

This result is tight, for any $d = \mathtt{poly}(n)$ we can build a distinguisher. Consider the following distribution $\mathbf{W}$ with a two stage sampling procedure:

1. Pick $i \leftarrow \{1, ..., d\}$ for some polynomial size d.
2. Sample $\mathbf{W}_i$ which is a coset g_i of some linear space of dimension $n - k$.

This distribution has support size $\log(dq^{n-k})$. For a random $n - k$ dimensional $\mathbf{B}$, with high probability $\mathbf{b} \in \mathbf{B} \cap \mathbf{W}_i$ where $\mathbf{b} \neq \mathbf{0}$ for each space $\mathbf{W}_i$. The adversary can calculate this $\mathbf{b}$. Then the adversary

[2] 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.

then just predicts a random g_i as the resulting inner-product. The construction requires superpolynomial q and does not achieve decoding for $t = \Theta(n)$ (see Sec. 1.1.1), otherwise it is "tight" with respect to providing computational security for general entropy.

Proof Idea It is well known how to measure the probability of intersection of subspaces if one subspace is random. There are three ways our setting differs from measuring this intersection. First, the distribution $\mathbf{W}$ is not linear, second the adversary doesn't have to "nullify" the entire space $\mathbf{B}$ only a single vector, and lastly, the adversary can predict any inner product not just 0.

Our proof is dedicated to removing these three obstacles in turn. First we try and place on a bound on how large a set E can be while being predictable in the MIPURS game. We introduce a measure of E's likelihood of intersecting heavily with a low dimensional subspace. We then bound this quantity using only the size of E (Lemma 4). This allows us to control how many vectors in E are mapped to 0 by every vector in the worst subspace. In Lemma 5 we switch from measuring how linear E is with respect to the worst case subspace to how linear E is with respect to the worst vector in an average case subspace. We show the adversary can't do much better on a single vector $\mathbf{b}$ as long as its chosen from a random $\mathbf{B}$.

The above argument considers the adversary predicting an inner product of 0, this can be transformed to an arbitrary inner product using standard techniques with a loss in parameters (Theorem 4). Once we have a bound on how large a predictable E, another superlogarithmic factor guarantees that all distributions $\mathbf{W}$ with enough minentropy are not predictable.

In addition to minentropy, we show other distributions are MIPURS. These distributions are important for the connections to LWE and fuzzy extractors; 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 variants of LWE are hard in the generic group model. These results hold for an arbitrary polynomial number of samples. Interestingly, Micciancio and Peikert [MP13] considered a uniform bit error for a restricted number of samples ($n = k + \Omega(1/\log k)$). They argued that this restriction is necessary due to an attack by Arora and Ge [AG11] which distinguishes when the error distribution has a constant number of values with a polynomial number of samples (requiring $n = \Theta(k^2)$ when $\mathbf{w}_i \in \{0, 1\}$). We observe this attack is fully generic relying on linearization of polynomials and Gaussian elimination.

2. **Location** Distributions where errors are either zero or random. Critically for applications, 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]. While Peikert considered uniform locations, we show a sufficient condition for security is that each subset of size k has an overwhelming probability of including a nonzero error.

1.1.1 Implications for fuzzy extractors

When $\mathbf{W}$ is a $(k, \beta) - \mathsf{MIPURS}$ distribution for a code with dimension k and $\beta = \mathtt{ngl}(n)$ then code-offset in the exponent is a secure fuzzy extractor for $\mathbf{W}$ in the generic group model. Showing this requires one additional step of key extraction, we use 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 is a reusable fuzzy extractor. Canetti and Goldwasser's result [CG99] says linear decoding is efficient when $tk = O(n \log n)$, for k that is just $\omega(\log n)$ this allows 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.[3] As k decreases fewer distributions are $(k, \beta) - \mathsf{MIPURS}$ for $\beta = \mathtt{ngl}$.

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 requires more study.

A unifying construction Our construction unifies multiple 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 $\mathbf{W}$ 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 Fuller et al. [FMR13] is known to be secure.

Canetti et al. [CFP+16] presented a fuzzy extractor that explicitly places specific subsets in a digital locker [CD08]. This construction explicitly writes each subset to be tested. To achieve meaningful error tolerance for an actual biometric, millions of these lockers are required [SSF18]. Their construction is secure when a random subset of bits is hard to predict (Definition 7).

A binary $w \in \{0,1\}^n$ where subsets are hard to predict can be amplified into a **Location** source, whose zero error positions may be correlated. If w has low weight, one can multiply w by a uniform random vector e. However, if w often has high weight this transform requires modification; see the discussion at the end of Section 4.

The code-offset in the exponent construction intentionally allows more flexibility in subset testing. Our analysis requires all subsets of have entropy see Definition 2.[4] The construction of Canetti et al. required an average subset to have entropy, see Definition 7. 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 Z (correlated with the reading $\mathbf{W}$) that can predict the error rate in a symbol $\mathbf{W}_i$. When Z_i is large this means a bit of $\mathbf{W}_i$ is less likely to differ. Examples include the magnitude of a convolution in the iris [SSF18] 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 $2 * 10^6$ 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 $\mathbf{W}$. Our construction can securely use confidence information since it is only needed at reproduction time.

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

[4] 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 has nonzero coordinates at just subset locations. However, the number of such subsets is very low so we keep our condition as all subsets having entropy.

1.1.2 Other Contributions

We present two secondary contributions: an application to pattern matching obfuscation and standard model results that show hardness of decoding random linear codes in the exponent assuming the hardness of discrete log.

Pattern Matching Obfuscation. In a recent work, Bishop et al. [BKM+18] show how to obfuscate a pattern $\mathbf{v}$ where each $\mathbf{v}_i \in \{0, 1, \perp\}$ indicating 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):[5]

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

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:

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

We show it is sufficient for probability distribution $\mathbf{s}$ 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.

Standard Model Results Peikert showed hardness of decoding Reed-Solomon codes in the standard model [Pei06, Theorem 3.1]. We improve Peikert's result, showing hardness of decoding for both random linear codes (Theorem 7) and Reed-Solomon codes (Theorem 9).[6] Both results provide a small improvement of 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

[5] 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.

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

discrete log becomes harder, creating a tension between these parameters. Peikert's result requires that $q \leq \binom{n}{k+1}/n^2$. In our application to the fuzzy extractors we consider small k for which $k = \omega(\log n)$. 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 7 and 9 consider an adversary that performs error correction: given $g^{\mathbf{y}}$ it returns $g^{\mathbf{z}}$ where the distance between $\mathsf{dis}(\mathbf{y}, \mathbf{z}) \leq t$ and $g^{\mathbf{z}}$ is a codeword. 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 $\mathbf{\Lambda}$ such that $g^{\mathbf{z}} = g^{\mathbf{\Lambda y}}$. The model is weaker than the generic group model as the adversary is allowed to see the elements $g^{\mathbf{y}}$ before creating $\mathbf{\Lambda}$. 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 $g^{\mathbf{z}} = g^{\Lambda \mathbf{y}}$. Thus, decoding is a problem where the algebraic model appears weaker than the generic group model.

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. Finally Section 6 shows hardness of decoding high entropy errors in the standard model.

2 Preliminaries

For random variables X_i over some alphabet $\mathcal{Z}$ we denote the tuple by $X = (X_1, ..., X_n)$. For a set of indices J, X_J denotes the restriction of X to the indices in J. For a vector $\mathbf{v}$ we denote the ith entry v_i. The *min-entropy* of X is $\mathrm{H}_\infty(X) = -\log(\max_x \Pr[X = x])$. The *average (conditional)* min-entropy [DORS08, Section 2.4] of X given Y is

$$\tilde{\mathrm{H}}_\infty(X \mid Y) = -\log(\mathop{\mathbb{E}}_{y \in Y} \max_x \Pr[X = x \mid Y = y]).$$

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 let $\cdot_c$ denote componentwise multiplication. Usually, we use capitalized letters for random variables and corresponding lowercase letters for their samples. In our theorems we consider a security parameter λ, when we use the term negligible and super polynomial, we assume other parameters are functions of λ. We elide this notation the dependence of other parameters on λ.

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* $\mathbf{B} : \mathbb{F}_q^{n-k} \to \mathbb{F}_q^n$ *denote uniform random linear operator (drawn independently of* $\mathbf{e}$*). We say that* $\mathbf{e}$ *is an* $(k, \beta) - \mathsf{MIPURS}$ *distribution if*

$$\mathop{\mathbb{E}}_{\mathbf{B}}\left[\min \mathbf{b} \in \mathtt{span}(\mathbf{B}) \setminus \mathbf{0}, \max_z \Pr[\langle \mathbf{b}, \mathbf{e}\rangle = z]\right] \leq \beta\,.$$

Theorem 2. *Let* λ *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* $\mathbf{w}$ *be a* $(k, \beta) - \mathsf{MIPURS}$ *distribution. Let* $\mathbf{U} \in (\mathbb{F}_q)^n$ *be uniformly distributed. Let* Σ *be a set of generic groups with domain of size* q*. Then for all adversaries* $\mathcal{D}$ *making at most* m *queries*

$$\Pr_{\sigma \xleftarrow{\$} \Sigma}\left[\mathcal{D}^\sigma(\mathbf{A}, \sigma(\mathbf{A}\mathbf{x} + \mathbf{w})) = 1\right] - \Pr[\mathcal{D}^\sigma(\mathbf{A}, \sigma(\mathbf{U})) = 1] < \gamma\left(\frac{2}{q} + \beta\right)$$

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

The proof of Theorem 2 is relatively straightforward and delayed until 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

The definition of a MIPURS distribution (Def 1) is admittedly unwieldily. It considers a property of a vector $\mathbf{w}$ with respect to a random matrix. In this section we distributions which satisfy this property. Since the general entropy case is the most technically involved we introduce the distributions where the analysis is straightforward first. We begin with distributions where each component of $\mathbf{w}$ is independent and contributes some entropy. The general minentropy case follows in Section 3.2.

To codify notation, we work in a prime order group of size q, a random linear code $\mathbf{A} \in \mathbb{F}_q^{n\times k}$. Since the code is generated uniformly and independently of other events, the null space can be represented by a random matrix $\mathbf{B} \in \mathbb{F}_q^{n\times n-k}$. As long as the adversary chooses linear tests in $\mathtt{span}(\mathbf{A})$ that are not identically $\mathbf{0}$ we can rely on the entropy in $\mathbf{x}$ to provide security. However, when $\mathbf{b} \in \mathtt{span}(\mathbf{b})$, we can only rely on the distribution of errors $\mathbf{w}$ to provide security.

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, Reg10a], and a uniform interval introduced by Döttling and Müller-Quade [DMQ13]. We show that these common LWE distributions fit within our MIPURS characterization.

Lemma 1. *Let* $\mathbf{w} = \mathbf{w}_1, ..., \mathbf{w}_n \in \mathbb{F}_q$ *be a distribution where each* w_i *is independently sampled. Let* $\alpha = \min_i \mathrm{H}_\infty(\mathbf{w}_i)$*. Let* $\ell \in \mathbb{Z}^+ \cup \{0\}$ *be some free parameter. Then* $\mathbf{w}$ *is a* $(k, \beta) - \mathsf{MIPURS}$ *distribution for*

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

Proof. We use $\mathbf{A} \in \mathbb{F}_q^{n\times k}$ to represent the random matrix from the definition of a MIPURS distribution and let $\mathbf{B} \in \mathbb{F}_q^{n\times n-k}$ represent its null space. We start by computing the minimum distance of $\mathbf{B}$. We

consider the probability that there exists some $\mathbf{b} \in \mathtt{span}(\mathbf{B})$ such that $\mathtt{wt}(\mathbf{b}) < k - l$. That single $\mathbf{b} \neq \mathbf{0}$ is uniformly distributed thus

$$\Pr_{\mathbf{b},\mathbf{B}}[\mathtt{wt}(\mathbf{b}) \leq k - l - 1] = \frac{\sum_{j=0}^{k-l-1} \binom{n}{j} q^j}{q^n} \leq \frac{(k-l-1)\binom{n}{k-\ell-1} q^{k-\ell-1}}{q^n} = \frac{(k-l-1)\binom{n}{k-\ell-1}}{q^{n-k+\ell+1}}.$$

Taking the union bound over all q^{n-k} possible $\mathbf{b}$ the probability of one such $\mathbf{b}$ existing is at most

$$\Pr_{\mathbf{B}}[\exists \mathbf{b} \neq 0 \in \mathbf{B} \wedge \mathtt{wt}(\mathbf{b}) \leq k - l] \leq \frac{(k-l-1)\binom{n}{k-\ell-1}}{q^{\ell+1}}.$$

In the case, when all $\mathbf{b} \neq 0$ of $\mathbf{B}$ have weight at least $k - \ell$ then :

$$\max_z \{\Pr_{\mathbf{B}}[\langle \mathbf{b}, \mathbf{w} \rangle = z | \mathbf{b} \neq 0 \wedge \mathbf{b} \in \mathbf{B}]\} \leq 2^{-\alpha}.$$

In the above, we argue that since the components of $\mathbf{w}$ are independent, predicting $\langle \mathbf{b}, \mathbf{w} \rangle$ is at least as hard as predicting $\mathbf{w}_i$ for each i such that $\mathbf{b}_i \neq 0$. This can be seen by fixing $\mathbf{b}$ and $\mathbf{w}_j$ for $j \neq i$ and noting that the value of $\mathbf{w}_i$ then uniquely determines $\langle \mathbf{b}, \mathbf{w} \rangle$. Thus, for i such that $\mathbf{b}_i \neq 0$,

$$\max_z \Pr[\langle \mathbf{b}, \mathbf{w} \rangle = z] \leq \max_z \Pr[\mathbf{w}_i = z].$$

We must take $\min_i \mathrm{H}_\infty(\mathbf{w}_i)$ as the adversary can choose $\mathbf{b}$ and thus the positions of $\mathbf{w}$ they are predicting. Note independence is crucial in the above argument. The argument follows by assuming that there exists some $\mathbf{w}$ such that $\langle \mathbf{b}, \mathbf{w} \rangle$ is constant in the case when there exist low weight vectors $\mathbf{b}$. □

Location Sources. The second family of error distributions we consider are $\mathbf{e}'$ given by the coordinate-wise product of a uniform vector $\mathbf{e} \in \mathbb{F}_q^n$ and a "selection vector" $\mathbf{w} \in \{0,1\}^n$: that is, $e'_i = e_i \cdot_c \mathbf{w}_i$ where $\mathbf{w}$ is assumed to be unpredictable on all large enough subsets ($\cdot_c$ is componentwise multiplication). More formally, we introduce a notion called subset entropy:

Definition 2. *Let a source $\mathbf{W} = W_1, \ldots, W_n$ consist of n-bit binary strings. For some parameters k, α we say that the source $\mathbf{W}$ is has (α, k)-**entropy subsets** if $\mathrm{H}_\infty(\mathbf{W}_{j_1}, \ldots, \mathbf{W}_{j_k}) \geq \alpha$ for any $1 \leq j_1, \ldots, j_k \leq n$.*

Lemma 2. *Let $\ell \in \mathbb{Z}^+ \cup \{0\}$ and $k \in \mathbb{Z}^+$ be some free parameters. Let $\mathbf{W} \in \{0,1\}^n$ be a distribution with $(\alpha, k - \ell)$ entropy subsets. Define the distribution $\mathbf{e}'$ as product of a uniform vector $\mathbf{e} \in \mathbb{F}_q^n$ and $\mathbf{W}$: that is, $e'_i = e_i \cdot_c \mathbf{w}_i$. Then the distribution $\mathbf{e}$ is a* MIPURS *distribution for $(k - \ell, \beta)$ for*

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

Proof. The proof follows the same basic structure as the proof of Lemma 1. Define $\mathbf{A}$ and $\mathbf{B}$ as above. Let ℓ be a free parameter. For some $\mathbf{b}$ in the span of $\mathbf{B}$ with weight at least $k - \ell$, consider the product $\langle \mathbf{b}, \mathbf{e}' \rangle = \sum_{i=1}^n \mathbf{b}_i \cdot \mathbf{e}_i$. 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 the inner product acts as a one time pad due to the inclusion of at least one coordinate of uniform vector $\mathbf{e}$. □

Linear Sources We now consider $(n-k+1)$-*linear sources*. Here we consider some matrix $\mathbf{E} \in \mathbb{F}_q^{n\times(n-k+1)}$ and define the distribution $\mathbf{w} = \mathbf{Es}$ for a uniformly random vector $\mathbf{s} \in \mathbb{F}_q^{n-k+1}$. Note the only condition we place on $\mathbf{E}$ is its dimension $(n-k+1)$. This distribution family follows from the intuition. If the nullspace of the code and the nullspace of the linear source have a low probability of overlapping then that is a secure distribution for our construction.

Lemma 3. *Let* $\mathbf{w}$ *be defined by a* $n-k+1$*-linear source. Then* $\mathbf{w}$ *is a* $(k,\beta)-$ MIPURS *distribution for*

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

Proof. Consider a random $\mathbf{A} \in \mathbb{F}_q^{n\times k}$ and let $\mathbf{B}$ represent its null space. $\mathbf{A}$ has full column rank with probability at least $1 - k/q^{n-k}$. Conditioned on this fact the null space contains q^{n-k} vectors. We bound the probability that there exists some nonzero vector $\mathbf{b} \in \mathtt{span}(\mathbf{B})$ where $\mathbf{b} \neq \mathbf{0}$ and $\mathbf{b} \in \mathtt{null}(\mathbf{E})$. For each of the q^{n-k} vectors $\mathbf{b}$ in $\mathbf{B}$ we designate an event determining if that vector is in a dimension $k-1$ space (the dimension of the null space of $\mathbf{E}$). We take a union bound over these events. Since the probability that a random vector in a dimension n space, falls within a dimension $k-1$ subspace is $q^{k-1}/q^n = q^{k-n-1}$ and we have q^{n-k} such events, we upper bound the probability of the null spaces having non-trivial intersection by $q^{n-k}/q^{k-n-1} = 1/q$.

Consider only $\mathbf{B}$ where $\mathbf{B} \cap \mathtt{null}(\mathbf{E}) = \{\mathbf{0}\}$. Then for all $\mathbf{b} \in \mathbf{B}$, $\mathbf{b} \neq \mathbf{0}$ it holds that $\mathbf{bE}$ is some nonzero vector and thus at least one uniform component of $\mathbf{s}$ contributes to the value of $\mathbf{bEs}$. □

3.2 Primary 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). Note that Lemma 3 above is tight. If we consider Linear sources of dimension $n-k$ they have high probability of having an intersection with the null space of $\mathbf{A}$ and this intersection is computable. So requiring min-entropy greater than q^{n-k} is necessary. Furthermore, for high enough correction capability, there is a distribution with min-entropy q^{n-k} that is wholly contained in a single Hamming ball where the ideal fuzzy extractor provides no security [FRS16].

The adversary in our setting is given a generating matrix of the code, $\mathbf{A}$, and is tasked with finding a linear combination that has the same outcome under the initial elements and the group operation. Our proof is in three parts. First consider some set of possible error vectors E of bounded size.

1. Theorem 3: We show that the number of vectors $\mathbf{e} \in E$ that have 0 inner product with any vector in the null space of $\mathbf{A}$ is small.

2. Theorem 4: We then build on this result showing if the adversary is allowed to choose the constant it predicts, it cannot increase the size of *predictable* error vectors too much. That is, for each constant g there are not many vectors $\mathbf{w}$ in the null space of $\mathbf{A}$ such that $\langle \mathbf{w}, \mathbf{e}\rangle = g$.

3. Lemma 6: We show that any distribution $\mathbf{W}$ with sufficient entropy cannot lie in the set of *predictable* error vectors E with high probability. This means that the first query of the adversary has high entropy. This is the MIPURS condition.

We note that these results do not depend on the generic group setting, they only ask the probability of predicting the result of some inner product from some high entropy distribution. As such, we do not use generic group notation in these proofs. For simplicity of statements we introduce $\kappa = n-k$ and

consider the null space $\mathbf{B} \in \mathbb{F}_q^{n \times \kappa}$. We codify the set of possible adversarial strategies in a definition we called κ-induced random variables. For the moment, we assume that $\mathbf{B}$ has dimension of κ and remove this restriction at the end of the proof (Corollary 3).

Definition 3. *Let* $\mathbf{b}$ *be a random variable taking values in* $\mathbb{F}_p^n$*. Let* $\mathbf{B}$ *be a 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* $\mathbf{b} \in \mathtt{span}(\mathbf{B})$ *with certainty:* $\Pr[\mathbf{b} \in \mathbf{B}] = 1$*. Note that unless* $\kappa = n$ *the random variables* $\mathbf{B}$ *and* $\mathbf{b}$ *must be dependent.*

The key concept we use is linear density. Since we don't know that the set of possible error values, denoted as E, acts linearly, we introduce linear density to measure how frequently a set overlaps with the best subspace of dimension ℓ:

Definition 4. *The* ℓ-linear density *of a set of vectors* $F = \{f_1, ..., f_n\}$ *in* $\mathbb{F}_p^n$ *is the maximum number of vectors that are covered by a subspace of a fixed dimension. Formally,*

$$\Delta^\ell(f_1, \ldots, f_m) = \max_{V, \dim(V)=\ell} |V \cap \{f_1, \ldots, f_m\}|.$$

We now show that if the set E is large enough there is no strategy for $\mathbf{b}$ that allows prediction with high probability if they have to obtain inner product 0. The next theorem (Thm. 4) will allow the adversary to predict an arbitrary inner product. For a κ induced random variable $\mathbf{b}$, first define

$$E_{\epsilon,\mathbf{b}} = \left\{ \mathbf{e} \in \mathbb{F}_p^n \mid \Pr_{\mathbf{b}}[\langle \mathbf{b}, \mathbf{e} \rangle = 0] \geq \epsilon \right\}.$$

and define $E_\epsilon = \max_{\mathbf{b}}(E_{\epsilon,\mathbf{b}})$ where the maximum is over all κ induced random variables in $\mathbb{F}_q^n$.

Theorem 3. *Let* q *be a prime and let* $d > 1$*,* $\kappa, m, \lambda, \ell \in \mathbb{Z}^+$ *be parameters for which* $\ell > \kappa > 0$*. Then if* $|E_\epsilon| > d \cdot q^\kappa$ *then*

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

Before proving Theorem 3, we introduce and prove two combinatorial lemmas (4 and 5). We then proceed with the proof of Theorem 3.

Lemma 4. *Let* q *be a prime, and let* $n \in \mathbb{Z}^+$*. Let* $d > 1$*,* $E \subset \mathbb{F}_q^n$ *where* $|E| = dq^\ell$ *and let* $F = \{f_1, \ldots, f_m\} \subset E$ *be uniformly and independently chosen in* E *(with replacement). Then for any* $\ell > 0$ *and* $\lambda \geq 0$

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

Proof. By definition of linear density, if $\Delta^\ell(F) \geq \ell + \lambda$ there must be at least one subset of $\ell + \lambda$ vectors of E contained in a subspace of dimension ℓ. Let $F' \subseteq F$ denote one such subset of $\lambda + \ell$ vectors. Towards bounding the probability of such an F' existing, the formation of F' can be thought of as follows: ℓ linearly independent vectors from F are found and a subspace $\mathbf{V}$ of dimension ℓ is formed from these linearly independent vectors. Then λ more unique vectors from F are found that are contained in $\mathbf{V}$.

We can count the probability of F' existing by first upper bounding the number of linearly independent subsets of ℓ vectors in F. We assume all subsets of ℓ vectors are linearly independent. There are $\binom{m}{\ell}$ subsets of F of ℓ unique vectors so we give this as the bound of the number of linearly independent subsets. Now, the probability that one of the remaining $m - \ell$ vectors in α falls within $\mathbf{V}$ is $|\mathbf{V}|/|E| = q^\ell / dq^\ell$. Each

element of F is uniformly and independently sampled, we upper bound the probability that λ vectors fall in $\mathbf{V}$ by

$$\binom{m-\ell}{\lambda}\left(\frac{q^\ell}{dq^\ell}\right)^\lambda = \binom{m-\ell}{\lambda} d^{-\lambda}.$$

We use this probability for each subspace to achieve the bound of the lemma. □

Lemma 5. *Let $\ell \in \mathbb{Z}^+$, let q be a prime, let $n \in \mathbb{Z}^+$, and let $\kappa < n$. Let $F = \{f_1, \ldots, f_m\} \subset \mathbb{F}_q^n$. Suppose that $|F| > \Delta^\ell(F)$. Then, for any κ-induced random variable $\mathbf{b}$ taking values in $\mathbb{F}_q^n$,*

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

Proof. For a set F, define $\beta \stackrel{def}{=} \Delta^\ell(F)$. If $\Delta^\ell(F) = \beta$ we know that there is at least one subset $F' \subset F$ where $|F'| = \beta$ where F' is contained by a subspace of dimension ℓ, call one such subspace $\mathbf{V}$. Suppose for the moment that there exists some vector $\mathbf{b}$ such that $\langle \mathbf{b}, f\rangle = 0$ for all $f \in F'$. By the definition of $\Delta(F)$ the dimension of $\texttt{span}(\{f | f \in F'\})$ must be ℓ otherwise one could add any arbitrary point F into F' contradicting the maximality of F'. One such point must exist otherwise $|F| = \Delta^\ell(F)$. Denote by $\mathbf{V}$ one such ℓ dimensional subspace noting that there are at most $\binom{m}{\ell}$ such subspaces. We now bound the probability (over $\mathbf{B}$) for a fixed $\mathbf{V}$ that $\mathbf{B} \cap \texttt{null}(\mathbf{V}) \neq \mathbf{0}$. Note if some κ-induced random variable $\mathbf{b}$ it is true that $\langle \mathbf{b}, f\rangle = 0$ for all $f \in F'$ it must be true that $\mathbf{b} \in \mathbf{B} \cap \texttt{null}(\mathbf{V})$. Since $\mathbf{B}$ is uniform in the space and has dimension κ the probability that $\mathbf{B} \cap \texttt{null}(\mathbf{V})$ is at most $q^\kappa q^{n-\ell}/q^n = q^{\kappa-\ell}$. We achieve the result by taking a union bound over sets of size ℓ representing possible sets of linearly independent vectors. □

Proof of theorem 3. Now we analyze a game in order to understand the relationship between our two paramenters of interest: ϵ and d. Fix some $\epsilon > 0$. In this game, we can imagine there being two stages of choice. The first is picking a set of m vectors uniformly from E_ϵ, call this set F, and the second is selecting a $\mathbf{b}$ vector from $\mathbf{B}$, a dimension κ subspace of $\mathbb{F}_q^n$. We study the expectation of the number of vectors in F that are orthogonal to the adversary's choice of $\mathbf{b}$. We first give a lower bound, by the linearity of expectation and the definition of E_ϵ:

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

We now seek to find an upper bound on this expectation using Lemmas 4 and 5. In the selection of the sample F from E_ϵ we classify samples as good and bad based on whether they satisfy the condition of Lemma 4. That is, we call a selection F bad if $\Delta^\ell(F) \geq \ell + \lambda$. For bad selections, we upperbound the expectation by m, for good selections we further split the expectation based on the selection of $\mathbf{B}$. We then call these samples great or terrible based on whether the condition of lemma 5. So a sample is terrible if for fixed $f_1, ..., f_m$ the selected $\mathbf{B}$ means there exists some $\mathbf{b}$ such that $|\{f_i \in F \mid \langle \mathbf{b}, f_i\rangle = 0\}| \geq \Delta^\ell(F)$. In the case of a terrible selection of $\mathbf{B}$, we upper bound the expectation by m again. Then if the experiment is neither bad or terrible, then we upper bound the expectation by $\ell + \lambda$ from our lemmas. So, for any $\ell > k$ and $\lambda > (m-\ell)/d$ we may apply the two lemmas above and find that

$$\mathop{\mathbb{E}}_{\mathbf{w},\mathbf{a}_i}[|\{\mathbf{a}_i \mid \langle \mathbf{a}_i, \mathbf{w}\rangle = 0\}|] \leq (\ell+\lambda) + m\left(\binom{m}{\ell}\binom{m-\ell}{\lambda}\left(\frac{1}{d}\right)^\lambda + \binom{m}{\ell} q^{k-\ell}\right)$$

and hence that

$$\epsilon \leq \binom{\ell+\lambda}{m} + \binom{m}{\ell}\left(\binom{m}{\lambda}\left(\frac{1}{d}\right)^{\lambda} + q^{k-\ell}\right).$$

□

Corollary 1. *Let κ and n be parameters satisfying $4 < \kappa < n$ and let q be a prime such that $q \geq (k+1)^{2(k+1)}$. Then for $\epsilon \geq 8eq^{-1/(2(k+2))}$ we have $|E_\epsilon| \leq 8eq^{\kappa}/\epsilon$.*

Proof. Set the parameters in Theorem 3 as:

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

First note that $\kappa + 1 = \ell \leq \lambda = \log q = \log 2 + (\kappa+2)\log d$ (as $q = d^{2(\kappa+2)}$). Then, consider a set E_ϵ. We have

$$\begin{aligned}\epsilon \leq \binom{\ell+\lambda}{m} + \binom{m}{\ell}\left(\left(\frac{me}{\lambda d}\right)^{\lambda} + q^{k-\ell}\right) &\leq \binom{2\lambda}{m} + 2\binom{m}{\ell}q^{-1} \\ &\leq \binom{4e}{d} + 2\binom{d\lambda/2e}{\ell}q^{-1} \\ &\leq \binom{4e}{d} + 2\left(\frac{d\lambda}{\ell}\right)^{\ell}q^{-1}.\end{aligned}$$

Since $q \geq \max(d^{2(\kappa+2)}, (k+1)^{2(k+1)})$ we have

$$\begin{aligned}2\left(\frac{d\lambda}{\ell}\right)^{\ell}q^{-1} \leq 2\frac{d^{\kappa+1}}{\sqrt{q}} \cdot \left(\frac{\log q}{\kappa+1}\right)^{\kappa+1}\frac{1}{\sqrt{q}} &\leq \frac{2}{d}\cdot\left(\frac{2(\kappa+1)\log(\kappa+1)}{\kappa+1}\right)^{\kappa+1}\frac{1}{\sqrt{q}} \\ &\leq \frac{2}{d}\cdot(2\log(\kappa+1))^{\kappa+1}\frac{1}{\sqrt{q}} \leq \frac{2}{d}\cdot(\kappa+1)^{\kappa+1}\frac{1}{\sqrt{q}} \leq \frac{2}{d} \leq \binom{4e}{d}\end{aligned}$$

where we have used the fact that $2\log(\kappa+1) \leq (\kappa+1)$ for $\kappa \geq 4$. We conclude that

$$\epsilon \leq \frac{8e}{d} \qquad \text{or, equivalently,} \qquad d \leq \frac{8e}{\epsilon}\,.$$

The largest ϵ for which we can apply this argument to yield the inequality is thus $8e/d$. □

Predicting Arbitrary Values We now show that the adversary cannot due much better than Theorem 3 by predicting any value $g \in \mathbb{F}_q$.

Theorem 4. *Let $\mathbf{b}$ be a κ-induced random variable in $\mathbb{F}_q^n$ and let g be a random variable over $\mathbb{F}_q$ (potentially correlated with $\mathbf{b}$). For $\epsilon > 0$ define*

$$E_\epsilon^* = \left\{\mathbf{e} \in \mathbb{F}_p^n \mid \Pr_{\mathbf{b},g}[\langle \mathbf{b}, \mathbf{e}\rangle = g] \geq \epsilon\right\}$$

Then $|E_{\epsilon^2/8}| \geq \frac{\epsilon^2}{8}|E_\epsilon^*|$, where E_ϵ is as defined in Theorem 3.

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

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

To see this, consider incrementally adding elements of E^*_ϵ into F' in so as to greedily increase

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

If this is process is carried out until no $\mathbf{e} \in E^*_\epsilon$ increases the total probability by more than δ, then it follows that every $F_\mathbf{e}$ 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.

Then it follows that for any $\mathbf{e} \in E^*_\epsilon$,

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

and hence

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

Then there exists an $\mathbf{f}^*$ for which

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

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

$$\mathop{\mathbb{E}}_{\mathbf{e} \in E^*_\epsilon} \Pr_{\mathbf{b}}[\langle \mathbf{b}, \mathbf{e} \rangle = \langle \mathbf{b}, \mathbf{f}^* \rangle] \geq \frac{\epsilon^2}{4}\,.$$

Using this expectation (of a probability), we bound the probability it is greater than $1/2$ its mean. We use the linear properties of inner product to achieve the homogenous case:

$$\Pr_{\mathbf{b}}\left[\Pr_{\mathbf{e} \in E^*_\epsilon}[\langle \mathbf{b}, \mathbf{e} - \mathbf{f}^* \rangle = 0] \geq \frac{\epsilon^2}{8}\right] \geq \frac{\epsilon^2}{8}$$

This translates directly to the claim of our theorem, $|E_{\epsilon^2/8}| \geq \frac{\epsilon^2}{8}|E^*_\epsilon|\,.$ □

Using Corollary 1, we have the following.

Corollary 2. *Let κ and n be parameters satisfying $4 < \kappa < n$ and let q be a prime such that $q \geq (k+1)^{2(k+1)}$. Then any $\epsilon \geq \dfrac{14}{q^{1/(4(k+2))}}$ it holds that*

$$|E^*_\epsilon| \leq \frac{8}{\epsilon^2}\frac{64eq^k}{\epsilon^2} = \frac{512eq^k}{\epsilon^4}.$$

Using standard techniques we can now argue that this implies all high min-entropy distributions are not predictable in the above game.

Lemma 6. *Let* $\mathbf{b}$ *be a* κ*-induced random variable in* $\mathbb{F}_q^n$*. Let* g *be an arbitrary random variable in* $\mathbb{F}_q$*. Let* W *be a random variable with* $\mathrm{H}_\infty(W) = s$*. Let* E_ϵ^* *be as defined in Theorem 4. Then for* $\epsilon > 0$

$$\Pr_{w \leftarrow W, \mathbf{b}, g}[\langle \mathbf{b}, w \rangle = g] \leq 2^{-s}|E_\epsilon^*| + \epsilon\,.$$

Proof. Our predictable set E_ϵ° gives us no guarantee on the instability of the inner product. If $a \in A_\epsilon^\circ$ then we upper bound the probability by 1. Because W has min-entropy s, we know that no element is select 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 w does not fall in the predictable set we bound the probability by ϵ (for simplicity, we ignore the multiplicative term less than 1). □

Corollary 3. *Let* k *and* n *be parameters satisfying* $4 < n - k < n$ *and let* q *be a prime such that* $q \geq (n-k+1)^{2(n-k+1)}$*. Let* $\epsilon \geq \frac{14}{q^{1/(4(2+n-k))}}$ *be a free parameter. Then for all distributions* $\mathbf{W} \in \mathbb{F}_q^n$ *such that* $\mathrm{H}_\infty(\mathbf{W}) \geq \log(512eq^{n-k}\epsilon^{-5})$*, it holds that then* $\Pr_{\mathbf{b},g,\mathbf{w}}[\langle \mathbf{b}, \mathbf{w} \rangle = g] \leq 2\epsilon + k/q^{n-k}$ *and thus* $\mathbf{W}$ *is* $(k, 2\epsilon + k/q^{n-k}) - \mathsf{MIPURS}$.

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

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]).

Definition 5. *Let* $\mathcal{W}$ *be a family of probability distributions over* $\mathcal{M}$*. 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* $(\mathcal{M}, \mathcal{W}, \kappa, t)$-computational fuzzy extractor *that is* (ϵ_{sec}, m)*-hard with error* δ *if* Gen *and* Rep *satisfy the following properties:*

- Correctness*: if* $\mathsf{dis}(\mathbf{w}, \mathbf{w}') \leq t$ *and* $(\mathsf{key}, \mathsf{pub}) \leftarrow \mathsf{Gen}(\mathbf{w})$*, then* $\Pr[\mathsf{Rep}(\mathbf{w}', \mathsf{pub}) = r] \geq 1 - \delta$.
- Security*: for any distribution* $W \in \mathcal{W}$*, the string* key *is close to random conditioned on* pub *for all* $\mathcal{A}$ *making at most* m *queries to the group oracle* σ*, that is*

$$\Pr_{\sigma \xleftarrow{\$} \Sigma}[\mathcal{A}^\sigma(\mathsf{Key}, \mathsf{Pub}) = 1] - \Pr[\mathcal{A}^\sigma(U, \mathsf{Pub}) = 1] \leq \epsilon_{sec}.$$

In the above, group elements in $\mathsf{Key}, U, \mathsf{Pub}$ are represented by group handles, the adversary additionally receives $\sigma(1)$. The errors are chosen before Pub: if the error pattern between $\mathbf{w}$ and $\mathbf{w}'$ depends on the output of Gen, then there is no guarantee about the probability of correctness.

Construction 1. *Let* λ *be a security parameter,* t *be a distance,* $k = \omega(\log \lambda)$*,* $\alpha \in \mathbb{Z}^+$*,* $\ell \in \mathbb{Z}^+$*, 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. Let* $\mathcal{W} \in \mathbb{F}_q^n$*, and let* dis *be the Hamming metric. Define* $(\mathsf{Gen}, \mathsf{Rep})$ *as follows:*

Gen $(w = w_1, ..., w_n)$

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. *For* $i = 1, ..., n$: *set* $\mathsf{c}_i = r^{\mathbf{A}_i \cdot \mathbf{x} + w_i}$.
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 $(w', \mathsf{pub} = (r, \mathbf{A}, \mathsf{c}_1 \dots \mathsf{c}_n))$

1. *For* $i = 1, ..., n$, *set* $\mathsf{c}_i = \mathsf{c}_i / r^{w_i}$.
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* $\perp$.

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 $w^1, ..., w^\rho$ 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 $\mathsf{pub}^1, \dots, \mathsf{pub}^\rho$ (the reproduction procedure Rep at the ith provider still obtains only a single w' close to w^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 6 (Reusable Fuzzy Extractor [CFP+16])**.** *Let* $\mathcal{W}$ *be a family of distributions over* $\mathcal{M}$. *Let* $(\mathsf{Gen}, \mathsf{Rep})$ *be a* $(\mathcal{M}, \mathcal{W}, \kappa, t)$*-computational fuzzy extractor that is* (ϵ_{sec}, m)*-hard with error* δ. *Let* $(W^1, W^2, \dots, W^\rho)$ *be* ρ *correlated random variables such that each* $W^j \in \mathcal{W}$. *Let* D *be an adversary. Define the following game for all* $j = 1, ..., \rho$*:*

- ***Sampling*** *The challenger samples* $w^j \leftarrow W^j$ *and* $u \leftarrow \mathbb{G}_q^\alpha$.
- ***Generation*** *The challenger computes* $(\mathsf{key}^j, \mathsf{pub}^j) \leftarrow \mathsf{Gen}(w^j)$.
- ***Distinguishing*** *The advantage of* D *is*

$$\begin{aligned} Adv(D) \stackrel{def}{=} &\Pr[D(\mathsf{key}^1, ..., \mathsf{key}^{j-1}, \mathsf{key}^j, \mathsf{key}^{j+1}, ..., \mathsf{key}^\rho, \mathsf{pub}^1, ..., \mathsf{pub}^\rho) = 1] \\ &- \Pr[D(\mathsf{key}^1, ..., \mathsf{key}^{j-1}, u, \mathsf{key}^{j+1}, ..., \mathsf{key}^\rho, \mathsf{pub}^1, ..., \mathsf{pub}^\rho) = 1]. \end{aligned}$$

$(\mathsf{Gen}, \mathsf{Rep})$ *is* $(\rho, \epsilon_{sec}, m)$*-reusable if for all* D *making at most* m *queries and all* $j = 1, ..., \rho$, *the advantage is at most* ϵ_{sec}. *In our theorems we assume that* D *is provided with handles to all group elements and the generic group oracle.*

Theorem 5. *Let all parameters be as in Construction 1. Let* $W^1, ..., W^\rho \in \mathbb{F}_q^n$ *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*

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

Proof. Our generic proof shows that an adversary that knows $\mathbf{A}$ is unable to distinguish between $\mathbf{Ax}+\mathbf{e}$ from $\mathbf{U}$ except with negligible probability. Without loss of generality, we assume that the adversary is trying to learn information about the first key. For the construction to be reusable for all distinguishers, it must be true that:

$$\begin{aligned}&|\Pr[\mathcal{D}(\mathbf{U}, r_1, \mathbf{A}_1, \mathbf{A}_1\mathbf{x}_1+\mathbf{e}_1, \{\mathsf{key}_i, \mathsf{pub}_i\}_{i=2}^{\rho}) = 1]\\ &-\Pr[\mathcal{D}(r^{\mathbf{x}_{0..\alpha-1}}, r_1, \mathbf{A}_1, \mathbf{A}_1\mathbf{x}_1+\mathbf{e}_1, \{\mathsf{key}_i, \mathsf{pub}_i\}_{i=2}^{\rho}) = 1| \leq \epsilon_{sec}.\end{aligned}$$

Crucially, in Theorem 2, we assume that handles are in a sufficient sparse space such that handles from one oracle never represent a valid handle for another oracle. Rather than initializing a joint oracle to answer all queries, one can separately initialize oracles for each application of the fuzzy extractor. This is because each application of the fuzzy extractor works for a different group generator. Then the $\rho-1$ oracles corresponding to other enrollments w_i are the same in both settings. Using a simple hybrid argument on Theorem 2 we can replace these oracles with uniform values. Once replaced by uniform values these oracles provide no information to the adversary. The theorem follows by a final application of Theorem 2. □

Handling binary W As shown in Lemma 2 when W is binary and subsets of W are hard to predict one can form a MIPURS distribution by multiplying by an auxiliary random and uniform random variable $e \in \mathbb{F}_q^n$. This has the effect of placing random errors in the locations where $W_i = 1$. 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 w are 1, denoted $w_j = 1$ we cannot use location j for decoding as it is a random value (even if $w'_j = 1$ as well). When one amplifies a binary w, we recommending using another uniform random variable $y \in \{0,1\}^n$ and check when $y_i \neq w_i$ to indicate when to include a random error. Then in reproduction the algorithm should restrict to locations where $y_i = w_i$. 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).

We analyze the correctness and efficiency of this more complicated construction in Appendix B and show correctness and efficiency. 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.

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 7 ([CFP+16] Sources with High Entropy Samples)**.** *Let the source $W = W_1, \ldots, W_n$ consist of strings of length n over some arbitrary alphabet $\mathcal{Z}$. We say that the source W is a source with a*

(k, β)-***entropy-samples*** *if*

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

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

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 2. *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 $\mathbf{A} \in (\mathbb{F}_q)^{2n \times n}$, $x_0 = 0, x_{1,\ldots,n-1} \leftarrow (\mathbb{F}_q)^{n-1}$.*
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$(g^{\mathbf{y}}, \mathbf{A}, x \in \{0,1\}^n)$*:*

1. *Define $\mathcal{I}$ as $\{i \in [1...2n] \mid x_{\lfloor i/2 \rfloor} = (i \mod 2)\}$.*
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.

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

Lastly, define the distribution $e' = e \cdot_c \mathsf{Bin}(v)_i$.

Theorem 6. *Let $\ell \in \mathbb{Z}^+$ be a free parameter. Define $\mathcal{V}$ as the set of all distributions V such that $E' = U^n_{\mathbb{F}_q} \cdot_c \mathsf{Bin}(V)$ is a distribution that is $(n, \beta) - \mathsf{MIPURS}$. Then Construction 2 is VBB secure for generic $\mathcal{D}$ making at most m queries with distinguishing probability at most*

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

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 answer, "For what distributions can we prove security of the code offset construction in the standard model only assuming the hardness of discrete log?" We use this as a comparison to the distributions we can prove secure in the generic group model. We examine hardness of decoding random linear codes in the exponent. In Appendix C we consider Reed-Solomon codes. Both results follow a three part outline:

1. A theorem of Brands [Bra93] which says that if given a uniformly distributed $g^{\mathbf{y}}$ one 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 $\mathsf{dis}(g^{\mathbf{x}}, g^{\mathbf{z}}) \leq c$ (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 $\mathsf{dis}(g^{\mathbf{z}}, g^{\mathbf{y}}) \leq c$ 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 λ that is only nonzero in positions where $g^{\mathbf{z}_i} = g^{\mathbf{y}_i}$ and $\langle \lambda, \mathbf{x} \rangle = \langle \lambda, \mathbf{y} \rangle = 0$. If $\mathcal{O}$ works on a random point $g^{\mathbf{y}}$ it is possible to compute a vector λ 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 supplemental material (Appendix C), we present a similar result for Reed-Solomon codes improving prior work of Peikert [Pei06].

An (n, k, q)- random linear code, denoted $\mathbb{RL}(n, k, q)$, is generated by a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times k}$ that is independent and uniform elements of $\mathbb{Z}_q$. The code is the set of $\mathbf{Ax}$ for all vectors $\mathbf{x} \in \mathbb{F}_q^k$. 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 $\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{e}$.

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 $\mathsf{dis}(\mathbf{y}, \mathbf{z}) \leq t$, 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 t and a point $\mathbf{y} \in \mathbb{Z}_q^n$ a bounded distance decoding algorithm returns some $\mathbf{z} \in \mathbb{C}$ such that $\mathsf{dis}(\mathbf{y}, \mathbf{z}) \leq t$. There is no guarantee that $\mathbf{z}$ is unique or is the point in the code closest to $\mathbf{y}$.

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

Instance Known generator g of $\mathbb{Z}_q^*$. Define $\mathbf{e}$ as a random vector of weight c in $\mathbb{Z}_q$. Define $g^{\mathbf{y}} = g^{\mathbf{A}\mathbf{x}+\mathbf{e}}$ 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 $\mathbf{z} = \mathbf{A}\mathbf{x}$ 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, $\mathsf{dis}(\mathbf{y}, \mathbf{C}) = \min_{\mathbf{c} \in \mathbf{C}} \mathsf{dis}(\mathbf{y}, \mathbf{c})$.

Our proofs use the notion of *thickness* of a point with respect to a codespace and a radius. 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 variables $\varphi(r, \mathbf{z}, \mathbf{y})$ 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 $\mathsf{Thick}(r, \mathbf{C}, \mathbf{y}) = \sum_{\mathbf{z} \in \mathbf{C}} \varphi(r, \mathbf{z}, \mathbf{y})$.

We now present the theorem of this section and our key technical lemma (Lemma 7), then prove the lemma and finally the theorem.

Theorem 7. *For positive integers n, k, c and q where $k < n \leq q$ and let g be a generator of $\mathbb{Z}_q^*$. If an efficient algorithm exists to solve $\mathtt{BDDE} - \mathtt{RL}(n, k, q, n-k-c, g)$ 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 $\mathsf{Vol}(n, r, q) \geq \binom{n}{k} q^r (1 - n/q)$, we get

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

Lemma 7. *Let a Code $\mathbb{RL}_{\mathbf{A}}(n, k, q)$ be defined by matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times k}$, then*

$$\Pr_{\mathbf{y} \in \mathbb{F}_q^n, \mathbf{A}}[\mathsf{dis}(\mathbf{y}, \mathbb{RL}_{\mathbf{A}}(n, k, q)) > n - k - c] \leq \frac{q^{n-k}}{\mathsf{Vol}(n, n-k-c, q)} + \frac{1}{q^{n-k}}.$$

Proof of Lemma 7. A Random Linear Code $\mathbb{RL}_{\mathbf{A}}(n, k, q)$ 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 be 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

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

□

Proof of Theorem 7. Suppose an algorithm $\mathcal{F}$ solves $\mathtt{BDDE} - \mathtt{RL}(n, k, q, n-k-c, g)$ with probability ϵ. We show that $\mathcal{F}$ 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 λ where $\lambda_i = \mathbf{B}_{i'}$ 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 7, $(\mathbf{y}, \mathbf{A})$ is a uniform instance of $\mathtt{BDDE} - \mathtt{RL}(n, k, q, n-k-c, g)$ with probability at least $1 - (q^{n-k}/\mathsf{Vol}(n, n-k-c, q) + k * q^{-(n-k)})$. 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 are grateful to The authors give special thanks to reviewer comments and feedback. The authors thank James Bartusek, Fermi Ma, and Mark Zhandry and their helpful discussions of their work. The work of Benjamin Fuller is funded in part by NSF Grant No. 1849904. This material is based upon work supported by the National Science Foundation under Grant No. 1801487.

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.

[BBR88] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[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, Tancrde 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.

[FRS16] Benjamin Fuller, Leonid Reyzin, and Adam Smith. When are fuzzy extractors possible? In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 277–306. Springer, 2016.

[GGH11] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 30–39. Springer, 2011.

[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.

[JS06] Ari Juels and Madhu Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38:237–257, 2006.

[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.

[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.

[Reg10a] 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.

[Reg10b] Oded Regev. The learning with errors problem (invited survey). *Annual IEEE Conference on Computational Complexity*, 0:191–204, 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.

[SSF18] Sailesh Simhadri, James Steel, and Benjamin Fuller. Reusable authentication from the iris. 2018.

[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.

[Wyn75] Aaron D Wyner. The wire-tap channel. *Bell system technical journal*, 54(8):1355–1387, 1975.

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 A. This proof is a relatively straightforward generic group proof. 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 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 $\{\sigma(x_i)\}_i$. 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 $\sigma(x + y)$.*

$\mathcal{A}$ 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 that could be used to initialize the implementation of 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 8 ([BKM+18] Lemma 7)**.** *Suppose there exists an algorithm $\mathcal{A}$ such that*

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

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 $\mathcal{G}_\mathcal{M}$. $\mathcal{A}^{\mathcal{G}_\mathcal{S}}(\mathcal{O}^{\mathcal{G}_\mathcal{S}})$ 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 $\mathcal{X} = (c_0, \ldots, c_i)$ 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 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 $\mathcal{O}_c$. This oracle picks a message $\mathbf{x}$, uses the generating matrix $\mathbf{A}$ and the error vector random variable $\mathbf{e}$ which is a $(k, \beta) - \mathsf{MIPURS}$ distribution.

The oracle begins by calculating the noisy codeword $\mathbf{b}_1, ..., \mathbf{b}_n$ as $\mathbf{b} = \mathbf{Ax}+\mathbf{e}$. 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 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 9. *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*

$$|\Pr[\mathcal{D}(\mathcal{O}_c) = 1] - \Pr[\mathcal{D}(\mathcal{O}^*) = 1]| \leq \gamma \left(\frac{2}{q} + \beta\right)$$

for $\gamma = ((m+n+2)(m+n+1))^2/8$.

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 $\lambda||\chi_1, ..., \chi_n$. 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 $\mathbf{A}$):

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

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,

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

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

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

Case 2. Decomposing the linear combination of the codeword into $\chi(\mathbf{Ax}+\mathbf{e})$ since χ is in the null space of A then our 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 $\langle\chi,e\rangle+\lambda$) 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 9 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. To create an oracle analogous to a uniform distribution as claimed in Theorem 2. Note that this oracle is different $\mathcal{O}_r$ 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 9, we can prove Theorem 2 using Lemma 8 (and the modification to the uniform oracle) where

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

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 Correctness of Fuzzy Extractor

In this section we present the more complex fuzzy extractor designed to handle binary w inputs (using two auxiliary variables). Correctness in this case is more complicated, we analyze correctness of this construction in place of Construction 1.

Construction 3. *Let λ be a security parameter, t be a distance, $k = \omega(\log \lambda)$, $\alpha \in \mathbb{Z}^+$, 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 $\mathcal{W} \in \{0,1\}^n$ and let dis be the Hamming metric. Let $\tau = \max(0.01, t/n)$. Define $(\mathsf{Gen}, \mathsf{Rep})$ as follows:*

$\mathsf{Gen}(w = w_1, ..., w_n)$

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 $w_i = y_i$, 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 $(\mathsf{key}, \mathsf{pub})$.*

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

1. *Let $\mathcal{I} = \{i | w'_i = y_i\}$.*
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 go to 4.*
   - *(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 constant $k' \stackrel{def}{=} k + \alpha = \Theta(n)$ 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 $k' \leq \frac{(1-\Theta(1))}{3} * n$. We use the q-ary entropy function which is a generalization of the binary entropy function to larger fields. $H_q(x)$ 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 8. *Let parameters be as in Construction 1. 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 8. We assume a fixed number of iterations in Rep denoted by ℓ. Recall we assume that $\mathsf{dis}(w, w') \leq t$ and that the value $\mathbf{y}$ is independent of both values (by Def 5, w' does not depend on the public value). We first consider the final check of whether $\mathsf{dis}(\mathsf{c}_{\mathcal{I}}, \mathsf{c}'_{\mathcal{I}}) \leq |\mathsf{c}_{\mathcal{I}}|(1 - 2\tau)$ will return correctly if and only $r^{\mathbf{x}} = r^{\mathbf{A}_{J_i}^{-1} \mathsf{c}_{J_i}}$. We stress that this is property is independent of the chosen subset and only depends on $\mathbf{A}, \mathbf{x}, w, w'$ and 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 w, w' such that $\mathsf{dis}(w, w') \leq t$ and define the random variable Z of length n where a bit i of z that indicates when $w_i = w'_i$ and when $w'_i = y_i$. We consider the setting when $t = \Theta(n)$, if $t = o(n)$ then $\tau \stackrel{def}{=} t/n \leq .01$ and the condition holds with high probability. Define $S = \{i | w_i = y_i = w'_i\}$. 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 10. *[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 10 completes the proof of Theorem 8. □

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 $w'_{\mathcal{J}_i} = w_{\mathcal{J}_i}$ is at least

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

This follows since $d(w_{\mathcal{I}}, w'_{\mathcal{I}}) \leq 2\tau * |\mathcal{I}|$ 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 $e^x \approx 1 + x$ to get

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

Suppose that correctness $1 - \delta \geq 1 - (\delta' + e^{-\Theta(n)})$ is desired. (Here, the $e^{-\Theta(n)}$ 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:

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

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

C 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 $\mathbf{Vx}$ 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 $\mathbf{Vx}$ 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 $g^{\mathbf{y}} = g^{\mathbf{Vx}+\mathbf{e}}$ 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 $\mathsf{dis}(g^{\mathbf{y}}, g^{\mathbf{z}}) \leq c$.

Theorem 9. *For any positive integers n, k, c, and q such that $q \geq n^2/4$, $c \leq n+k$, $k \leq n$ and a generator g of the group $\mathcal{G}$, 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 $\mathcal{G}$ with probability*

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

Proof. Like Theorem 7 the core of our theorem is a bound on the probability that a random point is close to a Reed-Solomon code.

Lemma 11. *For any positive integer $c \leq n - k$, define $\alpha = \frac{4q}{n^2}$, and any Reed-Solomon Code $\mathbb{RS}(n, k, q)$,*

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

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

Proof of Lemma 11. 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,

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

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,

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

We are now ready to proceed to the proof.

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 $n^k \geq \binom{n}{k}$ to arrive at the bound that

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

The proof then follows using our bound for variance by defining $\alpha = 4q/n^2$. This completes the proof of Lemma 11. □

The remainder of the proof is similar to the proof of Theorem 7. $\mathcal{A}$ works as follows: on input $\mathbf{y}$ where $\mathbf{y}$ is uniform over G^n immediately run $\mathcal{D}(g, \mathbf{y})$. By Lemma 11, $(g, \mathbf{v})$ is an instance of $\mathtt{BDDE} - \mathtt{RS}_{q,\mathcal{E},k,n-k-c}$ 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 $\mathsf{dis}(\mathbf{z}, \mathbf{y}) \leq n - k - c$. Take any $k+1$ indices $\mathcal{I} \subseteq [n]$ such that $\mathbf{y}_i = \mathbf{z}_i$ for $i \in E$. Then any k of the $\mathbf{y}_i$ interpolate to another one of the $\mathbf{y}_i$. We find the non-trivial Lagrange coefficients for the first k $\mathbf{y}_i$ call them λ_i such that $\prod_{i \in E} v_i^{\lambda_i} = 1$. Call the remaining point $\mathbf{y}_{k+1}$. let $\lambda_i = 0$ for $i \notin E$ and set λ_{k+1} to -1. Then $(\lambda_1, \ldots, \lambda_n)$ is a solution to $\mathtt{FIND} - \mathtt{REP}$. 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 9 follow in the case when $\alpha = 4q/n^2 \geq 2$ by noting that:

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

□