Hardness of Decoding Random Linear Codes in the Exponent

Luke Demarest* Benjamin Fuller† Alexander Russell‡

June 7, 2019

Abstract

The hardness of decoding random linear codes with errors is a complexity-theoretic assumption with broad applications to cryptography. In contrast, Reed-Solomon codes permit efficient decoding in many representations. Despite this, a result of Peikert (TCC 2006) proves that in groups where discrete log is hard, Reed-Solomon error correction is difficult if symbols are written in the exponent. We bring these two lines of work together, examining hardness of decoding random linear codes in the exponent.

Our main result is a pair of theorems that show hardness of decoding random linear codes in both the generic group model and the standard model. These results hold for different classes of distributions for errors. In the generic group model, security holds if all vectors in the dual of the random linear code have an unpredictable inner product with the error distribution. While this condition is technical, it is satisfied by:

1. Any distribution where symbols are independent and contribute a constant amount of entropy. This requirement is met by the discretized Gaussian (Regev, STOC 2005), uniform interval (Döttling and Müller-Quade, Eurocrypt 2013), and uniform bit error (Micciancio and Peikert, Crypto 2013). Thus, most variants of learning with errors are hard in the generic group model.
2. 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 (whose size is the nullity of the code) to have no errors.
3. Any distribution formed by a linear process as long as the dimension of the process is greater than the nullity of the linear code.

Our results in the standard model show hardness of decoding random linear codes with a uniform input point. This result improves on a result of Peikert (TCC 2006) who considered the problem for Reed-Solomon codes.

We apply these results to reusable fuzzy extractors and pattern matching obfuscation, improving parameters on recent state of the art constructions.

Keywords Learning with errors; error-correction; generic group model; fuzzy extractors; pattern-matching obfuscation

*Email: luke.johnson@uconn.edu. University of Connecticut.

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

‡Email: acr@uconn.edu University of Connecticut.

1 Introduction

Human secrets are noisy and nonuniform. In contrast, cryptographic schemes prefer uniform inputs and exact reproduction of values. This gap is illustrated when a user attempts to match a current noisy value against a nonuniform secret value. This occurs in biometric authentication using fuzzy extractors [DORS08] and pattern matching obfuscation [BKM+18]. Current constructions that secure all viable distributions are inefficient, relying on either multilinear maps [BCKP17] or unbounded runtime/space [FRS16, WCD+17].

To close this gap, we study the problem of decoding random linear codes in the exponent. We use this tool to provide constructions in both applications for nonuniform, noisy distributions (we review these applications in detail in Section 1.2 after introducing our technical results).

Our approach has two direct motivations: (i.) the flexibility and broad applicability of decoding random codes to cryptography (as codified by the learning with errors (LWE) assumption [Reg05]) and (ii.) the striking fact that decoding Reed-Solomon codes is difficult in the exponent even though efficient algorithms exist when symbols are in the clear [Pei06]. This suggests that the problem of decoding random linear codes in the exponent may be a useful cryptographic tool.

Hardness of Decoding Random Linear Codes. In the last twenty years, the hardness of decoding error correcting codes has become a core cryptographic assumption yielding almost every cryptographic primitive [BMvT78, Reg05, BV14, GVW15, WZ17]. The relevant cryptographic assumption is called learning with errors or LWE [Reg10]. The decision version of LWE asks an adversary to distinguish an encoded codeword with errors $\mathbf{Ax} + \mathbf{e}$ from a uniformly selected vector, where $\mathbf{x}$ describes a message, $\mathbf{A}$ is a random public matrix, and $\mathbf{e}$ is an error term. Here all operations are performed in the finite field $\mathbb{F}_q$. For particular distributions of $\mathbf{e}$ it is possible to show that any adversary that solves LWE implies an adversary that can approximately solve worst case lattice problems (`GapSVP` [BLP+13] or `SIVP` [Reg05]). While there is some flexibility in the distribution of $\mathbf{e}$ (e.g., a discretized Gaussian [Reg05], a uniform interval [DMQ13], or a binomial distribution [MP13]), known reductions require each symbol of $\mathbf{e}$ to be independently distributed.

Structured Codes in the Exponent. Structured codes also have far-reaching applications in cryptography. An important motivating example arises in threshold public key encryption [Des92]. Using a length n and dimension k Reed-Solomon code and El Gamal encryption [ElG85] one can exploit the linear structure of El-Gamal to distribute shares of private keys to n parties such that any k parties can decrypt messages without revealing their share of the secret key.

This scheme is private against adversaries that know fewer than k shares and is correct against adversaries that keep at most $n - k$ parties from participating in reconstruction. However, it is also possible to correct errors for Reed-Solomon codes: indeed, the Berlekamp-Welch [WB86] algorithm can efficiently correct up to $t = (n - k + 1)/2$ errors. Suppose there is an adversary that intentionally submits incorrect shares during reconstruction. A natural question is whether Reed-Solomon error-correction can be applied to ensure correctness against this adversary. As it happens, the Berlekamp-Welch decoding algorithm critically relies on nonlinear operations, so its not clear how to execute this algorithm "in the exponent." Peikert showed this is, in some sense, unavoidable with two results [Pei06, Theorems 3.1 and 4.1]:

1. If balls of radius t around codewords cover a noticeable fraction of the space, an algorithm that corrects t errors implies a solver for the discrete logarithm problem. Importantly, this algorithm

only has to find some codeword within distance t (and not, necessarily, the closest codeword). This is known as bounded distance decoding [CW07], in contrast to the more traditional unique decoding. In order for balls around codewords to cover a noticeable fraction of the space, one needs to consider an algorithm that corrects almost $t = n-k$ errors. Note for Reed Solomon codes performing bounded distance decoding in the exponent for distance $n - k$ is easy: One picks k points and interpolates the rest of the points which can be done linearly (and in the exponent). We stress that this result does not rule out algorithms which correct fewer errors.

2. In the generic group model [Sho97], adversaries are limited to algebraic manipulations with linear combinations of group elements. In particular, the adversary's algorithm must work for an arbitrary representation of the group. In this model, Peikert showed that given $g^{\mathbf{c}+\mathbf{e}}$, where $\mathbf{c}$ is a codeword and $\mathbf{e}$ is a random vector with t nonzero values, it is hard to find $g^{\mathbf{c}}$. In this model, no assumption is necessary about the density of codewords. The main requirements for this theorem are (i.) that t errors are uniformly distributed and (ii.) that the product of the dimension and number of errors is large enough (namely, that $tk = \omega(n \log n)$).

Peikert's work presented the above as negative results. However, these results can be seen as introducing a new type of hardness assumption. The goal of this work is to combine the above two lines of work and ask:

> Does placing a random linear code in the exponent of a group amplify the hardness of decoding?

1.1 Hardness of Decoding Random Linear Codes in the Exponent

In the generic group model, we establish that decoding random linear codes in the exponent is hard for a broad class of distributions which we call $(k, \beta) - \mathsf{MIPURS}$ or *maximum inner product unpredictable over random subspace* distributions (Theorem 1). Specifically, an error distribution $\mathbf{e}$ (taking values in $\mathbb{F}_q^n$) is $(k, \beta) - \mathsf{MIPURS}$ if—with high probability in selection of a random subspace $W \subset \mathbb{F}_q^n$ of dimension k—every nonzero $\mathbf{w} \in W$ has the property that

$$\forall z, \Pr[\langle \mathbf{w}, \mathbf{e} \rangle) = z] \leq \beta.$$

This is formally defined in Definition 3. This definition is useful because it encompasses error distributions of special interest:

1. **Independent** Any distribution where symbols are independent and contribute a constant amount of entropy including the discretized Gaussian [Reg05], uniform interval [DMQ13], and uniform bit error [MP13]. It follows that most previously considered variants of LWE are hard in the generic group model. These results all hold for an arbitrary polynomial number of samples. Interestingly, due to an attack by Arora and Ge, the uniform bit error is distinguishable in polynomial time when $n = \Theta(|x|^2)$ [AG11, MP13] so MIPURS is not sufficient if the adversary can perform nonlinear operations.

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]. Peikert's result holds if the product of errors and the size of the

message is large enough (when $tk = \omega(n \log n)$). (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 $tk = \Theta(n \log n)$.) 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 (Lemma 4).

3. **Linear** Distributions formed by linear processes whose dimension is greater than ℓ.

The quantitative relationships between these distributions and the MIPURS condition are established in Section 3.1. Our main result is that MIPURS with ℓ equal to the nullity of the random linear code is sufficient for hardness of distinguishing the noisy codeword and a uniform value in the generic group model (Theorem 1). Furthermore, if the error distribution is not MIPURS there is an information-theoretic distinguisher. Thus, the MIPURS condition is necessary and sufficient against information theoretic adversaries in the generic group model. The only non-efficient part of the adversary is mapping a subspace W to the vector w and point z at which they can predict.

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. We formally define the game in Section 2.

Standard Model Results. We show hardness of decoding random linear codes in the standard model (assuming only hardness of discrete log), however these results require the error $\mathbf{e}$ to have independent symbols, with $\mathbf{e}$ possessing t randomly chosen nonzero positions. We show this result for both random linear codes (Theorem 4) and Reed-Solomon codes (Theorem 6). Both results provide a small improvement of parameters over Peikert's result [Pei06, Theorem 3.1]. As stated, these arguments require that a random point lies close to a codeword with noticeable probability. As q increases this probability decreases but discrete log becomes harder, creating a tension between these parameters. Peikert's result requires that $q \leq \binom{n}{k+1}/n^2$. In our application to the fuzzy extractors we consider small k for which $k = \omega(\log \lambda)$. 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 4 and 6 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 gap is necessary?

1.2 Applications

Fuzzy Extractors. A fuzzy extractor derives stable keys from a noisy source of entropy [DORS08]. Formally, a fuzzy extractor is a pair of algorithms: Generate or $\mathsf{Gen}(\mathbf{w})$ takes an initial reading of a noisy source $\mathbf{w}$ and outputs a cryptographic key key and a public value pub; Reproduce or $\mathsf{Rep}(\mathbf{w}', \mathsf{pub})$ takes a subsequent reading $\mathbf{w}'$ and outputs key if $\mathbf{w}$ and $\mathbf{w}'$ are within distance t. The security guarantee is that key should be pseudorandom conditioned on pub. Fuzzy extractors secure distributions from nature such as biometric and physical uncloneable functions, so it is prudent to minimize assumptions about these distributions. For example, the Iriscode transform is estimated to have 249 bits of entropy out of a 2048 bit string [Dau04] (see discussion in [FRS16, SSF18]). Importantly bits of $\mathbf{w}$ are *correlated*.

Fuller et al. [FMR13] proposed a fuzzy extractor where the reading $\mathbf{w}$ served as the error vector for an LWE instance, that is $\mathsf{pub} = \mathbf{Ax} + \mathbf{w}$.[1] The Rep algorithm performs "guess and check" on subsets $\mathcal{I}_j \subseteq \{1, ..., |\mathbf{w}|\}$ decoded as

$$\mathbf{x} = \mathbf{A}_{\mathcal{I}_j}^{-1}\mathsf{pub}_{\mathcal{I}_j} - \mathbf{w}'_{\mathcal{I}_j} = \mathbf{A}_{\mathcal{I}_j}^{-1}(\mathbf{Ax} + \mathbf{w} - \mathbf{w}')_{\mathcal{I}_j}.$$

This decoding succeeds if $\mathbf{w}_{\mathcal{I}_j} = \mathbf{w}'_{\mathcal{I}_j}$. To achieve error tolerance, multiple independent sets $\mathcal{I}_j$ are sampled and checked. Importantly, this decoding algorithm selects a subset and is then entirely linear. To ensure hardness of the underlying lattice problem, the construction required: (i.) the dimension of $\mathbf{x}$ to be a constant fraction the length of w and (ii.) for $\mathbf{w}$ to be a distribution for which LWE is hard. This limited error tolerance to at most $t = O(\log |w|)$.

Canetti et al. [CFP+16] presented a fuzzy extractor that explicitly placed specific subsets in a digital locker [CD08]. Digital lockers can be constructed using exponentiation in a Diffie-Hellman group [BC10]. This construction sampled multiple subsets $w_{\mathcal{I}_j}$ and locked the same key in a digital locker secured with $w_{\mathcal{I}_j}$. Roughly, $\mathsf{pub} = \{r_j, r_j^{w_{\mathcal{I}_j}} \cdot \mathsf{key}, \mathcal{I}_j\}_j$. Decoding consisted of trying to open each digital locker. Digital lockers improved allowable error tolerance to $t = o(|w|)$. However, 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]. Canetti et al.'s construction is also *reusable*, allowing correlated versions of w to be used to derive multiple keys.

We introduce a new fuzzy extractor that places a random linear code in the exponent (where r is a random generator):

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

(The second component is a vector of group elements.) Decoding proceeds as in the previous constructions, randomly selecting subsets $\mathcal{I}_j$ and hoping they have no errors. This construction is secure if w is drawn from a `MIPURS` distribution. If the construction is instantiated with a random linear code of small dimension $|\mathbf{x}| = \omega(\log \lambda)$, error tolerance of $t = o(|w|)$ is possible. If independent generators are used each time that Gen is run, this construction is reusable.

A binary $w \in \{0,1\}^n$ can be amplified into a location source, whose zero error positions may be correlated (Definition 4). 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.

This construction has two important advantages over Canetti et al. [CFP+16]. First, storage linearly depends on the size of the source, in Canetti et al's construction storage was a complex function of (n, t) that grows quickly as t approaches a constant fraction of n. Second, many physical sources are

[1] The cryptographic key key is part of the $\mathbf{x}$ vector. Akavia, Goldwasser, and Vaikuntanathan showed when LWE is sufficiently hard parts of $\mathbf{x}$ are hardcore [AGV09].

sampled along with correlated side information that is called *confidence*. Confidence information is a secondary probability distribution Z (correlated with the reading W) that can predict the error rate in a bit W_i. When Z_i is large this means a bit of 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 $t = 3 \times 10^{-6} \cdot n$. This confidence information could not be used in Canetti et al's work to guide subset selection as it is correlated with W. Our construction can use Z at reproduction time only. Thus, nothing that depends on Z is revealed to the adversary.

Pattern Matching Obfuscation. In a recent work, Bishop et al. [BKM+18] show how to obfuscate a pattern $\mathbf{v}$ where each $v_i \in \{0, 1, \perp\}$ indicating that the bit v_i should match 0, 1 or either value. The goal is to allow a user to check for input string y, if y and 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 $\mathbf{A} \leftarrow (\mathbb{Z}_p)^{2n \times n}$. Then for a random $\mathbf{x}$ the construction outputs the following obfuscation:[2]

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

Bishop et al. prove security of the scheme in the generic group model. Intuitively, their argument rests on two facts: (i.) its hard to isolate wildcard positions where both values can be used to find $\mathbf{x}$ and (ii.) for nonwildcard positions its hard to pick a set without including errors. 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 4). In human language, it seems subsets of bits do have this property [Sha51, BPM+92, MZ11].

Concurrent Work. In concurrent work Bartusek, Lepoint, Ma, and Zhandry [BLMZ18] 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 $w < 0.774n$ to $w < n - \omega(\log n)$ 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

[2] 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.

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.

Organization. The remainder of the paper is organized as follows, Section 2 covers definitions and preliminaries, Section 3 presents our main theorem on hardness of decoding random linear codes in the generic group model. 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 of X given Y is $\tilde{\mathrm{H}}_\infty(X \mid Y) = -\log(\mathbb{E}_{y \in Y} \max_x \Pr[X = x \mid Y = y])$ [DORS08, Section 2.4]. 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 $\mathsf{Vol}(n, t, \mathcal{Z}) = \sum_{i=0}^{t} \binom{n}{i} (|\mathcal{Z}| - 1)^i$. U_n denotes the uniformly distributed random variable on $\{0, 1\}^n$. Unless otherwise noted logarithms are base 2. Usually, we use capitalized letters for random variables and corresponding lowercase letters for their samples.

2.1 The Generic Group Model and the Simultaneous Oracle Game

Definition 1 (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 2 (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. As stated in the introduction, 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 1 ([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.*

3 Hardness of Decoding in the Generic Group Model

In this section, we prove an upper bound for the probability that any generic adversary distinguish random linear codes with error from uniform as long as the error is sampled from a *Maximum Inner Product Unpredictable over Random Subspace* (MIPURS) distribution.

Definition 3. *Let $\mathbf{e}$ be a random variable taking values in $\mathbb{F}_q^n$ and let $A : \mathbb{F}_q^{n-k} \to \mathbb{F}_q^n$ denote uniform random linear operator (drawn independently of $\mathbf{e}$). We say that $\mathbf{e}$ is an $(k, \beta) -$ MIPURS distribution if*

$$\mathop{\mathbb{E}}_{A}\left[\min \mathbf{v} \in \mathbb{F}_q^{n-k} \setminus \mathbf{0}, \max_z \Pr[\langle \mathbf{Av}, \mathbf{e}\rangle = z]\right] \leq \beta\,.$$

Theorem 1. *Let λ be a security parameter. Let $q = q(\lambda)$ be a prime and $n = n(\lambda)$, $k = k(\lambda)$ be integers with $k \leq n \leq q$. Let $\mathbf{A} \in (\mathbb{F}_q)^{n\times k}$ and $\mathbf{x} \in (\mathbb{F}_q)^k$ be uniformly distributed. Let $\mathbf{e}$ be a $(k, \beta) -$ MIPURS distribution. Lastly, let $\mathbf{U} \in (\mathbb{F}_q)^n$ be uniformly distributed. Then for all* generic *adversaries $\mathcal{D}$ making at most m queries*

$$\Pr[\mathcal{D}(\mathbf{A}, \mathbf{Ax} + \mathbf{e}) = 1] - \Pr[\mathcal{D}(\mathbf{A}, \mathbf{U}) = 1] < ((m+1)m)^2\left(\frac{3}{2q} + \frac{\beta}{2}\right).$$

In particular, if $\alpha = \omega(\log \lambda)$, $q = \omega(\mathtt{poly}(\lambda)), n = \mathtt{poly}(\lambda)$, and $m = \mathtt{poly}(\lambda)$ then the statistical distance between the two cases is $\mathtt{ngl}(\lambda)$.

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

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

When presented a query, if the vector is in the oracle's table, it responds with the handle on the right side of the table. When the query is a new linear combination, it generates a distinct 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 2. *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*

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

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 there are $m(m+1)/2$ 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 $m(m+1)/2$ total values for the random variable, denoted $\texttt{EQ}$ representing the equality predicate pattern. Then, using a standard conditional min-entropy argument [DORS08, Lemma 2.2b]. Thus,

$$\forall i, \tilde{\mathrm{H}}_\infty(\mathbf{x}_i \mid \texttt{EQ}, \mathbf{A}) \geq \log q - \log \frac{m(m+1)}{2}.$$

Thus, it follows that after m queries,

$$\tilde{\mathrm{H}}_\infty(\chi(\mathbf{Ax}) \mid \mathbf{A}, \texttt{EQ}) \geq \log q - \log \frac{m(m+1)}{2}.$$

Thus, the probability that this linear combination represents a known value (on average across A) is:

$$\mathop{\mathbb{E}}_{\mathbf{A},\texttt{EQ}} \left[\max_z \Pr[(\chi(\mathbf{Ax}) = z \mid \mathbf{A}, \texttt{EQ}]\right] \leq \frac{m(m+1)}{2q}.$$

Case 2. Decomposing the linear combination of the codeword into $\chi(\mathbf{Ax}+\mathbf{e})$ we show that 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 $\log((m+1)m/2)$ bits so the probability is reduced to $\frac{(m+1)m\beta}{2}$

These two cases are mutually exclusive. Thus, to calculate the probability of either of these cases occurring after m queries 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 $(m+1)m/2$ 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

$$\left(\frac{(m+1)m}{2}\right)\left(\frac{(m+1)m}{2q} + \frac{(m+1)m\beta}{2}\right) = \left(\frac{((m+1)m)^2}{4}\right)\left(\frac{1}{q} + \beta\right).$$

This completes the proof of Lemma 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 1. 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 $1/q$ of returning a previously seen handle. We call this last oracle the uniform oracle and use it in our analysis with the added probability of failure.

Taking the result of this technical lemma, we can prove Theorem 1 using Lemma 1 (and the modification to the uniform oracle) where

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

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

$$1/2 + \left(\frac{((m+1)m)^2}{4}\right)\left(\frac{3}{q} + \beta\right)$$

then

$$\Pr[A(\mathcal{O}_c)=1]-\Pr[A(\mathcal{O}_r)=1] < 2\left(\frac{((m+1)m)^2}{2}\right)\left(\frac{3}{2q}+\frac{\beta}{2}\right)$$
$$= ((m+1)m)^2\left(\frac{3}{2q}+\frac{\beta}{2}\right).$$

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. □

3.1 Characterizing MIPURS

The definition of a MIPURS distribution (Def 3) is admittedly unwieldily. It considers a property of a vector $\mathbf{e}$ with respect to a random matrix. In this section we show that many natural sources satisfy this property. We begin with distributions where each component of $\mathbf{e}$ is independent and contributes some entropy.

Independent Sources In most versions of LWE, each error coordinate is independently distributed and contributes some entropy. Examples include the discretized Gaussian introduced by Regev [Reg05, Reg10], a uniform interval introduced by Döttling and Müller-Quade [DMQ13], and a uniform bit introduced by Micciancio and Peikert [MP13].

Lemma 3. *Let $\mathbf{e}=\mathbf{e}_1,...,\mathbf{e}_n \in \mathbb{F}_q$ be a distribution where each e_i is independently sampled and $\mathrm{H}_\infty(\mathbf{e}_i)=\Theta(1)$. Let $\ell \in \mathbb{Z}^+$ be some free parameter. Then for any $k=\omega(\log\lambda)$ then $\mathbf{e}$ is a $(k-\ell,\beta)-$ MIPURS distribution for*

$$\beta = \left(1-\frac{1}{q}\left(1-\frac{\binom{n}{k}}{q^\ell}\right)\right)2^{-(k-\ell)\mathrm{H}_\infty(\mathbf{e}_i)}+\frac{1}{q}\left(1-\frac{\binom{n}{k}}{q^\ell}\right).$$

Proof. We use $\mathbf{A}\in\mathbb{F}_q^{n\times n-k}$ to represent the random matrix from the definition of a MIPURS distribution and let $\mathbf{B}\in\mathbb{F}_q^{n\times k}$ represent its null space. Note that $\mathbf{A}$ is full rank with probability at least $1-1/q$.

Let ℓ be some parameter. With probability $\binom{n}{k}/q^l$, there will exist a $k\times k$ minor of $\mathbf{B}$ with rank less than $k-\ell$. As we will see this means that there is some $\mathbf{v}\in\mathtt{span}(\mathbf{A})$ where $\mathsf{wt}(\mathbf{v})\le k-\ell-1$. Otherwise, with probability at least $1-\binom{n}{k}/q^l$ all $\mathbf{v}\in\mathtt{span}(\mathbf{A})$ have $\mathsf{wt}(\mathbf{v})\ge k-\ell$. In the case, when all minors of $\mathbf{B}$ are full rank:

$$\max_z\{\Pr[\langle\mathbf{v},\mathbf{e}\rangle=z|\mathbf{A}]\}\le 2^{-\mathsf{wt}(v)\mathrm{H}_\infty(\mathbf{e}_i)}=2^{-(k-\ell)\mathrm{H}_\infty(\mathbf{e}_i)}.$$

The argument follows by assuming that there exists some $\mathbf{v}$ such that $\langle\mathbf{v},\mathbf{e}\rangle$ is constant in the case when $\mathbf{A}$ is not full rank or when $\mathbf{B}$ has minors of rank less than $k-\ell$. □

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" $\mathbf{s}\in\{0,1\}^n$: that is, $e'_i=e_i\cdot_c s_i$ where $\mathbf{s}$ is assumed to be unpredictable on all large enough subsets (here $\cdot_c$ is componentwise multiplication). More formally, we introduce a notion called subset entropy:

Definition 4. *Let a source $S=S_1,\ldots,S_n$ consist of n-bit binary strings. For some parameters k, α we say that the source S is has (α,k)-**entropy subsets** if $\mathrm{H}_\infty(S_{j_1},\ldots,S_{j_k})\ge\alpha$ for any $1\le j_1,\ldots,j_k\le n$.*

Lemma 4. *Let $\ell \in \mathbb{Z}^+$ and $k \in \mathbb{Z}^+$ be some free parameters. Let $\mathbf{s} \in \{0,1\}^n$ 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 a "selection vector" $\mathbf{s} \in \{0,1\}^n$: that is, $e'_i = e_i \cdot_c s_i$. Then the distribution $\mathbf{e}$ is a* MIPURS *distribution for $(k-\ell, \beta)$ for*

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

Proof. The proof follows the same basic structure as the proof of Lemma 3. Define $\mathbf{A}$ and $\mathbf{B}$ as above. Let ℓ be a free parameter. With probability $1 - \frac{1}{q} - \frac{\binom{n}{k}}{q^{\ell+1}}$ $\mathbf{A}$ is full rank and all $k \times k$ minors of $\mathbf{B}$ have rank at least $k-\ell$. For some $\mathbf{v}$ in the span of $\mathbf{A}$ with weight at least $k-\ell$ (assumed due to the rank of minors in $\mathbf{B}$), consider the product $\langle \mathbf{v}, \mathbf{e}' \rangle = \sum_{i=1}^n \mathbf{v}_i \cdot \mathbf{e}_i$. Define $\mathcal{I}$ as the set of nonzero coordinates in $\mathbf{v}$. 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 The last set of sources we consider are what we call $n-k+1$-*linear sources*. Here we consider some matrix $\mathbf{E} \in \mathbb{F}_q^{n \times (n-k+1)}$ and define the distribution $\mathbf{e} = \mathbf{E}\mathbf{s}$ 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 $(n-k+1)$.

Lemma 5. *Let $\mathbf{e}$ be defined by a $n-k+1$-linear source. Then $\mathbf{e}$ is a $(k, \beta) -$* MIPURS *distribution for*

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

Proof. Consider a random $\mathbf{A} \in \mathbb{F}_q^{n \times (n-k)}$. With probability at least $1 - 1/q$ the dimension of $\mathbf{A}$ is $n-k$. We start by bounding the probability that there exists some nonzero vector $\mathbf{v}$ where $\mathbf{v}$ is in both the span of $\mathbf{A}$ and the null space of $\mathbf{E}$. For each of the q^{n-k} vectors $\mathbf{v}$ in the span of $\mathbf{A}$ 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 $\mathbf{A}$ with trivial intersections with the null space of $\mathbf{E}$. Then for all $\mathbf{v} \in \mathtt{span}(A)$ it holds that $\mathbf{vE}$ is some nonzero vector and thus at least one uniform component of $\mathbf{s}$ contributes to the value of $\mathbf{vEs}$. □

4 Application to Fuzzy Extractors - Code Offset in the Exponent

Our primary 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 generic* $\mathcal{A}$ *making at most* m *queries to the group oracle* $\mathcal{O}$*, that is*

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

In the above, group elements in $\mathsf{Key}, U, \mathsf{Pub}$ are represented by group handles, the adversary additionally receives $\sigma(1)$. Additionally, 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.

4.1 Construction

The "code-offset" construction is a conceptually simple fuzzy extractor [DORS08]. The idea is for p to be a one time pad of w.[3] That is, $\mathsf{pub} = c \oplus w$. The Rep algorithm has pub and w' as inputs and computes $c' = \mathsf{pub} \oplus w'$. Importantly, if $\mathsf{dis}(w, w') \leq t$ then $\mathsf{dis}(c, c') \leq t$. If c is chosen from a suitable error correcting code, then it is possible to decode to c and recover w.

Importantly, if c is chosen from a error correcting code, it cannot be a uniform point and thus the one-time pad analysis does not apply. However, the conditional entropy of W given Pub or $\tilde{\mathrm{H}}_\infty(W \mid \mathsf{Pub})$ reduces by at most the gap between the size of the uniform distribution and the error correction code. That is, for code $C \in \{0,1\}^n$, $\tilde{\mathrm{H}}_\infty(W \mid \mathsf{Pub}) \geq \mathrm{H}_\infty(W) - (n - \log|C|)$.

The major problem with the code offset construction is the limited applicability to physical distributions W. In particular, for many distributions W this analysis provides no guarantee on the strength of the derived key (see discussion in [CFP+16]). To address this problem, Fuller et al. [FMR13] proposed to replace a structured code with a random linear code and rely on the hardness of LWE. Subsequent work adapted the construction to $\mathbb{F}_2$ [HRvD+16] and showed how to make the construction a reusable fuzzy extractor [ACEK17]. These constructions are a code-offset construction with a random code: $\mathsf{pub} = (\mathbf{A}, \mathbf{Ax} + \mathbf{w})$ where $\mathbf{A}$ and $\mathbf{x}$ are random. For a suitable $\mathbf{w}$ the value pub is pseudorandom. The associated decoding procedure is a simple guess and check, finding subsets $\mathcal{I}_j$ and then computing $\mathbf{x} = \mathbf{A}_{\mathcal{I}_j}^{-1}\mathsf{pub}_{\mathcal{I}_j} - \mathbf{w}'_{\mathcal{I}_j} = \mathbf{x}\mathbf{A}_{\mathcal{I}_j}^{-1}(\mathbf{w} - \mathbf{w}')_{\mathcal{I}_j}$. To achieve a hard LWE instance their correction capability was only $t = \Theta(\log|\mathbf{w}|)$ which is inadequate for most physical sources. Achieving a higher error tolerance could be achieved by reducing the dimension of the code $|\mathbf{x}|$ but this has a direct effect on the hardness of the underlying lattice problem. Since our generic group proof shows hardness for traditional LWE distributions, we can immediately expand the number of distributions for which this construction is secure. Thus, we move the code-offset to the exponent.

Before introducing the construction we observe it is possible to amplify the hardness of the distribution W. Since decoding finds a subset without errors (it does not rely on the magnitude of errors) we can augment errors into random errors. Consider a binary biometric W and a random vector E and multiply them component wise to get a distribution E'. If subsets of W are unguessable then the distribution formed E' is MIPURS (see Section 3.1 and Definition 4).

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). Thus, we introduce an auxiliary uniform random variable Y 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 standard 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).

[3] The cryptographic key is produced by applying a randomness extractor on w. [NZ93].

Construction 1. *Let λ be a security parameter, t be a distance and let $k = k(\lambda)$ be some parameter where $k = \omega(\log \lambda)$. Let α be a free parameter. Let $q = q(\lambda)$ be an ensemble of primes. Let $\mathbb{F}_q$ be the field with q elements. Let $\mathcal{W} \in \mathbb{F}_q^n$ be equipped with the Hamming metric and let $k = \omega(\log \lambda)$ be a parameter. Let $\tau = \max(0.01, t/n)$. Define* $(\mathsf{Gen}, \mathsf{Rep})$ *as follows:*

Gen

1. ***Input:*** $w = w_1, ..., w_n$
2. *Sample random generator r of $\mathbb{Z}_p^*$.*
3. *Sample $\mathbf{A} \leftarrow (\mathbb{F}_q)^{n \times (k+\alpha)}, \mathbf{x} \leftarrow (\mathbb{F}_q)^{k+\alpha}$.*
4. *Sample $\mathbf{y} \xleftarrow{\$} \{0,1\}^n$.*
5. *For $i = 1, ..., n$:*
   - *(i) If $w_i = y_i$, set $\mathsf{c}_i = r^{\mathbf{A}_i \cdot x}$.*
   - *(ii) Else set $\mathsf{c}_i \xleftarrow{\$} \mathbb{Z}_p^*$.*
6. *Set $\mathsf{key} = r^{\mathbf{x}_{0...\alpha-1}}$.*
7. *Output (key, p), $p = (r, \mathbf{y}, \mathbf{A}, \{\mathsf{c}_i\}_{i=1}^n)$.*

Rep

1. ***Input:*** $(w', p = (r, \mathbf{y}, \mathbf{A}, \mathsf{c}_1 \dots \mathsf{c}_\ell))$
2. *Let $\mathcal{I} = \{i | w'_i = y_i\}$.*
3. *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 $\mathsf{key} = r_{0...\alpha-1}^{\mathbf{A}_{J_i}^{-1} \mathbf{c}_{J_i}}$.*
4. *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 each 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 $u \leftarrow \{0,1\}^\kappa$.*
- ***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 generic D making at most m queries and all $j = 1, ..., \rho$, the advantage is at most ϵ_{sec}.*

Theorem 2. *Let all parameters be as in Construction 1. Let $\ell \in \mathbb{Z}^+$ be a free parameter. Let $W^1, ..., W^\rho \in \{0,1\}^n$ be distributions and define the random variables*

$$\mathbf{E}^j \stackrel{def}{=} \begin{cases} U_{\mathbb{Z}_p^*} & \text{if } Y_i^j \neq W_i^j, \\ 0 & \text{if } Y_i^j = W_i^j. \end{cases}$$

Suppose that $\mathbf{E}^j$ are $(k,\beta)-\mathsf{MIPURS}$ distributions for $1 \leq j \leq \rho$. Then $(\mathsf{Gen}, \mathsf{Rep})$ is a $(\rho, \epsilon_{sec}, m)$ reusable fuzzy extractor for all generic adversaries making at most m queries where

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

For generic adversaries making m queries, Construction 1 is a reusable secure fuzzy extractor if $\beta = \mathtt{ngl}(\lambda), q = \omega(\mathtt{poly}(\lambda)), m = \mathtt{poly}(\lambda)$, and $\rho = \mathtt{poly}(\lambda)$ for some $\epsilon_{sec} = \mathtt{ngl}(\lambda)$.

Proof. This argument requires a little understanding of the generic group proof from Section 3. This argument showed that an adversary knowing $\mathbf{A}$ was 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:

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

Crucially, in Theorem 1, 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 1 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 1. □

We show parameters where the construction is efficient and correct in Appendix A. If $k + \alpha$ is just barely $\omega(\log n)$ one can support error rates that are just barely $o(n)$.

Comparison with sample-then-lock As mentioned in the introduction, Canetti et al. [CFP+16] proposed a reusable fuzzy extractor based on digital lockers called *sample-then-lock*. Intuitively, a digital locker is a symmetric encryption that is semantically secure even when instantiated with keys that are correlated and only have entropy [CKVW10]. At a high level, their construction took multiple samples $w_{\mathcal{I}_j}$ from the input biometric and use these as keys for different digital lockers, all of which contained the same key. Our construction improves on the storage and use of confidence information over Canetti et al. (see the Introduction). On the other hand the fact that all subsets are available to an adversary does provide them with additional power. As mentioned in Section 3.1, 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 modification to this definition in Definition 4 is the natural one. Instead of $j_1, \ldots, j_k$ being a uniform subset, it can be any subset of n.

5 Application to 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 $\mathcal{C} = \mathcal{C}_n$ be a family of circuits where $\mathcal{C}_n$ takes inputs of length n and let $\mathcal{O}$ be a PPT algorithm taking $n \in \mathbb{N}$ and $C \in \mathcal{C}$ outputting a new circuit C'. Let $\mathcal{D} = \mathcal{D}_n$ 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 $\mathcal{D}$ over $\mathcal{C}$ 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 $q = q(n)$.*

$\mathcal{O}$:

1. *Input $\mathbf{v} \in \{0, 1, \perp\}^n, q, g$ where g is a generator of the group $\mathbb{Z}_q^*$.*
2. *Sample $\mathbf{A} \in (\mathbb{Z}_q)^{2n \times n}, x_0 = 0, x_{1,\ldots,n-1} \leftarrow (\mathbb{Z}_q)^{n-1}$.*
3. *Sample $\mathbf{e} \in \mathbb{Z}_q^{2n}$ uniformly.*
4. *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* $e_{2i} = 0, e_{2i+1} = 0$.

5. *Compute* $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}$.
6. *Output* $g^{\mathbf{y}}, \mathbf{A}$.

Eval

1. *Input* $g^{\mathbf{y}}, \mathbf{A}, x \in \{0,1\}^n$.
2. $\mathcal{I} = \{i \in [1...2n] \mid x_{\lfloor i/2 \rfloor} = (i \mod 2)\}$.
3. *Compute* $\mathbf{A}_{\mathcal{I}}^{-1}$. *If none exists output* $\perp$.
4. *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 3. *Let* $\ell \in \mathbb{Z}^+$ *be a free parameter. Define* $\mathcal{D}$ *as the set of all distributions* V *such that* $E' = U_{\mathbb{F}_q}^n \cdot_c \mathsf{Bin}(V)$ *is a distribution that is* $(n, \beta) - \mathsf{MIPURS}$. *Then Construction 2 is VBB secure for generic* $\mathcal{D}$ *making at most* m *queries with distinguishing probability at most*

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

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 1, 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 consider whether decoding is hard for groups where the discrete logarithm problem is believed to be hard. We first examine hardness of decoding random linear codes in the exponent. In Appendix B we consider Reed-Solomon codes. Both results follow the same 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 $\mathtt{FIND-REP}$ and completes the connection to hardness of discrete log.

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

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 $\mathbf{z} = \mathbf{Ax} + \mathbf{e}$.

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 $\mathtt{BDDE-RL}(n, k, q, c, g)$, 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 $g^{\mathbf{y}} = g^{\mathbf{Ax}+\mathbf{e}}$ where $\mathbf{A}, \mathbf{x}$ are uniformly distributed. Input is $g^{\mathbf{y}}, \mathbf{A}$.

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

For a code $\mathbf{C}$ we define the distance between a point $\mathbf{y}$ and the code as the minimum distance between $\mathbf{y}$ and any codeword $\mathbf{c}$ in $\mathbf{C}$. Formally, $\mathsf{dis}(\mathbf{y}, \mathbf{C}) = \min_{\mathbf{c} \in \mathbf{C}} \mathsf{dis}(\mathbf{y}, \mathbf{c})$.

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 6), then prove the lemma and finally the theorem.

Theorem 4. *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 $\mathtt{BDDE-RL}(n, k, q, n-k-c, g)$ 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 6. *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 6. 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

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

□

Proof of Theorem 4. 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 6, $(\mathbf{y}, \mathbf{A})$ 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

[ACEK17] Daniel Apon, Chongwon Cho, Karim Eldefrawy, and Jonathan Katz. Efficient, reusable fuzzy extractors from LWE. In *International Conference on Cyber Security Cryptography and Machine Learning*, pages 1–18. Springer, 2017.

[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.

[BC10] Nir Bitansky and Ran Canetti. On strong simulation and composable point obfuscation. In *Advances in Cryptology–CRYPTO 2010*, pages 520–537. Springer, 2010.

[BCKP17] Nir Bitansky, Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On virtual grey box obfuscation for general circuits. *Algorithmica*, 79(4):1014–1051, 2017.

[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.

[BLMZ18] James Bartusek, Tancrède Lepoint, Fermi Ma, and Mark Zhandry. New techniques for obfuscating conjunctions. Cryptology ePrint Archive, Report 2018/936, 2018. `https://eprint.iacr.org/2018/936`.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 575–584. ACM, 2013.

[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.

[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.

[BV14] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871, 2014.

[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.

[CW07] Qi Cheng and Daqing Wan. On the list and bounded distance decodability of reed–solomon codes. *SIAM Journal on Computing*, 37(1):195–209, 2007.

[Dau04] John Daugman. How iris recognition works. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):21 – 30, January 2004.

[Des92] Yvo Desmedt. Threshold cryptosystems. In *Advances in Cryptology – AUSCRYPT*, pages 1–14. Springer, 1992.

[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.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[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.

[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.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. *Journal of the ACM (JACM)*, 62(6):45, 2015.

[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.

[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.

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

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

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

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

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

[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.

[WCD+17] Joanne Woodage, Rahul Chatterjee, Yevgeniy Dodis, Ari Juels, and Thomas Ristenpart. A new distribution-sensitive secure sketch and popularity-proportional hashing. In *Annual International Cryptology Conference*, pages 682–710. Springer, 2017.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under lwe. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 600–611. IEEE, 2017.

A Correctness of Fuzzy Extractor

Correctness and Efficiency We now show the construction is also 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 5. *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 5. 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 7. *[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 7 completes the proof of Theorem 5. □

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 $t = o(n)$.

B Decoding Reed Solomon Codes in the Exponent

The Reed-Solomon family of error correcting codes [RS60] have extensive applications in cryptography. For the field $\mathbb{F}_q$ of size q, a message length k, and code length n, such that $k \leq n \leq q$, define the Vandermonde matrix $\mathbf{V}$ where the ith row, $\mathbf{V}_i = [i^0, i^1, ..., i^k]$. The Reed Solomon Code $\mathbb{RS}(n,k,q)$ is the set of all points $\mathbf{Vx}$ 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 6. *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 4 the core of our theorem is a bound on the probability that a random point is close to a Reed-Solomon code.

Lemma 8. *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 8. 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 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 8. □

The remainder of the proof is similar to the proof of Theorem 4. $\mathcal{A}$ works as follows: on input $\mathbf{y}$ where $\mathbf{y}$ is uniform over G^n immediately run $\mathcal{D}(g, \mathbf{y})$. By Lemma 8, $(g, \mathbf{v})$ 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 6 follow in the case when $\alpha = 4q/n^2 \geq 2$ by noting that:

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

□