Computational Fuzzy Extractors[⋆,⋆⋆]

Benjamin Fuller[a,*], Xianrui Meng[b], Leonid Reyzin[c]

[a] *University of Connecticut, 371 Fairfield Way, Storrs, CT 06269, United States.*
[b] *Amazon Web Services, Inc. 2021 7th Avenue, Seattle, WA 98121, United States.*
[c] *Boston University, 111 Cummington Mall, Boston, MA 02215, United States.*

Abstract

Fuzzy extractors derive strong keys from noisy sources. Their security is usually defined information-theoretically, with gaps between known negative results, existential constructions, and polynomial-time constructions. We ask whether using computational security can close these gaps. We show the following:

- **Negative Result:** Noise tolerance in fuzzy extractors is usually achieved using an information reconciliation component called a *secure sketch.* We show that secure sketches are subject to upper bounds from coding theory even when the information-theoretic security requirement is relaxed. Specifically, we define computational secure sketches using conditional HILL pseudoentropy (Håstad et al., SIAM J. Computing 1999). We show that a computational secure sketch implies an error-correcting code. Thus, HILL pseudoentropy is bounded by the size of the best error-correcting code. Similar bounds apply to information-theoretic secure sketches.
- **Positive Result:** We show that our negative result can be avoided by constructing and analyzing a computational fuzzy extractor directly. We modify the code-offset construction (Juels and Wattenberg, CCS 1999) to use random linear codes. Security is based on the Learning with Errors (LWE) problem and holds when the noisy source is uniform or symbol-fixing (that is, each dimension is either uniform or fixed). As part of the proof, we reduce symbol-fixing security to uniform error security.

Keywords: Fuzzy extractors, secure sketches, key derivation, learning with errors, error-correcting codes, computational entropy.

⋆A preliminary version of this work appeared in Asiacrypt 2013 [1]. The differences are described in Section 1.3.

⋆⋆Declarations of Interest: In addition to affiliations with UConn, Amazon and Boston University, the authors have previously been affiliated with MIT Lincoln Laboratory, Apple, MIT, IST Austria, and Algorand. Other than potential organizational conflicts of interest, the authors foresee no potential conflicts.

*Corresponding author

Email addresses: benjamin.fuller@uconn.edu (Benjamin Fuller), xianru@amazon.com (Xianrui Meng), reyzin@cs.bu.edu (Leonid Reyzin)

1. Introduction

Authentication requires a secret drawn from some high-entropy source. One of the primary building blocks for authentication is reliable key derivation. Unfortunately, many sources that contain sufficient entropy to derive a key are noisy and provide similar but not identical secret values at each reading. Examples of such sources include biometrics [2], measurements of capacitance [3], timing [4], motion [5], and quantum information [6].

Fuzzy extractors [7] derive reliable keys from noisy sources (see [8, 9, 10, 11] for applications of fuzzy extractors). The primitive consists of two algorithms: Generate (used once) and Reproduce (used subsequently). The Generate (Gen) algorithm takes an input w and produces a key r and a public value p. The Reproduce (Rep) algorithm is able to reproduce r given p and some value w' that is close to w (according to some predefined metric, such as Hamming distance). Crucially for security, knowledge of p should not reveal r; that is, r should be uniformly distributed conditioned on p. This feature is needed because p is not secret: for example, in a single-user setting (where the user wants to reproduce the key r from a subsequent reading w'), it would be stored in the clear; and in a key agreement application [8] (where two parties have w and w', respectively), the natural solution is to send p between the parties. (More techniques are possible when interactive communication is permitted; see Dupont et al. for a recent example [12].)

Fuzzy extractors use ideas from information-reconciliation and privacy amplification [6] and are defined (traditionally) as information-theoretic objects. Privacy amplification is usually performed with a randomness extractor [13]. Randomness extractors are well-understood [14]. Polynomial-time constructions of randomness extractors can extract randomness from all distributions with min-entropy with the help a short uniform nonsecret seed. A single randomness extractor simultaneously works for all probability distributions with sufficient entropy. Furthermore, for randomness extractors, the parameter gap between negative results, nonconstructive positive results, and polynomial-time constructions is relatively small.

Unfortunately, the state of fuzzy extractors is murkier. There is no crisp characterization of when key derivation is possible. Fuller, Reyzin, and Smith [15, 16] present one possible notion called *fuzzy min-entropy*. They show a non-polynomial-time algorithm that derives a key from each distribution with fuzzy min-entropy. Woodage et al. [17] subsequently improved the parameters. As a negative result, Fuller, Reyzin, and Smith [15, 16] and Fuller and Peng [18] show families of distributions where no fuzzy extractor can simultaneously work for the whole family, despite the fact that a fuzzy extractor exists for each element of the family. Thus, two main open areas of research for information-theoretic fuzzy extractors are providing polynomial-time constructions and providing constructions that simultaneously secure many distributions. This work asks:

Can computational security close these gaps?

1.1. Our Contribution

We consider the *sketch-then-extract* paradigm used in most fuzzy extractor constructions. This paradigm combines a *secure sketch* and a *randomness extractor*. A *secure sketch* is a one-round information-reconciliation protocol. It allows recovery of the original value w from any nearby value w'. A randomness extractor is then run on w to produce uniform bits. One could replace the usual, information-theoretic, randomness extractor with a computational one [19, 20, 21] (constructed, for example, by applying a pseudorandom generator to the output of an information-theoretic extractor), but a computational extractor helps only if the conditional min-entropy of w conditioned on the sketch is high enough (else, the computational extractor has no security). Since the security losses due to secure sketches are usually much higher than due to randomness extraction, the secure sketch becomes the bottleneck.

We ask if a computational secure sketch can overcome information-theoretic lower bounds. The most natural relaxation of the min-entropy requirement of the secure sketch is to require HILL entropy [22] (namely, that the distribution of w conditioned on the sketch be *indistinguishable* from a high-min-entropy distribution). Under this definition, one could use a randomness extractor to obtain r from w, resulting in a pseudorandom key.

Negative Result. We prove in Theorem 3.6 that the entropy loss of such *computational* HILL secure sketches is subject to coding bounds that are similar to the ones that constrain information-theoretic secure sketches. More precisely, for every secure sketch that retains m bits of computational entropy, there is an error-correcting code with 2^{m-2} codewords. This error-correcting code can then be used to instantiate an information-theoretic secure sketch.

The idea is that, by definition of HILL entropy, an adversary should not be able to distinguish a pair w, p from x, p where x is drawn from a distribution with actual entropy conditioned on p. For most points w' close to w, the output of $\mathsf{Rec}(w', p) = w$. Thus, the same must be true for points x drawn conditioned on a given p (or else we could build a distinguishing adversary), forcing the distribution of x conditioned on p to function as an error-correcting code.

Alternative Computational Definitions for Secure Sketches. We define computational secure sketches via HILL entropy. A natural question is whether a weaker definition of security for secure sketches could avoid the negative result. A minimum condition is computational unpredictability of w given p [23]. If such a definition is used, one can instantiate sketch-and-extract with a reconstructive extractor [24, 23] (one way to build such an extractor is via repeated, independent applications of the Goldreich-Levin hardcore function [25]). Constructing secure sketches with computational unpredictability of w given p, or proving negative results about them, is a fascinating open problem.

Let us briefly discuss two other alternative definitions of pseudoentropy, called inaccessible entropy [26, 27] and next-block pseudorandomness [28, 29]. Inaccessible entropy measures the difference between the entropy of w conditioned on p and the ability of an adversary to find other values w^* that are

consistent with p. Since inaccessible entropy is bounded above by actual entropy it is not clear how to adapt this tool.

Next-block pseudorandomness asks that the distribution of each symbol w_i of w is indistinguishable, conditioned on $w_1, ..., w_{i-1}, p$, from some distribution X_i such that the sum of the conditional entropies of X_i is high enough. Next-block pseudorandomness is used in building pseudorandom generators from one-way functions. It may be possible to build a good fuzzy extractor from this definition by modifying the subsequent extraction procedure, perhaps using techniques from [28, 29]. However, it may be that secure sketches based on this indistinguishability-style definition are subject coding-theory bounds similar to those for secure sketches based on HILL entropy, and this definition will not lead to improved constructions. Resolving these questions is another fascinating open problem.

For now, to avoid our negative result, we focus on directly constructing a *computational fuzzy extractor*. That is, in our construction, we will show that the output key r is indistinguishable from uniform (conditioned on p). To avoid the negative result for secure sketches, the pair (r, p) must be *one-way* in the value w.

Positive Result. We construct the first fuzzy extractor whose security relies on computational security arguments (Juels and Sudan suggested using computational security in [30]). The construction can derive a key r whose length is at least the entropy of the source w. Our construction is for the Hamming metric and uses the code-offset construction [31],[7, Section 5] used in prior work, but with two crucial differences. First, the key r is not extracted from w like in the sketch-and-extract approach; rather w "encrypts" r in a way that is decryptable with the knowledge of some close w' (this idea is similar to the way the code-offset construction is presented in [31] as a "fuzzy commitment"). Our construction uses private randomness within Gen, which is allowed in the fuzzy extractor setting but not for noiseless randomness extraction. Second, the code used is a random linear code, which allows us to use the Learning with Errors (LWE) assumption due to Regev [32, 33, 34] and derive a longer key r.

For security, we rely on the result of Döttling and Müller-Quade [35], which shows the hardness of decoding random linear codes when the error vector comes from the uniform distribution, with each coordinate ranging over a small interval. This allows us to use w as the error vector, assuming it is uniform. There have been subsequent works on uniform error LWE [36, 37]; however as we discuss in Section 4.2, these changes do not substantively effect our parameters. We also use a result of Akavia, Goldwasser, and Vaikuntanathan [38], which says that LWE has many hardcore bits, to hide r.

Because we use a random linear code, our decoding is limited to guessing a subset of locations and checking if it contained errors. Unfortunately, we cannot utilize the results that improve the decoding radius through the use of trapdoors (such as [32, 34]), because in a fuzzy extractor, there is no secret storage place for the trapdoor (in particular, Gen cannot pass a secret to Rep). If improved decoding algorithms are obtained for random linear codes, they will improve the

error-tolerance of our construction. However, the problem of generally decoding random linear codes is NP-hard [39].

The construction is secure whenever w is drawn from an error distribution that makes the decisional version of the LWE problem hard. Toward this end, we show the hardness of LWE when some dimensions of the error vector are fixed (and adversarially known), which may be of independent interest (Theorem 5.2). This allows w to come from a symbol-fixing source [40] (each dimension is either uniform or fixed).

1.2. Subsequent Work

Subsequent to the introduction of computational fuzzy extractors in the conference version of this work [1], other works built computational fuzzy extractors for noisy sources for which no efficient information-theoretic construction is known (e.g., [41]). Under strong cryptographic assumptions (semantically secure graded encodings), a polynomial-time computational fuzzy extractor exists for every source where the distance metric is computable in the complexity class NC^1 [42].

A desirable property for fuzzy extractors is *reusability* [43], which guarantees that a user can securely enroll the value w with multiple independent providers to get values $r_1, p_1, ..., r_\rho, p_\rho$. Even with noise between different enrollments, each key r_i should be private conditioned on the rest of the values. Boyen showed strong negative results on information-theoretically secure reuseable fuzzy extractors [43].

Apon et al. [44] showed that the construction presented in this paper achieves a weak form of reusability if it is modified so that the random code is a global parameter (instead of being created as part of Gen). They also show how to augment the reusability using either a random oracle or LWE-based symmetric encryption techniques. Other subsequent work used different cryptographic techniques to construct reusable computational fuzzy extractors [41, 45, 46, 47].

Our security arguments are based on the learning-with-errors assumption with $q > 2$. Herder et al. [48] present a similar construction when $q = 2$ that reduces to a form of learning parity with noise [49]. Herder et al.'s construction is secure when the bits of w are independent Bernoulli trials. They also show security when w comes from a class of affine transformations [48, Section 7]. Lastly, Huth et al. [50, 51, 52] implemented our construction on multiple devices, including a constrained 8-bit microcontroller.

1.3. Differences between [1] and this work

The same authors published a conference version of this work in Asiacrypt 2013 [1]. That work did not include proofs or a detailed discussion of parameters. The theorem statement and the underlying proof in Section 3 had a minor error (pointed out by Yasunaga and Yuzawa [53]). This version corrects the theorem statement and proof. There was also a second negative result for secure sketches that is superseded by a more recent result in [15]; this is discussed in Section 3. Additionally, the conference version focused on extracted key length for high-entropy inputs as the sole reason to move to computational security. Since the

conference version, it became evident that there are other important reasons. In particular, there is a large gap between known negative results for information-theoretic fuzzy extractors and positive constructions. There are many sources of practical importance, such as the iris [54] and physical unclonable functions [55], for which the best known information-theoretic fuzzy extractors provide little or no security. Since the publication of the conference version of this paper, computational constructions [55, 54] have been able to provide meaningful, albeit modest, security for such sources, while adding additional properties such as reusability. Lastly, this version discusses more recent results on uniform-error LWE and their applicability to our setting (in Section 4.2).

2. Preliminaries

For a random variable $X = X_1||...||X_n$ where each X_i is over some alphabet $\mathcal{Z}$, we denote by $X_{1,...,k} = X_1||...||X_k$. The *min-entropy* of X is

$$\mathrm{H}_\infty(X) = -\log(\max_x \Pr[X = x]),$$

and the *average (conditional)* min-entropy [7, Section 2.4] of X given Y is

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

The *statistical distance* between random variables X and Y with the same domain is $\Delta(X, Y) = \frac{1}{2}\sum_x |\Pr[X = x] - \Pr[Y = x]|$. For a distinguisher D (or a class of distinguishers $\mathcal{D}$) we write the *computational distance* between X and Y as $\delta^D(X, Y) = |\mathbb{E}[D(X)] - \mathbb{E}[D(Y)]|$. We denote by $\mathcal{D}_{s_{sec}}$ the class of randomized circuits which output a single bit and have size at most s_{sec}. 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, $B_t(x) = \{y|\mathsf{dis}(x, y) \le t\}$. If the size of a ball in a metric space does not depend on x, we denote by $|B_t(\cdot)|$ the size of a ball of radius t. For the Hamming metric over $\mathcal{Z}^n$, $|B_t(\cdot)| = \sum_{i=0}^{t} \binom{n}{i}(|\mathcal{Z}| - 1)^i$. U_n denotes the uniformly distributed random variable on $\{0, 1\}^n$. Usually, we use bold letters for vectors or matrices, capitalized letters for random variables, and lowercase letters for elements in a vector or samples from a random variable. We use $\mathtt{poly}(n)$ to denote some polynomial function of n and $\mathtt{ngl}(n)$ to denote some negligible function of n.

2.1. Fuzzy Extractors and Secure Sketches

We now recall definitions and lemmas from the work of Dodis et al. adapted to allow for a small probability of error [7, Sections 8]. Let $\mathcal{M}$ be a metric space with distance function dis.

Definition 2.1. *An $(\mathcal{M}, m, \ell, t, \epsilon)$-*fuzzy extractor *with error δ is a pair of randomized procedures, "generate" (Gen) and "reproduce" (Rep), with the following properties:*

1. *The generate procedure* Gen *on input* $w \in \mathcal{M}$ *outputs an extracted string* $r \in \{0,1\}^\ell$ *and a helper string* $p \in \{0,1\}^*$.

2. *The reproduction procedure* Rep *takes an element* $w' \in \mathcal{M}$ *and a bit string* $p \in \{0,1\}^*$ *as inputs.*

3. Correctness: *for every pair* w, w' *such that* $\mathsf{dis}(w,w') \leq t$, *for* $(R,P) \leftarrow \mathsf{Gen}(w)$, *then* $\mathsf{Rep}(w',P) = R$ *with probability (over the coins of* $\mathsf{Gen}, \mathsf{Rep}$*) at least* $1-\delta$. *If* $\mathsf{dis}(w,w') > t$, *then no guarantee is provided about the output of* Rep.

4. Security: *for any distribution* W *on* $\mathcal{M}$ *of min-entropy* m, *the string* R *is nearly uniform even for those who observe* P: *if* $(R,P) \leftarrow \mathsf{Gen}(W)$, *then* $\mathbf{SD}((R,P),(U_\ell,P)) \leq \epsilon$.

A fuzzy extractor is efficient if Gen *and* Rep *run in expected polynomial-time.*

We ask whether better parameters can be achieved by considering a fuzzy extractor with a computational security requirement. We therefore relax the security requirement of Definition 2.1 to require a pseudorandom output instead of a truly random output. We also modify the definition so that we can specify a general class of sources for which the fuzzy extractor is designed to work, rather than limiting ourselves to the class of sources with a given min-entropy m, as in definitions above. This modification can also be applied to definitions of information-theoretic secure sketches and fuzzy extractors.

Definition 2.2 (Computational Fuzzy Extractor)**.** *Let* $\mathcal{W}$ *be a family of probability distributions over* $\mathcal{M}$. *A pair of randomized procedures "generate"* (Gen) *and "reproduce"* (Rep) *is a* $(\mathcal{M}, \mathcal{W}, \ell, t)$-computational fuzzy extractor *that is* (ϵ, s_{sec})*-hard with error* δ *if* Gen *and* Rep *are a fuzzy extractor with the security property replaced with the following:*

4. Security: *for any* $W \in \mathcal{W}$, *the string* R *is pseudorandom conditioned on* P, *that is* $\delta^{\mathcal{D}_{s_{sec}}}((R,P),(U_\ell,P)) \leq \epsilon$.

Each efficient fuzzy extractor is a computational fuzzy extractor.

Remark. Fuzzy extractor definitions make no guarantee about Rep behavior when the distance between w and w' is larger than t. In the information-theoretic setting this seemed inherent as the "correct" R should be information-theoretically unknown conditioned on P. However, in the computationally setting this is not true. Looking ahead, in our construction R is information-theoretically determined conditioned on P (with high probability over the coins of Gen). Our Rep algorithm will never output an incorrect key (with high probability over the coins of Gen) but may not terminate. However, it is not clear this is the desired behavior. For this reason, we leave the behavior of Rep ambiguous when $\mathsf{dis}(w,w') > t$.

2.2. Secure sketches

Secure sketches are the main ingredient in the construction of most fuzzy extractors. Secure sketches produce a string s that does not decrease the entropy of w too much, while allowing recovery of w from a close w':

Definition 2.3. *An* $(\mathcal{M}, m, \tilde{m}, t)$-secure sketch *with error* δ *is a pair of randomized procedures, "sketch"* (SS) *and "recover"* (Rec)*, with the following properties:*

1. *The sketching procedure* SS *on input* $w \in \mathcal{M}$ *returns a bit string* $s \in \{0,1\}^*$.
2. *The recovery procedure* Rec *takes an element* $w' \in \mathcal{M}$ *and a bit string* $s \in \{0,1\}^*$.
3. Correctness: *if* $\mathsf{dis}(w, w') \le t$*, then* $\Pr[\mathsf{Rec}(w', \mathsf{SS}(w)) = w] \ge 1 - \delta$ *(probability over the coins of* SS *and* Rec*).*
4. Security: *for any distribution* W *over* $\mathcal{M}$ *with min-entropy* m*, the value of* W *can be recovered by the adversary who observes* $\mathsf{SS}(W)$ *with probability no greater than* $2^{-\tilde{m}}$*. That is,* $\tilde{\mathrm{H}}_\infty(W|\mathsf{SS}(W)) \ge \tilde{m}$.

A secure sketch is efficient *if* SS *and* Rec *run in expected polynomial-time.*

In the above definition, the errors are chosen before the algorithms are run. Correctness is not guaranteed if the error pattern between w and w' depends on the output of the algorithms. A fuzzy extractor can be produced from a *secure sketch* and an *average-case randomness extractor*. An average-case extractor is a generalization of a strong randomness extractor [13, Definition 2]) (Vadhan [56, Problem 6.8] showed that all strong extractors are average-case extractors with a slight loss of parameters):

Definition 2.4. *Let* χ_1*,* χ_2 *be finite sets. A function* $\mathtt{ext} : \chi_1 \times \{0,1\}^d \to \{0,1\}^\ell$ *is an* (m, ϵ)-average-case extractor *if for all pairs of random variables* X, Y *over* χ_1, χ_2 *such that* $\tilde{H}_\infty(X|Y) \ge m$*, then* $\Delta((\mathtt{ext}(X, U_d), U_d, Y), (U_\ell, U_d, Y)) \le \epsilon$.

Lemma 2.5. *Assume* $(\mathsf{SS}, \mathsf{Rec})$ *is an* $(\mathcal{M}, m, \tilde{m}, t)$*-secure sketch with error* δ*, and let* $\mathtt{ext} : \mathcal{M} \times \{0,1\}^d \to \{0,1\}^\ell$ *be a* $(\tilde{m}, \epsilon)$*-average-case extractor. Then the following* $(\mathsf{Gen}, \mathsf{Rep})$ *is an* $(\mathcal{M}, m, \ell, t, \epsilon)$*-fuzzy extractor with error* δ*:*

- $\mathsf{Gen}(w)$: *generate* $x \leftarrow \{0,1\}^d$*, set* $p = (\mathsf{SS}(w), x), r = \mathtt{ext}(w; x)$*, and output* (r, p).
- $\mathsf{Rep}(w', (s, x))$: *recover* $w = \mathsf{Rec}(w', s)$ *and output* $r = \mathtt{ext}(w; x)$.

3. Impossibility of Computational Secure Sketches

In this section, we show that a sketch that retains HILL entropy implies a large error-correcting code. For inputs that have full entropy this immediately implies a sketch that retains nearly the same amount of min-entropy. HILL entropy is

a commonly used computational notion of entropy [22]. It was extended to the conditional case by Hsiao, Lu, Reyzin [23]. Here we recall a weaker definition due to Gentry and Wichs [57] (the term *relaxed HILL entropy* was introduced in [58]); since we show impossibility even for this weaker definition, impossibility for the stronger definition follows immediately.

Definition 3.1. *Let (W, S) be a pair of random variables. W has* relaxed HILL entropy *at least k conditioned on S, denoted $H^{\mathtt{HILL\text{-}rlx}}_{\epsilon,s_{sec}}(W|S) \geq k$ if there exists a joint distribution (X, Y), such that $\tilde{H}_\infty(X|Y) \geq k$ and $\delta^{\mathcal{D}_{s_{sec}}}((W, S), (X, Y)) \leq \epsilon$.*

Intuitively, HILL entropy acts like as average min-entropy for all computationally bounded observers. Thus, redefining secure sketches using HILL entropy is a natural relaxation of the original information-theoretic definition; in particular, the sketch-and-extract construction in Lemma 2.5 would yield pseudorandom outputs if the secure sketch ensured high HILL entropy. We will consider secure sketches that retain relaxed HILL entropy: that is, we say that $(\mathsf{SS}, \mathsf{Rec})$ is a *HILL-entropy $(\mathcal{M}, m, \tilde{m}, t)$ secure sketch* that is (ϵ, s_{sec})-hard with error δ if it satisfies Definition 2.3, with the security requirement replaced by $H^{\mathtt{HILL\text{-}rlx}}_{\epsilon,s_{sec}}(W|\mathsf{SS}(W)) \geq \tilde{m}$.

Unfortunately, we will show below that such a secure sketch implies an error correcting code with approximately $2^{\tilde{m}}$ points that can correct t random errors (see [7, Lemma C.1] for a similar bound on information-theoretic secure sketches). For the Hamming metric, our result essentially matches the bound on information-theoretic secure sketches of [7, Proposition 8.2]. In fact, for the Hamming metric on the uniform distribution, HILL-entropy secure sketches imply information-theoretic ones with similar parameters, and, therefore, the HILL relaxation gives no advantage.

The intuition for building error-correcting codes from HILL-entropy secure sketches is as follows. In order to have $H^{\mathtt{HILL\text{-}rlx}}_{\epsilon,s_{sec}}(W|\mathsf{SS}(W)) \geq \tilde{m}$, there must be a distribution X, Y such that $\tilde{\mathrm{H}}_\infty(X|Y) \geq \tilde{m}$ and (X, Y) is computationally indistinguishable from $(W, \mathsf{SS}(W))$. Sample a sketch $s \leftarrow \mathsf{SS}(W)$. We know that SS followed by Rec likely succeeds on $W|s$ (i.e., $\mathsf{Rec}(w', s) = w$ with high probability for $w \leftarrow W|s$ and $w' \leftarrow B_t(w)$). Consider the following experiment: 1) sample $y \leftarrow Y$, 2) draw $x \leftarrow X|y$ and 3) $x' \leftarrow B_t(x)$. By indistinguishability, $\mathsf{Rec}(x', y) = x$ with high probability. This means we can construct a large set $\mathcal{C}$ from the support of $X|y$. $\mathcal{C}$ will be an error correcting code and Rec an efficient decoder. We can then use standard arguments to turn this code into an information theoretic sketch.

To make this intuition precise, we need an additional technical condition: sampling a random neighbor of a point is efficient.

Definition 3.2. *We say a metric space $(\mathcal{M}, \mathsf{dis})$ is* (s_{neigh}, t)-neighborhood samplable *if there exists a randomized circuit Neigh of size s_{neigh} that for all $t' \leq t$, $\mathsf{Neigh}(w, t')$ outputs a random point at distance t' of w.*

We review the definition of a Shannon code [59]:

Definition 3.3. *Let $\mathcal{C}$ be a set over space $\mathcal{M}$. We say that $\mathcal{C}$ is a* (t, ϵ)-Shannon code *if there exists an efficient procedure* Rec *such that for all $t' \leq t$ and for all $c \in \mathcal{C}$, $\Pr[\mathsf{Rec}(\mathsf{Neigh}(c, t')) \neq c] \leq \epsilon$. To distinguish it from the average-error Shannon code defined below, we will sometimes call it a* maximal-error *Shannon code.*

This is a slightly stronger formulation than usual, in that for every size $t' < t$ we require the code to correct t' random errors.[1] Shannon codes work for all codewords. We can also consider a formulation that works for an "average" codeword.

Definition 3.4. *Let C be a distribution over space $\mathcal{M}$. We say that C is an* (t, ϵ)-average error Shannon code *if there exists an efficient procedure* Rec *such that for all $t' \leq t$ $\Pr_{c \leftarrow C}[\mathsf{Rec}(\mathsf{Neigh}(c, t')) \neq c] \leq \epsilon$.*

An average error Shannon code is one whose average probability of error is bounded by ϵ. See [60, Pages 192-194] for definitions of average and maximal error probability. An average-error Shannon code is convertible to a maximal-error Shannon code with a small loss. We use the following pruning argument from [60, Pages 202-204]:

Lemma 3.5. *Let C be a (t, ϵ)-average error Shannon code with recovery procedure* Rec *such that $\mathrm{H}_\infty(C) \geq k$. There is a set $\mathcal{C}'$ with $|\mathcal{C}'| \geq 2^{k-1}$ that is a $(t, 2t\epsilon)$-(maximal error) Shannon code with recovery procedure* Rec.

Proof. Let C be the (t, ϵ)-average error Shannon code with recovery procedure Rec such that $\mathrm{H}_\infty(C) \geq k$. Then for all $t' \leq t$

$$\sum_{c \in C} \Pr[C = c] \Pr[c' \leftarrow \mathsf{Neigh}(c, t') \wedge \mathsf{Rec}(c') \neq c] \leq \epsilon.$$

For c denote by $\epsilon_{c,t'} = \Pr[c' \leftarrow \mathsf{Neigh}(c, t') \wedge \mathsf{Rec}(c') \neq c]$. Then by Markov's inequality:

$$\Pr_{c \in C}\left[\epsilon_{c,t'} \leq 2t \mathop{\mathbb{E}}_{c \leftarrow C}[\epsilon_{c,t'}]\right] = \Pr_{c \in C}\left[\epsilon_{c,t'} \leq 2t\epsilon\right] \geq 1 - \frac{1}{2t}$$

Let $C'_{t'}$ denote the of set all $c \in C$ where $\epsilon_{c,t'} \leq 2t\epsilon$. Note that $\Pr_{c \leftarrow C}[c \in C'_{t'}] \geq 1 - \frac{1}{2t}$. Define the set

$$C' \stackrel{def}{=} \bigcap_{1 \leq t' \leq t} C'_{t'}.$$

[1] In the standard formulation, the code must correct a random error of size up to t, which may not imply that it can correct a random error of a much smaller size t', because the volume of the ball of size t' may be negligible compared to the volume of the ball of size t. For codes that are monotone (if decoding succeeds on a set of errors, it succeeds on all subsets), these formulations are equivalent. However, we work with an arbitrary recover functionality that is not necessarily monotone.

Since $\forall t', \Pr_{c \leftarrow C}[c \in C'_{t'}] \geq 1 - \frac{1}{2t}$ then $\Pr_{c \leftarrow C}[c \in C'] \geq \frac{1}{2}$. Since $H_\infty(C) \geq k$, we know $|C'| \geq 2^{k-1}$ (otherwise $\Pr_{c \leftarrow C}[c \in C'] = \sum_{c \in C'} \Pr[C = c]$ would be less than $2^{k-1}\frac{1}{2^k} = 1/2$). This completes the proof of Lemma 3.5. □

We can now formalize the intuition above and show that a sketch that retains $\tilde{m}$-bits of relaxed HILL entropy implies a good error correcting code with nearly $2^{\tilde{m}}$ points.

Theorem 3.6. *Let $(\mathcal{M}, \mathsf{dis})$ be a metric space that is (s_{neigh}, t)-neighborhood samplable. Let $(\mathsf{SS}, \mathsf{Rec})$ be an HILL-entropy $(\mathcal{M}, m, \tilde{m}, t)$-secure sketch that is (ϵ, s_{sec})-secure with error δ. Let s_{rec} denote the size of the circuit that computes Rec. If $s_{sec} \geq t(s_{neigh} + s_{rec})$, then there exists a value s and a set $\mathcal{C}$ with $|\mathcal{C}| \geq 2^{\tilde{m}-2}$ that is a $(t, 4t(\epsilon + t\delta))$-Shannon code with recovery procedure $\mathsf{Rec}(\cdot, s)$.*

Proof. Let W be an arbitrary distribution of min-entropy m. Let (X, Y) be a joint distribution such that $\tilde{\mathrm{H}}_\infty(X|Y) \geq \tilde{m}$ and

$$\delta^{\mathcal{D}_{s_{sec}}}((W, \mathsf{SS}(W)), (X, Y)) \leq \epsilon\,,$$

where $s_{sec} \geq t(s_{neigh} + s_{rec})$. One such (X, Y) must exist by the definition of conditional HILL entropy. Define D as:

1. Input $w \in \mathcal{M}, z \in \{0,1\}^*, t$.
2. For all $1 \leq t' \leq t$:
   $w' \leftarrow \mathsf{Neigh}(w, t')$.
   If $\mathsf{Rec}(w', z) \neq w$ output 0.
3. Output 1.

By correctness of the sketch $\Pr[D(W, \mathsf{SS}(W)) = 1] \geq 1 - t\delta$. Since

$$\delta^D((W, \mathsf{SS}(W)), (X, Y)) \leq \epsilon,$$

we know $\Pr[D(X, Y) = 1] \geq 1 - (\epsilon + t\delta)$. Let X_y denote the random variable $X|Y = y$. By Markov's inequality, there exists a set S_Y such that $\Pr[Y \in S_Y] \geq 1/2$ and for all $y \in S_Y$, $\Pr[D(X_y, y) = 1] \geq 1 - 2(\epsilon + t\delta)$.

Because $\tilde{\mathrm{H}}_\infty(X|Y) \geq \tilde{m}$, we know that $\mathbb{E}_{y \leftarrow Y} \max_x \Pr[X_y = x] \leq 2^{\tilde{m}}$. Applying Markov's inequality to the random variable $\max_x \Pr[X_y = x]$, there exists a set S'_Y such that $\Pr[y \in S'_Y] > 1/2$, and for all $y \in S'_Y$, $\mathrm{H}_\infty(X_y) \geq \tilde{m} - 1$ (we can use the strict version of Markov's inequality here, because the random variable $\max_x \Pr[X_y = x]$ is positive). Fix one value $y \in S_Y \cap S'_Y$ (which exists because the sum of probabilities of S_Y and S'_Y is greater than 1). Thus, for all such that $t', 1 \leq t' \leq t$,

$$\Pr_{x \leftarrow X_y}[x' \leftarrow \mathsf{Neigh}(x, t') \wedge \mathsf{Rec}(x', y) = x] \geq 1 - 2(\epsilon + t\delta).$$

Thus, X_y is a $(t, 2(\epsilon + t\delta))$-average error Shannon code with recovery $\mathsf{Rec}(\cdot, y)$ and $2^{\tilde{m}-1}$ points. The statement of the theorem follows by application of Lemma 3.5. □

For the Hamming metric, any Shannon code (as defined in Definition 3.3) can be converted into an information-theoretic secure sketch (as described in [7, Section 8.2] and references therein). The idea is to use the code offset construction, and convert worst-case errors to random errors by randomizing the order of the symbols of w first, via a randomly chosen permutation π (which becomes part of the sketch and is applied to w' during Rec). The formal statement of this result can be expressed in the following Lemma (which is implicit in [7, Section 8.2]).

Lemma 3.7. *For an alphabet $\mathcal{Z}$, let $\mathcal{C}$ over $\mathcal{Z}^n$ be a (t, δ)-maximal error Shannon code. Then there exists a $(\mathcal{Z}^n, m, m - (n \log |\mathcal{Z}| - \log |\mathcal{C}|), t)$ secure sketch with error δ for the Hamming metric over $\mathcal{Z}^n$.*

Putting together Theorem 3.6 and Lemma 3.7 means that for the Hamming metric a HILL-entropy secure sketch implies an information-theoretic one:

Corollary 3.8. *Let $\mathcal{Z}$ be an alphabet. Let $(\mathsf{SS}, \mathsf{Rec})$ be an (ϵ, s_{sec})-HILL-entropy $(\mathcal{Z}^n, m, \tilde{m}, t)$-secure sketch with error δ for the Hamming metric over $\mathcal{Z}^n$, with Rec' of circuit size s_{rec}. If $s_{sec} \geq t(s_{rec} + n \log |\mathcal{Z}|)$, then for any m' there exists a $(\mathcal{Z}^n, m', m' - (n \log |\mathcal{Z}| - \tilde{m}) - 2, t)$ (information-theoretic) secure sketch with error $4t(\epsilon + t\delta)$.*

In Corollary 3.8 we make no claim about the efficiency of the resulting $(\mathsf{SS}, \mathsf{Rec})$, because the proof of Theorem 3.6 is not constructive.

The conference version of this work contained an upper bound on the unpredictability of a computational secure sketch [1, Theorem 2]. Roughly, the result said that the sketch had unpredictability at most $n \log |\mathcal{Z}| - \log |B_t(\cdot)|$. This bound was based on a simple adversary that guesses a random point. Fuller, Reyzin, and Smith [15, Definition 3] introduced fuzzy min-entropy which measures an adversaries success when provided with the functionality of a secure sketch. Fuzzy min-entropy is formally defined as

$$\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(W) \stackrel{def}{=} -\log \left(\max_{w'} \Pr[W \in B_t(w')] \right).$$

Fuzzy min-entropy is maximized for the uniform distribution where $\mathrm{H}^{\mathtt{fuzz}}_{t,\infty}(U_n) = n \log |\mathcal{Z}| - \log |B_t(\cdot)|$. This matches the bound of [1, Theorem 2] for the uniform distribution but provides tighter bounds for other distributions. Thus, the prior unpredictability bound has been strictly improved and is omitted.

Avoiding the Bound. The bound arises because Rec must function as decoder for any indistinguishable distribution. The lower bound is strongest for high entropy sources. If a source contains only codewords (of an error correcting code), no sketch is necessary. This same situation occurs when considering lower bounds for information-theoretic sketches [7, Appendix C]. In the Introduction, we discuss whether one could construct secure sketches that retains other forms of pseudoentropy.

Fuzzy extractors are not required to output the same point; they can instead output a consistent r. If some efficient algorithm can take the output of the

reproduce algorithm Rep and efficiently transform it back to w, the lower bound applies. This means that we need to consider constructions that are hard to invert (either information-theoretically or computationally). This intuition was formalized by Yasunaga and Yuzawa [53] who show a similar result if Rep is invertible to some potential w^*.

A natural way to avoid this result is if Rec outputs a fresh random variable. Such an algorithm is called a computational fuzzy conductor. See Kanukurthi and Reyzin [61] for the definition of a fuzzy conductor and Canetti et al. [41] for the computational version. The definition replaces the pseudorandomness condition in Definition 2.2 with a HILL entropy requirement.

Our construction (in Section 4) has pseudorandom output and immediately satisfies definition of a computational fuzzy extractor (Definition 2.2). Canetti et al. first use a conductor and then a computational extractor [41, Construction 2].

4. A Computational Fuzzy Extractor based on LWE

As stated in the introduction, our construction of a computational fuzzy extractor treats the input w (drawn from the source W) as the noise term added to a codeword of a random linear code. Thus, the security of our construction depends on the distribution given by W. In this section we consider a uniform source W; we consider other distributions in Section 5. Our construction uses the code-offset construction [31], [7, Section 5] instantiated with a random linear code over a finite field $\mathbb{Z}_q$. Let Decode_t be an algorithm that decodes a random linear code with at most t errors. We present such an algorithm in Section 4.3.

Construction 4.1. *Let n be a security parameter and let $m \geq n$. Let q be a prime. Define* Gen, Rep *as follows:*

Gen

1. *Input: $w \leftarrow W$ (W is distributed over $\mathbb{Z}_q^m$).*
2. *Sample $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{x} \in \mathbb{Z}_q^n$ uniformly.*
3. *Compute $p = (\mathbf{A}, \mathbf{A}\mathbf{x} + w)$ and $r = \mathbf{x}_{1,\ldots,n/2}$.*
4. *Output (r, p).*

Rep

1. *Input: (w', p) (where $d(w, w') \leq t$).*
2. *Parse p as $(\mathbf{A}, \mathbf{c})$; let $\mathbf{b} = \mathbf{c} - w'$.*
3. *Let $x = \mathsf{Decode}_t(\mathbf{A}, \mathbf{b})$.*
4. *Output $r = x_{1,\ldots,n/2}$.*

We know that decoding a random linear code is NP-hard [39]. For this construction to be secure, it should be computationally hard to decode a random linear code with errors distributed according to W. In fact, we need more: we actually need to show that $X_{1,\ldots,n/2}$ are hardcore bits. That is, we need

$$\delta^{\mathcal{D}_{ssec}}((X_{1,\ldots,n/2}, P), (U_{n/2 \log q}, P)) \leq \epsilon.$$

Furthermore, this construction is only useful if Decode_t can be efficiently implemented.

The rest of this section is devoted to making these statements precise. We first review some properties of random linear codes in Section 4.1. We then describe the LWE problem and the security of our construction in Section 4.2. We describe one possible polynomial-time Decode_t (which corrects more errors than is possible by exhaustive search) in Section 4.3. In Section 4.4, we describe parameter settings that allow us to extract as many bits as the input entropy, resulting in a lossless construction. In Section 4.5, we compare Construction 4.1 to using a sketch-and-extract approach (Lemma 2.5) instantiated with a computational extractor.

4.1. Properties of Random Linear Codes

For correctness of our construction, we need a random linear code to have high distance with overwhelming probability. We will use the q-ary entropy function, denoted $H_q(x)$ and defined as $H_q(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x)$. Note that $H_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$. In the region $[0, \frac{1}{2}]$ for any value $q' \geq q$, $H_{q'}(x) \leq H_q(x)$. The following theorem is standard in coding theory:

Theorem 4.2. *[62, Theorem 8] For prime $q, \delta \in [0, 1-1/q), 0 < \epsilon < 1 - H_q(\delta)$ and sufficiently large m, the following holds for $n = \lceil (1 - H_q(\delta) - \epsilon)m \rceil$. If $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ 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 $1 - e^{-\Omega(m)}$.*

Our setting is the case where $m = poly(n) \geq 2n$ and $\delta = O(\frac{\log n}{n})$. This setting of parameters satisfies Theorem 4.2:

Corollary 4.3. *Let n be a parameter and let $m = \mathtt{poly}(n) \geq 2n$. Let q be a prime and $\tau = O(\frac{m}{n}\log n)$. For large enough values of n, when $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is drawn uniformly, the code generated by $\mathbf{A}$ has distance at least τ (and relative distance $\tau/m = \delta = O(\log n/n)$) with probability at least $1 - e^{-\Omega(m)} \geq 1 - e^{-\Omega(n)}$.*

Proof. Let c be some constant. Let $\delta = \tau/m = \frac{c\log n}{n}$. We show the corollary for the case when $m = 2n$ (increasing the size of m only increases the relative distance). It suffices to show that for sufficiently large n, there exists $\epsilon > 0$ where $1 - H_q(\frac{c\log n}{n}) - \epsilon = 1/2$ or equivalently that $H_q(\frac{c\log n}{m}) < 1/2$, as then setting $\epsilon = 1/2 - H_q(\frac{c\log n}{n})$ satisfies Theorem 4.2. For sufficiently large n:

- $\frac{c\log n}{n} < 1/2$, so we can work with the binary entropy function H_2.
- $\frac{c\log n}{n} < .1 < 1/2$ and thus $H_q(\frac{c\log n}{n}) < H_q(.1)$.

Putting these statements together, for large enough n, $H_q(\frac{c\log n}{n}) < H_q(.1) < H_2(.1) < 1/2$ as desired. This completes the proof. □

We also need that random matrices are full rank with high probability. We use the following claim (techniques from Cooper [63]):

Lemma 4.4. *Let $q \geq 2$ be a prime. Let $\alpha, \beta \in \mathbb{Z}^+$ and let $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{\alpha \times (\alpha+\beta)}$ be uniform. Then $\Pr[\mathtt{rank}(\mathbf{S}) = \alpha] > 1 - q^{-\beta}$.*

Proof. Let p_i be the probability that the ith row is linearly dependent on the previous $i-1$ rows. By the union bound, the probability that α rows are linearly dependent is bounded by $\sum_{i=1}^{\alpha} p_i$. Since $i-1$ rows can span a space of size at most q^{i-1}, the probability p_i that a randomly chosen ith row is in that space is at most $q^{i-1}/q^{\alpha+\beta}$. So

$$\Pr[\mathtt{rank}(\mathbf{S}) < \alpha] = \sum_{i=1}^{\alpha} \frac{q^{i-1}}{q^{\alpha+\beta}} = \frac{q^{\alpha}-1}{q-1}\frac{1}{q^{\alpha+\beta}} < q^{-\beta}.$$

□

4.2. Security of Construction 4.1

The LWE problem was introduced by Regev [32, 33, 34] as a generalization of "learning parity with noise." For a thorough discussion of the LWE problem and related lattice problems (which we do not define here) see [32, 34]. We now recall the decisional version of the problem.

Definition 4.5 (Decisional LWE)**.** *Let n be a security parameter. Let $m = m(n) = \mathtt{poly}(n)$ be an integer and $q = q(n) = \mathtt{poly}(n)$ be a prime*[2]*. Let $\mathbf{A}$ be the uniform distribution over $\mathbb{Z}_q^{m \times n}$, X be the uniform distribution over $\mathbb{Z}_q^n$, and χ be an arbitrary distribution on $\mathbb{Z}_q^m$. The decisional version of the* LWE *problem, denoted* dist-LWE$_{n,m,q,\chi}$*, is to distinguish the LWE distribution $(\mathbf{A}, \mathbf{A}X + \chi)$ from the uniform distribution over $(\mathbb{Z}_q^{m \times n}, \mathbb{Z}_q^m)$.*

We say that dist-LWE$_{n,m,q,\chi}$ *is (ϵ, s_{sec})-secure if any (probabilistic) distinguisher of size s_{sec} can distinguish the* LWE *instances from uniform with advantage no more than ϵ.*

Regev [32, 34] shows that dist-LWE$_{n,m,q,\chi}$ can be reduced to approximately solving lattice problems known as GAPSVP and SIVP when the distribution χ of errors is Gaussian. Let $\bar{\Psi}_\rho$ be the discretized Gaussian distribution with variance $(\rho q)^2/2\pi$, where $\rho \in (0,1)$ with $\rho q > 2\sqrt{n}$. If GAPSVP and SIVP are hard to approximate (on lattices of dimension n) within polynomial factors for quantum algorithms, then dist-LWE$_{n,m,q,\bar{\Psi}_\rho^m}$ is secure. (Later, Peikert [64] together with Brakerski et al. [65] showed security of LWE based on hardness of approximating lattice problems for classical, rather than quantum, algorithms.)

The above formulation of LWE requires the error term to come from the discretized Gaussian distribution. For our purposes, we use a different formulation, due to Döttling and Müller-Quade [35], which shows security of LWE when

[2]Unlike in common formulations of LWE, where q can be any integer, we need q to be prime for decoding.

errors come from the uniform distribution over a small interval, under the same assumptions.[3] This allows us to directly encode w as the error term in an LWE problem by splitting it into m blocks. The size of these blocks is dictated by the following result of Döttling and Müller-Quade.

Theorem 4.6. *[35, Theorem 6] Let n be a security parameter. Let $q = q(n) = \mathtt{poly}(n)$ be a prime and $m = m(n) = \mathtt{poly}(n)$ be an integer with $m \geq 3n$. Let $\sigma \in (0,1)$ be an arbitrarily small constant and let $\rho = \rho(n) \in (0, 1/10)$ be such that $\rho q \geq 2n^{1/2+\sigma}m$. Define the value $\alpha = \rho/(mn^{\sigma})$. Then define χ as the uniform distribution over $[-\rho q, \rho q]^m$. If $\mathsf{dist\text{-}LWE}_{(n,m,q,\bar{\Psi}^m_\alpha)}$ is secure then $\mathsf{dist\text{-}LWE}_{(n,m,q,\chi)}$ is secure.*

To extract pseudorandom bits, we use a result of Akavia, Goldwasser, and Vaikuntanathan [38] to show that X has many simultaneously hardcore bits. The result says that if $\mathsf{dist\text{-}LWE}_{(n-k,m,q,\chi)}$ is secure then any k coordinates of X in a $\mathsf{dist\text{-}LWE}_{(n,m,q,\chi)}$ instance are hardcore. We state their result for a general error distribution (noting that their proof does not depend on the error distribution).

Lemma 4.7. *[38, Lemma 2] Let χ be a distribution over $\mathcal{Z}_q^m$. Suppose that $\mathsf{dist\text{-}LWE}_{(n-k,m,q,\chi)}$ is (ϵ, s_{sec}) secure, then*

$$\delta^{\mathcal{D}_{s_{sec}'}}((X_{1,\ldots,k}, \mathbf{A}, \mathbf{A}X + \chi), (U, \mathbf{A}, \mathbf{A}X + \chi)) \leq \epsilon\,,$$

where U denotes the uniform distribution over $\mathbb{Z}_q^k$, $\mathbf{A}$ denotes the uniform distribution over $\mathbb{Z}_q^{m\times n}$, X denotes the uniform distribution over $\mathbb{Z}_q^n$, $X_{1,\ldots,k}$ denote the first k coordinates of x, and $s'_{sec} \approx s_{sec} - n^3$.

The security of Construction 4.1 follows from Theorem 4.6 and Lemma 4.7 when parameters are set appropriately (see Theorem 4.11), because we use the hardcore bits of X as our key.

Improved Uniform LWE reductions. Subsequent to the conference version of this work [1], Bogdanov et al. [36] and Bai et al. [37] presented additional reductions from uniform LWE to standard LWE with a discretized Gaussian error distribution. For the purposes of this work, the most important change is that they require a smaller uniform interval. Bogdanov et al.'s result requires $\rho = \Omega(m\alpha/\sqrt{\log n})$ and a multiplicative dimensionality loss of $\log n$ (in the underlying Gaussian LWE). Bai et al. has a slighly worse $\rho = \Omega(m\alpha/\log n)$ but with no dimensionality loss. Bai et al's main theorem is the following:

Theorem 4.8. *[37, Theorem 5.1] Let n be a security parameter and let m be a positive integer. Let $\alpha, \rho > 0$ be real numbers with $\rho = \Omega(m\alpha/\log n)$. Let q be some prime positive integer. Furthermore, suppose that*

[3] Micciancio and Peikert provide a similar formulation in [66]. The result of Döttling and Müller-Quade provides better parameters for our setting.

- $m > (n \log q)/(\log(\alpha+\rho)^{-1})$ *and*
- $q = \mathtt{poly}(m, n)$.

Let χ *be the uniform distribution over* $[-\theta q, \theta q]^m$ *where* $\theta = \frac{1}{q}\lfloor q\rho \rfloor$. *Then if* $\mathsf{dist\text{-}LWE}_{(n,m,q,\bar{\Psi}^m_\alpha)}$ *is secure then* $\mathsf{dist\text{-}LWE}_{(n,m,q,\chi)}$ *is secure.*

The main differences between the result of Bai et al. [37] and Döttling and Müller-Quade [35] are two fold (ignoring the difference between ρ and θ).

1. The noise magnitude required in Döttling and Müller-Quade is $\rho \geq n^{\Omega(1)} m\alpha$ while in Bai et al. it is $\rho \geq (m\alpha)/\log n$.
2. The required number of samples in Bai et al. is $m \geq \Omega(n \log q/\log(\alpha+\rho)^{-1})$ while in Döttling and Müller-Quade it is $m \geq 3n$.

In this work, our goal is to extract a key that is as long as the input source. In standard formulations (with $\alpha = 2\sqrt{n}/q$), substituting parameters, one has that:

$$\begin{aligned}\log(\alpha+\rho)^{-1} &= \log \frac{q}{q\alpha + q\rho} = \log\left(\frac{q}{2\sqrt{n} + m\sqrt{n}/\log n}\right) \\ &= \log q - \log(2\sqrt{n} + m\sqrt{n}/\log n).\end{aligned}$$

Thus, the term

$$\frac{\log q}{\log(\alpha+\rho)^{-1}} = \frac{\log q}{\log q - \log\left(2\sqrt{n} + \frac{m\sqrt{n}}{\log n}\right)}.$$

Suppose that $q = \Omega(n^c)$ for some $c > 3/2$. Then it is possible for above equation to be bounded by some constant and thus $m = \Omega(n)$. Verifying this is slightly complex as there is a circular relationship: m is lower bounded by a quantity including ρ which is lower bounded by a quantity including m. As long as $q = \Omega(n^c)$ for some $c > 3/2$, this system is satisfiable for $m = \Omega(n)$. This means for large enough q, one can reduce the required noise from size $\rho q = \Omega(n^{3/2+o(1)})$ to $\rho q = n^{3/2}/\log n$. Appendix A and Appendix B present parameters using Theorem 4.6 due to Döttling and Müller-Quade. This parameterization can be slightly improved using Bai et al.'s result. However, this improvement is small compared to how parameters depend on the number of errors to be corrected and the desired running time.

4.3. Efficiency of Construction 4.1

Construction 4.1 is useful only if Decode_t can be efficiently implemented. We need a decoding algorithm for a random linear code with t errors that runs in polynomial-time. We present a simple Decode_t that runs in polynomial-time and can correct t errors. Note that correcting $t = \Theta(\log n)$ errors corresponds to correcting at least $\binom{n}{t}(q-1)^t$ error patterns which is superpolynomial.

Construction 4.9. *We consider a setting of (n, m, q, χ) where $m \geq 3n$. We describe* Decode_t*:*

1. *Input* $\mathbf{A}, \mathbf{b} = \mathbf{Ax} + w - w'$
2. *Randomly select rows without replacement* $i_1, ..., i_{2n} \leftarrow [1, m]$.
3. *Restrict* $\mathbf{A}, \mathbf{b}$ *to rows* $i_1, ..., i_{2n}$*; denote these* $\mathbf{A}_{i_1,...,i_{2n}}, \mathbf{b}_{i_1,...,i_{2n}}$.
4. *Find* n *rows of* $\mathbf{A}_{i_1,...,i_{2n}}$ *that are linearly independent. If no such rows exist, output* $\perp$ *and stop.*
5. *Denote by* $\mathbf{A}', \mathbf{b}'$ *the restriction of* $\mathbf{A}_{i_1,...,i_{2n}}, \mathbf{b}_{i_1,...,i_{2n}}$ *(respectively) to these rows. Compute* $\mathbf{x}' = (\mathbf{A}')^{-1}\mathbf{b}'$.
6. *If* $\mathbf{b} - \mathbf{Ax}'$ *has more than* t *nonzero coordinates, go to step (2).*
7. *Output* $\mathbf{x}'$.

The algorithm is an information set decoding algorithm. Lee and Brickel [67], Berman and Karpinski [68] and Peters [69] optimize the above algorithm by 1) selecting rows in a a structured way to improve probability of linear independence, 2) swapping out rows rather than starting from scratch, and 3) saving partial Gaussian elimination results (for computing the inverse). These techniques improve concrete efficiency but do not asymptotically improve the number of errors that can be corrected in polynomial time. As noted by [70, 71], when $m = \Theta(n)$, any algorithm that operates generically on the elements of $\mathbf{A}X + (w - w')$ can be efficient only when $t = O(\log n)$, establishing a hurdle to asymptotic improvement.

Each step is computable in time $O(n^3)$. For Decode_t to be efficient, we need t to be small enough so that with probability at least $\frac{1}{\mathtt{poly}(n)}$, none of the $2n$ rows selected in step 2 have errors (i.e., so that w and w' agree on those rows). If this happens, and $\mathbf{A}_{i_1,...,i_{2n}}$ has rank n (which is highly likely), then $\mathbf{x}' = \mathbf{x}$, and the algorithm terminates. However, we also need to ensure correctness: we need to make sure that if $\mathbf{x}' \neq \mathbf{x}$, we detect it in step 6. This detection will happen if $\mathbf{b} - \mathbf{Ax}' = \mathbf{A}(\mathbf{x} - \mathbf{x}') + (w - w')$ has more than t nonzero coordinates. It suffices to ensure that $\mathbf{A}(\mathbf{x} - \mathbf{x}')$ has at least $2t + 1$ nonzero coordinates (because at most t of those can be zeroed out by $w - w'$), which happens whenever the code generated by $\mathbf{A}$ has distance $2t + 1$.

Setting $t = O(\frac{m}{n} \log n)$ is sufficient to ensure efficiency. Random linear codes have sufficient distance with probability $1 - e^{-\Omega(n)}$ (the exact statement is in Corollary 4.3), so this also ensures correctness. The formal statement is below:

Lemma 4.10. *Let d be a positive constant and assume that* $\mathsf{dis}(W, W') \leq t$ *where* $t \leq d(\frac{m}{n} - 2) \log n$*. Then* Decode_t *runs in expected time* $O(n^{4d+3})$ *operations in* $\mathbb{Z}_q$ *(this expectation is over the choice of random coins of* Decode_t*, regardless of the input, as long as* $\mathsf{dis}(w, w') \leq t$*). It outputs X with probability* $1 - e^{-\Omega(n)}$ *(this probability is over the choice of the random matrix* $\mathbf{A}$ *and random choices made by* Decode_t*).*

Proof. Note that Decode_t will stop if w and w' agree on all the rows selected in Step 2 (it may also stop for other reasons—namely, in step 4; but we do not use this fact to bound the expected running time). The probability of each selected row having an error is at most $\frac{t}{m-i}$ where i is the number of rows already selected. That is,

$$\begin{aligned}
\Pr[i_1, ..., i_{2n} \text{ have no errors}] &\geq \prod_{i=0}^{2n-1} \left(1 - \frac{t}{m-i}\right) \geq \prod_{i=0}^{2n-1} \left(1 - \frac{d\left(\frac{m}{n} - 2\right)\log n}{m-i}\right) \\
&\geq \prod_{i=0}^{2n-1} \left(1 - \frac{d\log n}{n}\left(\frac{m-2n}{m-i}\right)\right) \geq \prod_{i=0}^{2n-1} \left(1 - \frac{d\log n}{n}\right) \\
&= \left(1 - \frac{d\log n}{n}\right)^{2n} = \left(\left(1 - \frac{d\log n}{n}\right)^{\frac{n}{d\log n}}\right)^{2d\log n} \\
&\geq \frac{1}{4^{2d\log n}} = \frac{1}{n^{4d}}\,.
\end{aligned}$$

(The second-to-last step holds as long as $n \geq 2d\log n$.) Because at each iteration, we select $2n$ rows independently at random, the expected number of iterations is at most n^{4d}; each iteration takes $O(n^3)$ operations in $\mathbb{Z}_q$, which gives us the expected running time bound. The probability that Decode_t outputs $\perp$ is bounded by

$$\begin{aligned}
&\Pr[\mathsf{Decode}_t \to \perp] \\
&\quad\leq \sum_{j=1}^{\infty} \Pr[\mathsf{Decode}_t \to \perp \text{ in } j\text{th iteration of step 4}] \\
&\quad= \sum_{j=1}^{\infty} \Pr[\mathsf{Decode}_t \text{ continues to } j \text{ iters.} \wedge \mathtt{rank}(\mathbf{A}_{i_1,...,i_{2n}}) < n] \\
&\quad\leq \sum_{j=1}^{\infty} \Pr[i_1, ..., i_{2n} \text{ had errors } j-1 \text{ times} \wedge \mathtt{rank}(\mathbf{A}_{i_1,...,i_{2n}}) < n] \\
&\quad= \sum_{j=1}^{\infty} \Pr[i_1, ..., i_{2n} \text{ had errors } j-1 \text{ times}] \cdot \Pr[\mathtt{rank}(\mathbf{A}_{i_1,...,i_{2n}}) < n] \\
&\quad\leq \sum_{j=1}^{\infty} \left(1 - \frac{1}{n^{4d}}\right)^{j-1} \cdot q^{-n} \\
&\quad= n^{4d} e^{-\Omega(n)} \\
&\quad= e^{-\Omega(n)}\,.
\end{aligned}$$

The fourth line from the bottom follows from the fact that the locations of the errors are assumed to be independent of the sketch, and therefore independent of the matrix $\mathbf{A}$. The third line from the bottom follows from Claim 4.4 when $\beta = n$; note that, because we use the union bound and evaluate the probability

separately for each value of j, we can treat $\mathbf{A}_{i_1,\ldots,i_{2n}}$ as a randomly chosen $2n \times n$ matrix, ignoring the fact that these matrices are correlated.

We claim that if the code generated by $\mathbf{A}$ has distance at least $2t+1$, then Decode_t will output $\perp$ or the correct $\mathbf{x}' = \mathbf{x}$. Indeed, suppose $\mathbf{x}' \neq \mathbf{x}$. Since $\mathbf{A}(\mathbf{x} - \mathbf{x}')$ has at least $2t+1$ nonzero coordinates by the minimum distance of the code generated by $\mathbf{A}$, and at most t of those can be zeroed out by the addition of $w - w'$, such an $\mathbf{x}'$ will not pass Step 6.

The probability that the code generated by $\mathbf{A}$ has distance lower than $2t+1$ is at most $e^{-\Omega(n)}$ (see Corollary 4.3), and the probability of outputting $\perp$ is also $e^{-\Omega(n)}$ (computed above). This gives the correctness bound for Decode_t. □

4.4. Lossless Computational Fuzzy Extractor

We now state a setting of parameters that yields a lossless construction. We split W into m blocks each of size $\log \rho q$ (from Theorem 4.6) with

$$|W| = \mathrm{H}_\infty(W) = m \log \rho q.$$

Our key consists of hardcore bits of X (namely, coordinates $X_{1,\ldots,k}$) and is of size $k \log q$ (from Lemma 4.7). Thus, to get $|W| = |X_{1,\ldots,k}|$ we need $k \log q = m \log \rho q$. While the vector w is of higher dimension than the vector X, each coordinate of w is sampled using fewer bits than each coordinate of X. Thus, by increasing the size of q (while keeping ρq fixed) we can set $k \log q = m \log \rho q$, yielding a key of the same size as our source. The formal statement is below.

Theorem 4.11. *Let n be a security parameter and let the number of errors $t = c \log n$ for some positive constant c. Let d be a positive constant (giving us a tradeoff between running time of* Rep *and $|w|$). Consider the Hamming metric over the alphabet $\mathcal{Z} = [-2^{b-1}, 2^{b-1}]$, where $b = \log 2(c/d+2)n^2 = O(\log n)$. Let W be uniform over $\mathcal{M} = \mathcal{Z}^m$, where $m = (c/d+2)n = O(n)$. There is a setting of $q = \mathtt{poly}(n, m)$ such that if* $\mathsf{dist\text{-}LWE}_{(n,m,q,\mathcal{Z}^m)}$ *is (ϵ, s_{sec})-secure, then Construction 4.1 is a $(\mathcal{M}, W, m \log |\mathcal{Z}|, t)$-computational fuzzy extractor that is (ϵ, s_{sec})-hard with error $\delta = e^{-\Omega(n)}$. The generate procedure* Gen *takes $O(n^2)$ operations over $\mathbb{Z}_q$, and the reproduce procedure* Rep *takes expected time $O(n^{4d+3})$ operations over $\mathbb{Z}_q$.*

Proof. Security follows by combining Theorem 4.6 and Lemma 4.7; efficiency follows by Lemma 4.10. For a more detailed explanation of the various parameters and constraints see Appendix A. □

4.5. Comparison with computational-extractor-based constructions

As mentioned in the Introduction, an alternative approach to building a computational fuzzy extractor is to use a computational extractor (e.g., [19, 20, 21]) in place of the information-theoretic extractor in the sketch-and-extract construction. We will call this approach *sketch-and-comp-extract*. (A simple example of a computational extractor is a pseudorandom generator applied to the output of an information-theoretic extractor; note that LWE-based pseudorandom generators exist [72, 73].)

This approach (specifically, its analysis via Lemma 2.5) works as long as the amount of entropy $\tilde{m}$ of w conditioned on the sketch s remains high enough to run a computational extractor. However, as discussed in Section 3, $\tilde{m}$ decreases with the error parameter t due to coding bounds. There are practical sources, such as the iris [74, Section 5], where after sketch losses there is too little entropy left to run a computational extractor once s is known.

In contrast, our approach does not require the entropy of w conditioned on $p = (\mathbf{A}, \mathbf{A}X + w)$ to be high enough for a computational extractor. The key difference in our approach is that instead of extracting from w, we hide secret randomness using w. Computational extractors are not allowed to have secret randomness [19, Definition 3].

Additionally, our construction has a unique feature: security need not depend on the error-tolerance t. Instead the time to recover is determined by t.

Unfortunately, LWE parameter sizes require relatively long w and our construction demonstrates a low error tolerance. However, we believe the conceptual framework can lead to better constructions. As an example, Herder et al. [48] achieve practical error correction using a random linear code. Their decode algorithm uses extra information in the source to pinpoint dimensions unlikely to have an error (called *confidence information*).

5. Extending to Nonuniform Sources

We note that Construction 4.1 is secure whenever the source W is an *LWE admissible distribution*, meaning that using W as the error vector in LWE makes decoding/distinguishing computationally hard. (The instance has to be sufficiently hard for there to be a large number of hardcore bits.) Towards this end, we show hardness of LWE when a small number of dimensions of the error vector are fixed. We recall the notion of a symbol fixing source (from [40, Definition 2.3]):

Definition 5.1. *Let $W = (W_1, ..., W_{m+\alpha})$ be a distribution where each W_i takes values over an alphabet $\mathcal{Z}$. We say that it is a $(m + \alpha, m, |\mathcal{Z}|)$* symbol fixing source *if for α indices $i_1, \ldots, i_\alpha$, the symbols W_{i_α} are fixed, and the remaining m symbols are chosen uniformly at random. Note that $H_\infty(W) = m \log |\mathcal{Z}|$.*

Symbol-fixing sources are a very structured class of distributions. However, extending Construction 4.1 to such a class is not obvious. Although symbol-fixing sources are deterministically extractable [40], we cannot first run a deterministic extractor before using Construction 4.1. This is because we need to preserve distance between w and w' and an extractor must not preserve distance between input points. Instead, we directly show the security of LWE with symbol-fixing sources. The following theorem states that dist-LWE with symbol-fixing sources is implied by the standard dist-LWE (but for n and m reduced by the amount of fixed symbols).

Theorem 5.2. *Let n be a security parameter, m, α, and q be polynomial in n, where q is a prime, and $\beta \in \mathbb{Z}^+$ be such that $q^{-\beta}$ is negligible in n. Let U denote the uniform distribution over $\mathcal{Z}^m$ for an alphabet $\mathcal{Z} \subset \mathbb{Z}_q$, and let W denote an $(m+\alpha, m, |\mathcal{Z}|)$ symbol fixing source over $\mathcal{Z}^{m+\alpha}$. If $\mathsf{dist\text{-}LWE}_{n,m,q,U}$ is secure, then $\mathsf{dist\text{-}LWE}_{n+\alpha+\beta,m+\alpha,q,W}$ is also secure.*

Theorem 5.2 also holds for an arbitrary error distribution (not just uniform errors) in the following sense. Let χ' be an arbitrary error distribution. Define χ as the distribution where m dimensions are sampled according to χ' and the remaining dimensions have some fixed error. Then, security of $\mathsf{dist\text{-}LWE}_{n,m,q,\chi'}$ implies security of $\mathsf{dist\text{-}LWE}_{n+\alpha+\beta,m+\alpha,q,\chi}$. We show this stronger version of the theorem below.

The intuition for this result is as follows. Providing a single sample with no error "fixes" at most a single variable. Thus, if there are significantly more variables than samples with no error, search LWE should still be hard. We are able to show a stronger result that $\mathsf{dist\text{-}LWE}$ is still hard. The nontrivial part of the reduction is using the additional $\alpha+\beta$ variables to "explain" a random value for the last α samples, without knowing the other variables. The β parameter is the slack needed to ensure that the "free" variables have the influence on the last α samples.

Proof of Theorem 5.2. We assume that all of the fixed blocks are located at the end and their fixed value is 0. If the blocks are fixed to some other value, the reduction is essentially the same. In the reduction, the distinguisher is allowed to depend on the source and can know the positions of the fixed blocks and their values. For a matrix $\mathbf{A}$ we will denote the i-th row by $\mathbf{a}_i$. For a set T of column indices, we denote by $\mathbf{A}_T$ the restriction of the matrix $\mathbf{A}$ to the columns contained in T. Similarly, for a vector $\mathbf{x}$ we denote by $\mathbf{x}_T$ the restriction of $\mathbf{x}$ to the variables contained in T. We use similar notations for the complement of T, denoted T^c. For a matrix or vector we use T to denote the transpose. We use i as an index into matrix rows and the error vector and j as an index into matrix columns and the solution vector.

Let n be a security parameter, and m, q, α be polynomial in n. Let β be such that $q^{-\beta}$ is negligible in n. All operations are computed modulo q, and we omit " mod q" notation. Let χ' be some error distribution over $\mathbb{Z}_q^m$ and let χ over $\mathbb{Z}_q^{m+n}$ be defined by sampling χ' to obtain values on dimensions $1, ..., m$ and then appending α 0s.

Let D be a distinguisher that breaks $\mathsf{dist\text{-}LWE}_{(m+\alpha),(n+\alpha+\beta),q,\chi}$ with advantage $\epsilon > 1/\mathtt{poly}(n)$. Let $\mathbf{A}$ denote the uniform distribution over $\mathbb{Z}_q^{(m+\alpha)\times(n+\alpha+\beta)}$, X denote the uniform distribution over $\mathbb{Z}_q^{(n+\alpha+\beta)}$, and U denote the uniform distribution over $\mathbb{Z}_q^{m+\alpha}$. Then

$$|\Pr[D(\mathbf{A}, \mathbf{A}X + \chi) = 1] - \Pr[D(\mathbf{A}, U) = 1]| > \epsilon.$$

We build a distinguisher that breaks $\mathsf{dist\text{-}LWE}_{m,n,q,\chi}$. Let $\mathbf{A'}$ denote the uniform distribution over $\mathbb{Z}_q^{m\times n}$, X' denote the uniform distribution over $\mathbb{Z}_q^n$,

and U' denote the uniform distribution over $\mathbb{Z}_q^m$. We will build a distinguisher D' of polynomial size for which

$$|\Pr[D'(\mathbf{A}', \mathbf{A}'X' + \chi') = 1] - \Pr[D'(\mathbf{A}', U') = 1]| > (\epsilon - \mathtt{ngl}(n))(1 - \mathtt{ngl}(n)) \geq \epsilon - \mathtt{ngl}(n). \quad (1)$$

D' will make a single call to D, so we focus on how to prepare a random block-fixing instance for D from the instance that D' is given. The code for D' is given in Figure 1.

The distinguisher D' has an advantage when $\mathbf{S}$ is of rank α. This occurs with overwhelming probability:

Lemma 5.3. *Let* $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{\alpha \times (\alpha+\beta)}$ *be randomly generated. Then* $\Pr[\mathtt{rank}(\mathbf{S}) = \alpha] \geq 1 - \mathtt{ngl}(n)$.

Proof. Direct result of Claim 4.4 because $q^{-\beta}$ is negligible in n. □

The probability that a random $\mathbf{S}$ is not full rank is negligible in n, so the distinguisher D must still have an advantage when the matrix $\mathbf{S}$ is full rank. That is,

$$|\Pr[D(\mathbf{A}, \mathbf{A}X+\chi) = 1|\mathtt{rank}(\mathbf{S}) = \alpha] - \Pr[D(\mathbf{A}, U) = 1|\mathtt{rank}(\mathbf{S}) = \alpha]| > \epsilon - \mathtt{ngl}(n).$$

It suffices to show that D' prepares a good instance for D conditioned on $\mathbf{S}$ being full rank. We show this in the following three claims:

1. If $\mathbf{A}'$ is a random matrix then $\mathbf{A}$ is a random matrix subject to the condition that $\mathtt{rank}(\mathbf{S}) = \alpha$.
2. If $\mathbf{b}' = \mathbf{A}'\mathbf{x}' + \mathbf{e}'$ for uniform $\mathbf{A}'$ and $\mathbf{x}'$, then $\exists \mathbf{x}$ (uniformly distributed and independent of $\mathbf{A}$ and $\mathbf{e}'$) such that $\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where $\mathbf{e}_i = \mathbf{e}'_i$ for $1 \leq i \leq m$ and $\mathbf{e}_i = 0$ otherwise.
3. If the conditional distribution $\mathbf{b}' \,|\, \mathbf{A}'$ is uniform, then the conditional distribution $\mathbf{b} \,|\, \mathbf{A}$ is also uniform.

Claim 5.4. *The matrix* $\mathbf{A}$ *is distributed as a uniformly random choice from the set of all matrices whose bottom-right* $\alpha \times (\alpha + \beta)$ *submatrix* $\mathbf{S}$ *satisfies* $\mathtt{rank}(\mathbf{S}) = \alpha$.

Proof. The bottom α rows of $\mathbf{A}$ (namely, $\mathbf{R}|\mathbf{S}$) are randomly generated (conditioned on $\mathtt{rank}(\mathbf{S}) = \alpha$). The top left $m \times n$ quadrant of $\mathbf{A}$ is also random because it is produced as a sum of a uniformly random $\mathbf{A}'$ with some values that are uncorrelated with $\mathbf{A}'$. The submatrix of the top-right $m \times (\alpha + \beta)$ quadrant corresponding to $\mathbf{Q}_{T^c}$ (recall this is the restriction of $\mathbf{Q}$ to the columns that are not in T) is also random because it is initialized with random values to which some uncorrelated values are then added. It is important to note that all these values are independent of γ_i values.

1. Input $\mathbf{A}', \mathbf{b}'$, where $\mathbf{A}' \xleftarrow{\$} \mathbb{Z}_q^{m \times n}$ and $\mathbf{b}'$ is either uniform over $\mathbb{Z}_q^m$ or $\mathbf{b}' = \mathbf{A}'\mathbf{x}' + \mathbf{e}'$ for $\mathbf{e}' \xleftarrow{\$} \chi'$ and uniform $\mathbf{x}' \xleftarrow{\$} \mathbb{Z}_q^n$.

2. Choose $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_q^{\alpha \times n}$ uniformly at random. Initialize $\mathbf{Q} \in \mathbb{Z}_q^{m \times (\alpha+\beta)}$ to be the zero matrix.

3. Let $\mathbf{b}^* = (\mathbf{b}', b^*_{m+1}, \ldots, b^*_{m+\alpha})$, for uniformly chosen $(b^*_{m+1}, \ldots, b^*_{m+\alpha}) \xleftarrow{\$} \mathbb{Z}_q^\alpha$.

4. Choose $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{\alpha \times (\alpha+\beta)}$ uniformly at random.

   If $\mathtt{rank}(\mathbf{S}) < \alpha$, stop and output a random bit.

5. Find a set of α linearly independent columns in $\mathbf{S}$. Let T be the set of indices of these columns.

6. For all $1 \leq j \leq \alpha + \beta$, $j \notin T$:

   Choose $x_{n+j} \xleftarrow{\$} \mathbb{Z}_q$ uniformly at random.

   For $i = 1, ..., m$:

   Choose $\mathbf{Q}_{i,j} \xleftarrow{\$} \mathbb{Z}_q$ uniformly at random.

   Set $b^*_i = b^*_i + \mathbf{Q}_{i,j} x_{n+j}$.

7. Initialize $\mathbf{A}^* = \left(\begin{array}{c|c} \mathbf{A}' & \mathbf{Q} \\ \hline \mathbf{R} & \mathbf{S} \end{array}\right)$.

8. For $i = 1, ..., m$:

   Choose a row vector $\gamma_i \leftarrow \mathbb{Z}_q^{1 \times \alpha}$ uniformly at random.

   Set $\mathbf{a}_i \leftarrow \mathbf{a}^*_i + \gamma_i(\mathbf{R}||\mathbf{S})$

   Set $b_i \leftarrow b^*_i + \gamma_i(b^*_{m+1}, ..., b^*_{m+\alpha})^\mathsf{T}$

9. For $i = m+1, \ldots, m+\alpha$:

   Set $\mathbf{a}_i \leftarrow \mathbf{a}^*_i$

   Set $b_i = b^*_i$.

10. Output $D(\mathbf{A}, \mathbf{b})$.

Figure 1: A distinguisher D' for LWE using a distinguisher D for LWE with a block fixing source

Thus, we restrict attention to the $m \times \alpha$ submatrix of $\mathbf{A}$ that corresponds to $\mathbf{Q}_T$ in $\mathbf{A}^*$ (note that these values are 0 in $\mathbf{A}^*$). Consider a particular row i. That row is computed as $\gamma_i \mathbf{S}_T$. Since $\mathbf{S}_T$ is a full rank square matrix and γ_i is uniformly and independently generated, that row is also uniform and independent of other entries in $\mathbf{A}$. □

Lemma 5.5. *If D' is provided with input distributed as $\mathbf{A}', \mathbf{b}' = \mathbf{A}'\mathbf{x}' + \mathbf{e}'$ then $\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{e}$, where*

- *$e_i = e'_i$ for $1 \leq i \leq m$,*
- *$e_i = 0$ for $m < i \leq m + \alpha$,*
- *$x_j = x'_j$ for $1 \leq j \leq n$,*
- *and x_j is uniform and independent of $\mathbf{A}$ and $\mathbf{e}'$ for $n < j \leq n + \alpha + \beta$,*

Proof. Partially define $\mathbf{x}$ as $x_j = x'_j$ if $1 \leq j \leq n$ and x_j as the value generated in step 6 for $j > n$ and $j \notin T$. Define the remaining variables $\mathbf{x}_T$ as the solution to the following system of equations.

$$\mathbf{S}_T \mathbf{x}_T = \begin{pmatrix} b^*_{m+1} \\ \vdots \\ b^*_{m+\alpha} \end{pmatrix} - \mathbf{R}\mathbf{x}' - \mathbf{S}_{T^c}\mathbf{x}_{T^c}. \tag{2}$$

A solution $\mathbf{x}_T$ exists as $\mathbf{S}_T$ is full rank. Moreover, it is uniform and independent of $\mathbf{A}$ and $\mathbf{e}$, because $b^*_{m+1}, \ldots, b^*_{m+\alpha}$ are uniform and independent of $\mathbf{A}$ and $\mathbf{e}$.

We now show that $\mathbf{b}^* = \mathbf{A}^*\mathbf{x} + \mathbf{e}$. All entries in matrix $\mathbf{Q}$ corresponding to variables in T are set to zero. Thus, the values of $\mathbf{x}^T$ do not affect b^*_i for $1 \leq i \leq m$. The values of $\mathbf{x}_{T^c}$ are manually set, and $\mathbf{Q}_{i,j}\mathbf{x}_j$ is added to the corresponding b^*_i. Thus, for $1 \leq i \leq m$, we have $\mathbf{b}^* = \mathbf{A}^*\mathbf{x} + \mathbf{e}$. For $m < i$, this constraint is also satisfied by the values of $\mathbf{x}_T$ set in Equation 2.

Thus, it remains to show that step 8 preserves this solution. We now show that for all rows $1 \leq i \leq m$, if $b^*_i = \mathbf{a}^*_i\mathbf{x} + e_i$ then $b_i = \mathbf{a}_i\mathbf{x} + e_i$. Recall the other rows are not modified. We have the following for $1 \leq i \leq m$:

$$\begin{aligned} \mathbf{a}_i\mathbf{x} + e_i &= (\mathbf{a}^*_i + \gamma_i(\mathbf{R}||\mathbf{S}))\,\mathbf{x} + e_i \\ &= \mathbf{a}^*_i\mathbf{x} + e_i + \gamma_i(\mathbf{R}||\mathbf{S})\mathbf{x} \\ &= b^*_i + \gamma_i(\mathbf{R}||\mathbf{S})\mathbf{x} \end{aligned}$$

Recall that $b_i = b^*_i + \gamma_i(b^*_{m+1}, ..., b^*_{m+k})$. We consider the product $(\mathbf{R}||\mathbf{S})\mathbf{x}$. It

suffices to show that $(\mathbf{R}||\mathbf{S})\mathbf{x} = (b^*_{m+1}, ..., b^*_{m+\alpha})$,

$$\begin{aligned}(\mathbf{R}||\mathbf{S})\mathbf{x} &= \mathbf{R}\begin{pmatrix}\mathbf{x}_1\\ \vdots \\ \mathbf{x}_n\end{pmatrix} + \mathbf{S}_{T^c}\mathbf{x}_{T^c} + \mathbf{S}_T\mathbf{x}_T \\ &= \mathbf{R}\begin{pmatrix}\mathbf{x}_1\\ \vdots \\ \mathbf{x}_n\end{pmatrix} + \mathbf{S}_{T^c}\mathbf{x}_{T^c} + \begin{pmatrix}b^*_{m+1}\\ \vdots \\ b^*_{m+\alpha}\end{pmatrix} - \mathbf{R}\begin{pmatrix}\mathbf{x}_1\\ \vdots \\ \mathbf{x}_n\end{pmatrix} - \mathbf{S}_{T^c}\mathbf{x}_{T^c} \\ &= \begin{pmatrix}b^*_{m+1}\\ \vdots \\ b^*_{m+\alpha}\end{pmatrix}.\end{aligned}$$

This completes the proof of the claim. □

Lemma 5.6. *If the conditional distribution* $\mathbf{b}' \mid \mathbf{A}'$ *is uniform, then* $\mathbf{b} \mid \mathbf{A}$ *is also uniform.*

Proof. Since $\mathbf{R}, \mathbf{S}$, and $\mathbf{Q}$ are chosen independently of $\mathbf{b}'$, the distribution $\mathbf{b}' \mid \mathbf{A}^*$ is uniform. Let $\mathbf{b}^*$ be the vector generated after step 6. Its first m coordinates are computed by adding the uniform vector $\mathbf{b}'$ to values that are independent of $\mathbf{b}^*$, and its remaining α coordinates $b^*_{m+1}, \ldots, b^*_{m+\alpha}$ are chosen uniformly. Thus $\mathbf{b}^* \mid \mathbf{A}^*$ is uniform.

Let $\mathbf{\Gamma}$ represent the matrix formed by γ_i. It is independent of $\mathbf{b}^*$ and $\mathbf{A}^*$, so $\mathbf{b}^* \mid (\mathbf{A}^*, \mathbf{\Gamma})$ is uniform. Let $\mathbf{\Gamma}' = \left(\begin{array}{c|c}\mathbf{I_m} & \mathbf{\Gamma} \\ \hline \mathbf{0} & \mathbf{I}_\alpha\end{array}\right)$. Note that $\mathbf{b} = \mathbf{\Gamma}'\mathbf{b}^*$. Since $\mathbf{b}^* \mid (\mathbf{A}^*, \mathbf{\Gamma})$ is uniform, and $\mathbf{\Gamma}'$ is invertible, $\mathbf{b} \mid (\mathbf{A}^*, \mathbf{\Gamma})$ must also be uniform. Since $\mathbf{A}$ is a deterministic function of $\mathbf{A}^*$ and $\mathbf{\Gamma}$ (assuming Step 5 is deterministic—if not, we can fix the coins used), the distribution $\mathbf{b} \mid \mathbf{A}$ is the same as $\mathbf{b} \mid (\mathbf{A}^*, \mathbf{\Gamma})$ and is thus also uniform. □

To sum up, the reduction D' runs in polynomial-time and Claims 5.4, 5.5, and 5.6 show that when $\mathtt{rank}(\mathbf{S}) = \alpha$, then D' properly prepares the instance for D. Thus,

$$\begin{aligned}&|\Pr[D'(\mathbf{A}, \mathbf{A}X + \chi) = 1] - \Pr[D'(\mathbf{A}, U) = 1]| \\ &= |\Pr[D'(\mathbf{A}', \mathbf{u}') = 1 | \mathtt{rank}(\mathbf{S}) = \alpha] - \Pr[D'(\mathbf{A}', \mathbf{A}'\mathbf{x} + \mathbf{e}) = 1 | \mathtt{rank}(\mathbf{S}) = \alpha]| \\ &\cdot \Pr[\mathtt{rank}(\mathbf{S}) = \alpha] \\ &= |\Pr[D(\mathbf{A}, \mathbf{A}X + \chi) = 1 | \mathtt{rank}(\mathbf{S}) = \alpha] - \Pr[D(\mathbf{A}, U) = 1 | \mathtt{rank}(\mathbf{S}) = \alpha]| \\ &\cdot \Pr[\mathtt{rank}(\mathbf{S}) = \alpha] \\ &\geq (\epsilon - \mathtt{ngl}(n))(1 - \mathtt{ngl}(n)) \geq \epsilon - \mathtt{ngl}(n)\end{aligned}$$

(where the second line follows because D' outputs a random bit when $\mathtt{rank}(\mathbf{S}) < \alpha$). Thus, Equation (1) is satisfied, which completes the proof. □

Theorem 5.2 allows us to construct a lossless computational fuzzy extractor from block-fixing sources, as shown in the following theorem:

Theorem 5.7. *Let n be a security parameter and let $t = c \log n$ for some positive constant c. Let $d \leq c$ be a positive constant and consider the Hamming metric over the alphabet $\mathcal{Z} = [-2^{b-1}, 2^{b-1}]$, where $b \approx \log 2(c/d+2)n^2 = O(\log n)$. Let $\mathcal{M} = \mathcal{Z}^{m+\alpha}$ where $m = (c/d+2)n = O(n)$ and $\alpha \leq n/3$. Let $\mathcal{W}$ be the class of all $(m+\alpha, m, |\mathcal{Z}|)$-symbol fixing sources. There is a setting of $q = \mathtt{poly}(n)$ such that if $\mathsf{dist\text{-}LWE}_{(n,m,q,\mathcal{Z}^m)}$ is (ϵ, s_{sec})-secure, then Construction 4.1 is an $(\mathcal{M}, \mathcal{W}, m \log |\mathcal{Z}|, t)$-computational fuzzy extractor that is (ϵ, s_{sec})-hard with error $\delta = e^{-\Omega(n)}$. The generate procedure Gen takes $O(n^2)$ operations over $\mathbb{Z}_q$, and the reproduce procedure Rep takes expected time $O(n^{4d+3} \log n)$ operations over $\mathbb{Z}_q$.*

Proof. Security follows by Lemmas 4.6 and 4.7 and Theorem 5.2 . Efficiency follows by Lemma 4.10. For a more detailed explanation of parameters see Appendix B. □

Note: A similar theorem for the case of a single fixed dimension was shown in the concurrent work by Brakerski et al. [65, Lemma 4.3]. The proof techniques of Brakerski et al. can be extended to multiple fixed dimensions, improving the parameters of Theorem 5.2. Roughly, the idea is to randomly generate $\mathbf{R}||\mathbf{S}$ as before. Instead of just appending these rows to $\mathbf{A}'$, the matrix $\mathbf{A}$ is formed by multiplying $\mathbf{A}'$ by an invertible matrix $\mathbf{U} \in \mathbb{Z}_q^{n \times n}$. Then $\mathbf{R}||\mathbf{S}$ is appended. This approach creates more flexibility in finding a vector $\mathbf{x}$ that explains the matrix. The advantage of this approach is that only $\mathbf{R}||\mathbf{S}$ has to be full rank instead of the submatrix $\mathbf{S}$. This removes the need for the β extra dimensions in the theorem statement. The original formulation is formalized above.

Acknowledgements

The authors are grateful to Jacob Alperin-Sheriff, Ran Canetti, Yevgeniy Dodis, Nico Döttling, Daniele Micciancio, Jörn Müller-Quade, Chris Peikert, Oded Regev, Adam Smith, and Daniel Wichs for helpful discussions, creative ideas, and important references. In particular, the authors thank Nico Döttling for describing his result on LWE with uniform errors.

This work is supported in part by National Science Foundation grants 0831281, 1012910, 1012798, and 1849904. The work of Benjamin Fuller is sponsored in part by the United States Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

References

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

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

[3] Pim Tuyls, Geert-Jan Schrijen, Boris Skoric, Jan Geloven, Nynke Verhaegh, and Rob Wolters. Read-proof hardware from protective coatings. In Louis Goubin and Mitsuru Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, volume 4249 of *Lecture Notes in Computer Science*, pages 369–383. Springer Berlin Heidelberg, 2006.

[4] G. Edward Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th annual Design Automation Conference*, pages 9–14. ACM, 2007.

[5] Claude Castelluccia and Pars Mutaf. Shake them up!: A movement-based pairing protocol for CPU-constrained devices. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 51–64. ACM, 2005.

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

[7] 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.

[8] Xavier Boyen, Yevgeniy Dodis, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Secure remote authentication using biometric data. In *EUROCRYPT*, pages 147–163. Springer, 2005.

[9] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proceedings of the 41st annual ACM Symposium on Theory of Computing*, pages 601–610, New York, NY, USA, 2009. ACM.

[10] Nishanth Chandran, Bhavana Kanukurthi, Rafail Ostrovsky, and Leonid Reyzin. Privacy amplification with asymptotically optimal entropy loss. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pages 785–794, New York, NY, USA, 2010. ACM.

[11] Nishanth Chandran, Bhavana Kanukurthi, Rafail Ostrovsky, and Leonid Reyzin. Privacy amplification with asymptotically optimal entropy loss. *Journal of the ACM (JACM)*, 61(5):1–28, 2014.

[12] Pierre-Alain Dupont, Julia Hesse, David Pointcheval, Leonid Reyzin, and Sophia Yakoubov. Fuzzy password-authenticated key exchange. In *Advances in Cryptology – EUROCRYPT*, pages 393–424. Springer, 2018.

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

[14] Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77(67-95):10, 2002.

[15] Benjamin Fuller, Leonid Reyzin, and Adam Smith. When are fuzzy extractors possible? In *Advances in Cryptology – ASIACRYPT*, pages 277–306. Springer, 2016.

[16] Benjamin Fuller, Leonid Reyzin, and Adam Smith. When are fuzzy extractors possible? *IEEE Transactions on Information Theory*, 2020.

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

[18] Benjamin Fuller and Lowen Peng. Continuous-source fuzzy extractors: Source uncertainty and insecurity. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 2952–2956. IEEE, 2019.

[19] Hugo Krawczyk. Cryptographic extraction and key derivation: The HKDF scheme. In *Advances in Cryptology–CRYPTO 2010*, pages 631–648. Springer, 2010.

[20] Boaz Barak, Yevgeniy Dodis, Hugo Krawczyk, Olivier Pereira, Krzysztof Pietrzak, François-Xavier Standaert, and Yu Yu. Leftover hash lemma, revisited. In *Advances in Cryptology–CRYPTO 2011*, pages 1–20. Springer, 2011.

[21] Dana Dachman-Soled, Rosario Gennaro, Hugo Krawczyk, and Tal Malkin. Computational extractors and pseudorandomness. In *Theory of Cryptography*, pages 383–403. Springer, 2012.

[22] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[23] Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In *EUROCRYPT*, pages 169–186, 2007.

[24] Boaz Barak, Ronen Shaltiel, and Avi Wigderson. Computational analogues of entropy. In *11th International Conference on Random Structures and Algorithms*, pages 200–215, 2003.

[25] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st annual ACM Symposium on Theory of Computing*, pages 25–32, 1989.

[26] Iftach Haitner, Omer Reingold, Salil Vadhan, and Hoeteck Wee. Inaccessible entropy. In *Proceedings of the 41st annual ACM Symposium on Theory of Computing*, pages 611–620. ACM, 2009.

[27] Iftach Haitner, Thomas Holenstein, Omer Reingold, Salil Vadhan, and Hoeteck Wee. Universal one-way hash functions via inaccessible entropy. In *Advances in Cryptology – EUROCRYPT*, pages 616–637. Springer, 2010.

[28] Salil Vadhan and Colin Jia Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. In *Proceedings of the 44th annual ACM Symposium on Theory of Computing*, pages 817–836. ACM, 2012.

[29] Iftach Haitner, Omer Reingold, and Salil Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. *SIAM Journal on Computing*, 42(3):1405–1430, 2013.

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

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

[32] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th annual ACM Symposium on Theory of Computing*, pages 84–93, New York, NY, USA, 2005. ACM.

[33] Oded Regev. The learning with errors problem (invited survey). *Annual IEEE Conference on Computational Complexity*, 0:191–204, 2010.

[34] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[35] 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, *Advances in Cryptology – EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 18–34. Springer, 2013.

[36] Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen. On the hardness of learning with rounding over small modulus. In *Theory of Cryptography Conference*, pages 209–224. Springer, 2016.

[37] Shi Bai, Tancrède Lepoint, Adeline Roux-Langlois, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. Improved security proofs in lattice-based cryptography: using the Rényi divergence rather than the statistical distance. *Journal of Cryptology*, 31(2):610–640, 2018.

[38] 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.

[39] 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.

[40] Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, 2007.

[41] 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.

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

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

[44] 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.

[45] Yunhua Wen, Shengli Liu, and Shuai Han. Reusable fuzzy extractor from the decisional Diffie–Hellman assumption. *Designs, Codes and Cryptography*, pages 1–18, 2018.

[46] Quentin Alamélou, Paul-Edmond Berthier, Chloé Cachet, Stéphane Cauchie, Benjamin Fuller, Philippe Gaborit, and Sailesh Simhadri. Pseudoentropic isometries: A new framework for fuzzy extractor reusability. In *AsiaCCS 2018*, 2018.

[47] Yunhua Wen and Shengli Liu. Reusable fuzzy extractor from LWE. In *Australasian Conference on Information Security and Privacy*, pages 13–27. Springer, 2018.

[48] 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*, 14(1):65–82, 2017.

[49] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology – CRYPTO*, pages 278–291. Springer, 1993.

[50] Christopher Huth, Daniela Becker, Jorge Guajardo, Paul Duplys, and Tim Güneysu. LWE-based lossless computational fuzzy extractor for the internet of things. In *Hardware Oriented Security and Trust (HOST), 2017 IEEE International Symposium on*, pages 154–154. IEEE, 2017.

[51] Christopher Huth, Daniela Becker, Jorge Guajardo, Paul Duplys, and Tim Güneysu. Securing systems with scarce entropy: LWE-based lossless computational fuzzy extractor for the IoT. *IACR Cryptology ePrint Archive*, 2016:982, 2016.

[52] Christopher Huth, Daniela Becker, Jorge Guajardo Merchan, Paul Duplys, and Tim Güneysu. Securing systems with indispensable entropy: LWE-based lossless computational fuzzy extractor for the internet of things. *IEEE Access*, 5:11909–11926, 2017.

[53] Kenji Yasunaga and Kosuke Yuzawa. On the possibilities and limitations of computational fuzzy extractors. Cryptology ePrint Archive, Report 2014/605, 2014. `http://eprint.iacr.org/`.

[54] Sailesh Simhadri, James Steel, and Benjamin Fuller. Reusable authentication from the iris. In *Information Security Conference*, pages 465–485, 2019.

[55] Chenglu Jin, Charles Herder, Ling Ren, Phuong Nguyen, Benjamin Fuller, Srinivas Devadas, and Marten van Dijk. FPGA implementation of a cryptographically-secure puf based on learning parity with noise. *Cryptography*, 1(3):23, 2017.

[56] Salil Vadhan. *Pseudorandomness*. Foundations and Trends in Theoretical Computer Science. Now Publishers, 2012.

[57] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *Proceedings in the 43rd annual ACM Symposium on the Theory of Computation*, pages 99–108, 2011.

[58] Leonid Reyzin. Some notions of entropy for cryptography. In *Information Theoretic Security*, pages 138–142. Springer, 2011.

[59] Claude E. Shannon, Warren Weaver, Richard E. Blahut, and Bruce Hajek. *The mathematical theory of communication*, volume 117. University of Illinois press Urbana, 1949.

[60] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-InterScience, 2nd edition, 2006.

[61] Bhavana Kanukurthi and Leonid Reyzin. Key agreement from close secrets over unsecured channels. In *Advances in Cryptology – EUROCRYPT*, pages 206–223, 2009.

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

[63] Colin Cooper. On the rank of random matrices. *Random Structures & Algorithms*, 16(2):209–232, 2000.

[64] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *Proceedings of the 41st annual ACM Symposium on Theory of Computing*, pages 333–342, New York, NY, USA, 2009. ACM.

[65] 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.

[66] 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.

[67] Pil Joong Lee and Ernest F Brickell. An observation on the security of mceliece's public-key cryptosystem. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 275–280. Springer, 1988.

[68] Piotr Berman and Marek Karpinski. Approximating minimum unsatisfiability of linear equations. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 514–516. Society for Industrial and Applied Mathematics, 2002.

[69] Christiane Peters. Information-set decoding for linear codes over Fq. In *International Workshop on Post-Quantum Cryptography*, pages 81–94. Springer, 2010.

[70] Ran Canetti and Shafi Goldwasser. An efficient threshold public key cryptosystem secure against adaptive chosen ciphertext attack. In *Advances in Cryptology – EUROCRYPT*, pages 90–106. Springer, 1999.

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

[72] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. On pseudorandom generators with linear stretch in NC 0. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 260–271, 2006.

[73] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. On pseudorandom generators with linear stretch in nc 0. *Computational Complexity*, 17(1):38–69, 2008.

[74] Marina Blanton and William MP Hudelson. Biometric-based non-transferable anonymous credentials. In *International Conference on Information and Communications Security*, pages 165–180. Springer, 2009.

Appendix A. Parameter Settings for Construction 4.1

In this section, we explain the different parameters that go into our construction. In Theorem 4.11 we give a parametrization that yieldslossless fuzzy extractor from a security parameter n and an error t. In this section, we discuss constraints imposed by 1) efficient decoding 2) maintaining security of the LWE instance and 3) ensuring no entropy loss of the construction. We begin by reviewing the parameters that make up our construction:

- $|W|$: The length of the source.
- t: Number of errors that can be supported.
- n: LWE security parameter (i.e., number of field elements in X), which must be greater than some minimum value n_0 for security.
- q: The size of the field.
- ρ: The fraction of the field needed for error sampling.
- m: The size of each number of samples in the LWE instance.
- k: The number of hardcore elemebnts in X (from Lemma 4.7).

We will split the source $|W|$ into m blocks each of size $2\rho q + 1$ (that is, $|W| = m\log(2\rho q+1)$). We focus on t, n, q, ρ, and m noting that $|W| = m\log(2\rho q+1)$. As stated above we have three constraints:

- Maintain security of LWE. Theorem 4.6 says that we get security for all n greater than some minimum n_0 and $q = \texttt{poly}(n)$ and $\rho q \geq \alpha n^{\sigma} mq$. For convenience we consider $\alpha q = 2\sqrt{n}$ as needed in the original Regev reduction [32, 34]. The only reason to increase ρq over this minimum amount (other than security) is if the number of errors in W decreases with a slightly larger block size. We ignore this effect and assume that $\rho q = 2n^{1/2+\sigma} m$.
- Maintain efficient decoding of Construction 4.9. Using Lemma 4.10, this means that $t \leq d\log n(m/n - 2)$.
- Minimize entropy loss of the construction. We will output $X_{1,\ldots,k}$ so the entropy loss of the construction is $|W| - |X_{1,\ldots,k}|$. We want the entropy loss to be zero, that is, $|W| = |X_{1,\ldots,k}|$. Substituting, one has $m\log(2\rho q+1) = k\log q$.

Collecting constraints we can support any setting where t, n, q, ρ, m, k satisfy the following constraints (for constants d, f):

$$\begin{aligned} n_0 &< n - k \\ t &\leq d\log n\left(\frac{m}{n} - 2\right) \\ q &= n^f \\ \rho q &= 2n^{1/2+\sigma} m \\ m\log(2\rho q + 1) &= k\log q \end{aligned}$$

Substituting $q = n^f$ and $\rho q = 2n^{1/2+\sigma}m$ yields the following system of equations:

$$\begin{aligned} n_0 &< n - k \\ t &\leq d \log n \left(\frac{m}{n} - 2\right) \\ m \log(4n^{1/2+\sigma}m + 1) &= k \log n^f \end{aligned}$$

This is the most general form of our construction, we can support any n, t, m that satisfy these equations for constants d, f. However, the last equation may have no solution for f constant. Putting the last equation in terms of f one has:

$$\begin{aligned} n_0 &< n - k \\ t &\leq d \log n \left(\frac{m}{n} - 2\right) \\ f &= \frac{m}{k} \frac{\log 4n^{1/2+\sigma}m + 1}{\log n} \end{aligned}$$

To ensure f is a constant, we set $t = c \log n$ for some constant c and that $k = n/g$ for some constant $g > 1$. Finally we assume that m is the minimum value such that $t \leq d \log n(m/n - 2)$ (that is, there are only as many dimensions as necessary for decoding using Lemma 4.10):

$$\begin{aligned} n_0 &< n - k \\ m &= \frac{(c/d + 2)n \log n}{\log n} = \left(\frac{c}{d} + 2\right) n \\ f &= \frac{m}{k} \frac{\log 4n^{1/2+\sigma}m + 1}{\log n} = \frac{g(c + 2d)}{d} \frac{\log(\frac{4(c+2d)}{d} n^{3/2+\sigma} + 1)}{\log n} = O(1) \end{aligned}$$

Assuming $n - k = n(1 - 1/g) > n_0$ and letting $t = c \log n$ we get the following setting:

$$\begin{aligned} m &= \left(\frac{c}{d} + 2\right) n \\ q &= n^f = n^{(\frac{m}{k} \frac{\log(4n^{1/2+\sigma}m+1)}{\log n})} = \texttt{poly}(n) \\ \rho q &= 2n^{1/2+\sigma}m = 2(\frac{c}{d} + 2)n^{3/2+\sigma} \end{aligned}$$

Note, that $f > \frac{m}{k} \geq \frac{m}{n} \geq \frac{(c/d+2)n}{n} \geq 3$ as long as $d < c$ (this also ensures that $m \geq 3n$, as required for Lemma 4.10 to hold). Since $\rho q = 2n^{1/2+\sigma}m = O(n^{5/2})$ in our setting $\rho = O(n^{-1/2})$. Thus, for large enough settings of parameters ρ is less than $1/10$ as required by Theorem 4.6.

Furthermore, we get decoding using $O(n^{4d+3})$ $\mathbb{Z}_q$ operations. We can output a k fraction of X and the bits will be pseudorandom (conditioned on $\mathbf{A}, \mathbf{A}X + W$). The parameter g allows is a tradeoff between the number of dimensions

needed for security and the size of the field q. In Theorem 4.11, we set $g = 2$ and output the first half of X. Setting $1 < g < 2$ achieves an increase in output length (over the input length of W). We also (arbitrarily) set $\sigma = 1/2$ to simplify the statement of Theorem 4.11, making $\rho q = 2(c/d + 2)n^2$.

Appendix B. Parameter Settings for Theorem 5.7

We repeat parameter settings for block fixing sources. We now have $m + \alpha$ as the number of samples, while $n + \alpha + \omega(1)$ is the number of variables. We can support any setting where $t, n, q, \rho, m, k, \alpha$ satisfy the following constraints (for $\beta = \omega(1)$ and constants d, f):

$$\begin{aligned} n_0 &< n - k - \alpha - \beta \\ t &\leq d \log n \left(\frac{m}{n} - 2\right) \\ q &= n^f \\ \rho q &= 2n^{1/2+\sigma} m \\ m \log(2\rho q + 1) &= k \log q \end{aligned}$$

Substituting $q = n^f$ and $\rho q = 2n^{1/2+\sigma} m$ yields the following system of equations:

$$\begin{aligned} n_0 &< n - k - \alpha - \beta \\ t &\leq d \log n \left(\frac{m}{n} - 2\right) \\ m \log(4n^{1/2+\sigma} m + 1) &= k \log n^f \end{aligned}$$

As before we can support any setting any n, t, m, α that satisfy these equations for $\beta = \omega(1)$ and constants d, f. However, the last equation may have no solution for f constant. Putting the last equation in terms of f one has:

$$\begin{aligned} n_0 &< n - k - \alpha - \beta \\ t &\leq d \log n \left(\frac{m}{n} - 2\right) \\ f &= \frac{m}{k} \frac{\log(4n^{1/2+\sigma} m + 1)}{\log n} \end{aligned}$$

To ensure f is a constant, we set $t = c \log n$ for some constant c and that $k, \alpha = n/3$ and $\beta = \log n$. Finally we assume that m is the minimum value such that $t \leq d \log n(m/n - 2)$ (that is, there are only as many dimensions as necessary for decoding using Lemma 4.10):

$$\begin{aligned} n_0 &< n/3 - \log n \\ m &= \frac{(c/d + 2)n \log n}{\log n} = (\frac{c}{d} + 2)n \\ f &= \frac{m}{k} \frac{\log(4n^{1/2+\sigma} m + 1)}{\log n} = \left(3(\frac{c}{d} + 2)\right) \frac{\log(4(\frac{c}{d} + 2)n^{3/2+\sigma} + 1)}{\log n} = O(1) \end{aligned}$$

Assuming $n/3 - \log(n) > n_0$ and letting $t = c \log n$ we get the following setting:

$$\begin{aligned} m &= (\frac{c}{d} + 2)n \\ q &= n^f = n^{\frac{m}{n} \frac{\log(4n^{1/2+\sigma}m+1)}{\log n}} = \texttt{poly}(n) \\ \rho q &= 2n^{1/2+\sigma}m = 2(\frac{c}{d} + 2)n^{3/2+\sigma} \end{aligned}$$

As before we arbitrarily set $\sigma = 1/2$, giving $\rho q = 2(\frac{c}{d} + 2)n^2$. Also, if $c < d$ then we get efficient decoding and $\rho = o(1)$ satisfying the conditions of Theorem 4.6.