

# Computational Fuzzy Extractors

Benjamin Fuller<sup>1</sup>, Xianrui Meng<sup>2</sup>, and Leonid Reyzin<sup>2</sup>

<sup>1</sup> Boston University and MIT Lincoln Laboratory

<sup>2</sup> Boston University

**Abstract.** Fuzzy extractors derive strong keys from noisy sources. Their security is defined information-theoretically, which limits the length of the derived key, sometimes making it too short to be useful. We ask whether it is possible to obtain longer keys by considering computational security, and show the following.

- **Negative Result:** Noise tolerance in fuzzy extractors is usually achieved using an information reconciliation component called a “secure sketch.” The security of this component, which directly affects the length of the resulting key, is subject to lower bounds from coding theory. We show that, even when defined computationally, secure sketches are still subject to lower bounds from coding theory. Specifically, we consider two computational relaxations of the information-theoretic security requirement of secure sketches, using conditional HILL entropy and unpredictability entropy. For both cases we show that computational secure sketches cannot outperform the best information-theoretic secure sketches in the case of high-entropy Hamming metric sources.
- **Positive Result:** We show that the negative result can be overcome by analyzing computational fuzzy extractors directly. Namely, we show how to build a computational fuzzy extractor whose output key length equals the entropy of the source (this is impossible in the information-theoretic setting). Our construction is based on the hardness of the Learning with Errors (LWE) problem, and is secure when the noisy source is uniform or symbol-fixing (that is, each dimension is either uniform or fixed). As part of the security proof, we show a result of independent interest, namely that the decision version of LWE is secure even when a small number of dimensions has no error.

**Keywords:** Fuzzy extractors, secure sketches, key derivation, Learning with Errors, error-correcting codes, computational entropy, randomness extractors.

## 1 Introduction

Authentication generally 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 [14], human memory [37], pictorial passwords [9], measurements of capacitance [35], timing [34], motion [10], quantum information [5] etc.).

Fuzzy extractors [15] achieve reliable key derivation from noisy sources (see [7,16,11] for applications of fuzzy extractors). The setting 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  $p$ . This information allows the Reproduce (Rep) algorithm 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 [7] (where two parties have  $w$  and  $w'$ , respectively), it would be transmitted between the parties.

Fuzzy extractors use ideas from information-reconciliation [5] and are defined (traditionally) as information-theoretic objects. The entropy loss of a fuzzy extractor is the difference between the entropy of  $w$  and the length of the derived key  $r$ . In the information-theoretic setting, some entropy loss is necessary as the value  $p$  contains enough information to reproduce  $r$  from any close value  $w'$ . A goal of fuzzy extractor constructions is to minimize the entropy loss, increasing the security of the resulting application. Indeed, if the entropy loss is too high, the resulting secret key may be too short to be useful.

We ask whether it is possible to obtain longer keys by considering computational, rather than information theoretic, security.

*Our Negative Results.* We first study (in Section 3) whether it could be fruitful to relax the definition of the main building block of a fuzzy extractor, called a *secure sketch*. A secure sketch is a one-round information reconciliation protocol: it produces a public value  $s$  that allows recovery of  $w$  from any close value  $w'$ . The traditional secrecy requirement of a secure sketch is that  $w$  has high min-entropy conditioned on  $s$ . This allows the fuzzy extractor of [15] to form the key  $r$  by applying a randomness extractor [28] to  $w$ , because randomness extractors produce random strings from strings with conditional min-entropy. We call this the *sketch-and-extract* construction.

The most natural relaxation of the min-entropy requirement of the secure sketch is to require HILL entropy [21] (namely, that the distribution of  $w$  conditioned on  $s$  be *indistinguishable* from a high-min-entropy distribution). Under this definition, we could still use a randomness extractor to obtain  $r$  from  $w$ , because it would yield a pseudorandom key. Unfortunately, it is unlikely that such a relaxation will yield fruitful results: we prove in Theorem 1 that the entropy loss of such secure sketches is subject to the same coding bounds as the ones that constrain information-theoretic secure sketches.

Another possible relaxation is to require that the value  $w$  is unpredictable conditioned on  $s$ . This definition would also allow the use of a randomness extractor

to get a pseudorandom key, although it would have to be a special extractor—one that has a reconstruction procedure (see [22, Lemma 6]). Unfortunately, this relaxation is also unlikely to be fruitful: we prove in Theorem 2 that the unpredictability is at most  $\log$  the size of the metric space minus  $\log$  the volume of the ball of radius  $t$ . For high-entropy sources of  $w$  over the Hamming metric, this bound matches the best information-theoretic security sketches.

*Our Positive Results.* Both of the above negative results arise because a secure sketch functions like a decoder of an error-correcting code. To avoid them, we give up on building computational secure sketches and focus directly on the entropy loss in fuzzy extractors. Our goal is to decrease the entropy loss in a fuzzy extractor by allowing the key  $r$  to be pseudorandom conditioned on  $p$ .

By considering this computational secrecy requirement, we construct the first *lossless* computational fuzzy extractors (Construction 1), where the derived key  $r$  is as long as the entropy of the source  $w$ . Our construction is for the Hamming metric and uses the code-offset construction [23],[15, 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 [23] as a “fuzzy commitment”). Our construction uses private randomness, which is allowed in the fuzzy extractor setting but not in 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 [30,31] and derive a longer key  $r$ .

Specifically, we use the recent result of Döttling and Müller-Quade [17], 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. We also use a result of Akavia, Goldwasser, and Vaikuntanathan [1], which says that LWE has many hardcore bits, to hide  $r$ .

Because we use a random linear code, our decoding is limited to reconciling a logarithmic number of differences. Unfortunately, we cannot utilize the results that improve the decoding radius through the use of trapdoors (such as [30]), because in a fuzzy extractor, there is no secret storage place for the trapdoor. If improved decoding algorithms are obtained for random linear codes, they will improve error-tolerance of our construction. Given the hardness of decoding random linear codes [6], we do not expect significant improvement in the error-tolerance of our construction.

In Section 5, we are able to relax the assumption that  $w$  comes from the uniform distribution, and instead allow  $w$  to come from a symbol-fixing source [24] (each dimension is either uniform or fixed). This relaxation follows from our results about the hardness of LWE when samples have a fixed (and adversarially known) error vector, which may be of independent interest (Theorem 4).

*An Alternative Approach.* Computational extractors [26,3,13] have the same goal of obtaining a pseudorandom key  $r$  from a source  $w$  in the setting without errors.

They can be constructed, for example, by applying a pseudorandom generator to the output of an information-theoretic extractor. One way to build a computational *fuzzy* extractor is by using a computational extractor instead of the information-theoretic extractor in the sketch-and-extract construction of [15]. However, this approach is possible only if conditional min-entropy of  $w$  conditioned on the sketch  $s$  is high enough. Furthermore, this approach does not allow the use of private randomness; private randomness is a crucial ingredient in our construction. We compare the two approaches in Section 4.4.

## 2 Preliminaries

For a random variable  $X = X_1 || \dots || X_n$  where each  $X_i$  is over some alphabet  $\mathcal{Z}$ , we denote by  $X_{1,\dots,k} = X_1 || \dots || X_k$ . The *min-entropy* of  $X$  is  $H_\infty(X) = -\log(\max_x \Pr[X = x])$ , and the *average (conditional) min-entropy* of  $X$  given  $Y$  is  $\bar{H}_\infty(X|Y) = -\log(\mathbb{E}_{y \in Y} \max_x \Pr[X = x|Y = y])$  [15, Section 2.4]. 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  $\delta^{\mathcal{D}}(X, Y) = |\mathbb{E}[D(X)] - \mathbb{E}[D(Y)]|$ . 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  $(\mathcal{M}, \text{dis})$ , the *(closed) ball of radius  $t$  around  $x$*  is the set of all points within radius  $t$ , that is,  $B_t(x) = \{y | \text{dis}(x, y) \leq t\}$ . 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.

### 2.1 Fuzzy Extractors and Secure Sketches

We now recall definitions and lemmas from the work of Dodis et. al. [15, Sections 2.5–4.1], adapted to allow for a small probability of error, as discussed in [15, Sections 8]. Let  $\mathcal{M}$  be a metric space with distance function  $\text{dis}$ .

**Definition 1.** An  $(\mathcal{M}, m, \ell, t, \epsilon)$ -fuzzy extractor with error  $\delta$  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. The correctness property of fuzzy extractors guarantees that for  $w$  and  $w'$  such that  $\text{dis}(w, w') \leq t$ , if  $R, P$  were generated by  $(R, P) \leftarrow \text{Gen}(w)$ , then  $\text{Rep}(w', P) = R$  with probability (over the coins of **Gen**, **Rep**) at least  $1 - \delta$ . If  $\text{dis}(w, w') > t$ , then no guarantee is provided about the output of **Rep**.

3. The security property guarantees that for any distribution  $W$  on  $\mathcal{M}$  of min-entropy  $m$ , the string  $R$  is nearly uniform even for those who observe  $P$ : if  $(R, P) \leftarrow \text{Gen}(W)$ , then  $\text{SD}((R, P), (U_\ell, P)) \leq \epsilon$ .

A fuzzy extractor is efficient if  $\text{Gen}$  and  $\text{Rep}$  run in expected polynomial time.

Secure sketches are the main technical tool in the construction of 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.** An  $(\mathcal{M}, m, \tilde{m}, t)$ -secure sketch with error  $\delta$  is a pair of randomized procedures, “sketch” ( $\text{SS}$ ) and “recover” ( $\text{Rec}$ ), with the following properties:

1. The sketching procedure  $\text{SS}$  on input  $w \in \mathcal{M}$  returns a bit string  $s \in \{0, 1\}^*$ .
2. The recovery procedure  $\text{Rec}$  takes an element  $w' \in \mathcal{M}$  and a bit string  $s \in \{0, 1\}^*$ . The correctness property of secure sketches guarantees that if  $\text{dis}(w, w') \leq t$ , then  $\Pr[\text{Rec}(w', \text{SS}(w)) = w] \geq 1 - \delta$  where the probability is taken over the coins of  $\text{SS}$  and  $\text{Rec}$ . If  $\text{dis}(w, w') > t$ , then no guarantee is provided about the output of  $\text{Rec}$ .
3. The security property guarantees that for any distribution  $W$  over  $\mathcal{M}$  with min-entropy  $m$ , the value of  $W$  can be recovered by the adversary who observes  $w$  with probability no greater than  $2^{-\tilde{m}}$ . That is,  $\tilde{H}_\infty(W|\text{SS}(W)) \geq \tilde{m}$ .

A secure sketch is efficient if  $\text{SS}$  and  $\text{Rec}$  run in expected polynomial time.

Note that in the above definition of secure sketches (resp., fuzzy extractors), the errors are chosen before  $s$  (resp.,  $P$ ) is known: if the error pattern between  $w$  and  $w'$  depends on the output of  $\text{SS}$  (resp.,  $\text{Gen}$ ), then there is no guarantee about the probability of correctness.

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 [28, Definition 2]) (in particular, Vadhan [36, Problem 6.8] showed that all strong extractors are average-case extractors with a slight loss of parameters):

**Definition 3.** Let  $\chi_1, \chi_2$  be finite sets. A function  $\text{ext} : \chi_1 \times \{0, 1\}^d \rightarrow \{0, 1\}^\ell$  a  $(m, \epsilon)$ -average-case extractor if for all pairs of random variables  $X, Y$  over  $\chi_1, \chi_2$  such that  $\tilde{H}_\infty(X|Y) \geq m$ , we have  $\Delta((\text{ext}(X, U_d), U_d, Y), U_\ell \times U_d \times Y) \leq \epsilon$ .

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

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

The main parameter we will be concerned with is the entropy loss of the construction. In this paper, we ask whether a smaller entropy loss can be achieved

by considering a fuzzy extractor with a computational security requirement. We therefore relax the security requirement of Definition 1 to require a pseudorandom output instead of a truly random output. Also, for notational convenience, we 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 that consists of all sources of a given min-entropy  $m$ , as in definitions above (of course, this modification can also be applied to prior definitions of information-theoretic secure sketches and fuzzy extractors).

**Definition 4 (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  $(\epsilon, s_{\text{sec}})$ -hard with error  $\delta$  if Gen and Rep satisfy the following properties:*

- *The generate procedure Gen on input  $w \in \mathcal{M}$  outputs an extracted string  $R \in \{0, 1\}^\ell$  and a helper string  $P \in \{0, 1\}^*$ .*
- *The reproduction procedure Rep takes an element  $w' \in \mathcal{M}$  and a bit string  $P \in \{0, 1\}^*$  as inputs. The correctness property guarantees that for all  $w, w'$  where  $\text{dis}(w, w') \leq t$ , if  $(R, P) \leftarrow \text{Gen}(w)$  then  $\Pr[\text{Rep}(w', P) = R] \geq 1 - \delta$  where the probability is over the randomness of (Gen, Rep). If  $\text{dis}(w, w') > t$ , then no guarantee is provided about the output of Rep.*
- *The security property guarantees that for any distribution  $W \in \mathcal{W}$ , the string  $R$  is pseudorandom conditioned on  $P$ , that is  $\delta^{\mathcal{D}_{s_{\text{sec}}}}((R, P), (U_\ell, P)) \leq \epsilon$ .*

Any efficient fuzzy extractor is also a computational fuzzy extractor with the same parameters.

**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  $\text{dis}(w, w') > t$ .

### 3 Impossibility of Computational Secure Sketches

In this section, we consider whether it is possible to build a secure sketch that retains significantly more computational than information-theoretic entropy. We consider two different notions for computational entropy, and for both of them show that corresponding secure sketches are subject to the same upper bounds as those for information-theoretic secure sketches. Thus, it seems that relaxing security of sketches from information-theoretic to computational does not help.

In particular, for the case of the Hamming metric and inputs that have full entropy, our results are as follows. In Section 3.1 we show that a sketch that

retains HILL entropy implies a sketch that retains nearly the same amount of min-entropy. In Section 3.2, we show that the computational unpredictability of a sketch is at most  $\log |\mathcal{M}| - \log |B_t(\cdot)|$ . Dodis et al. [15, Section 8.2] construct sketches with essentially the same information-theoretic security<sup>1</sup>. In Section 3.3, we discuss mechanisms for avoiding these bounds.

### 3.1 Bounds on Secure Sketches Using HILL Entropy

HILL entropy is a commonly used computational notion of entropy [21]. It was extended to the conditional case by Hsiao, Lu, Reyzin [22]. Here we recall a weaker definition due to Gentry and Wichs [19] (the term relaxed HILL entropy was introduced in [32]); since we show impossibility even for this weaker definition, impossibility for the stronger definition follows immediately.

**Definition 5.** *Let  $(W, S)$  be a pair of random variables.  $W$  has relaxed HILL entropy at least  $k$  conditioned on  $S$ , denoted  $H_{\epsilon, s_{sec}}^{\text{HILL-rlx}}(W|S) \geq k$  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 is as good 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 1 would yield pseudo-random outputs if the secure sketch ensured high HILL entropy. We will consider secure sketches that retain relaxed HILL entropy: that is, we say that  $(\text{SS}, \text{Rec})$  is a *HILL-entropy*  $(\mathcal{M}, m, \tilde{m}, t)$  *secure sketch* that is  $(\epsilon, s_{sec})$ -hard with error  $\delta$  if it satisfies Definition 2, with the security requirement replaced by  $H_{\epsilon, s_{sec}}^{\text{HILL-rlx}}(W|\text{SS}(W)) \geq \tilde{m}$ .

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 [15, 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 [15, Proposition 8.2]. In fact, we show that, for the Hamming metric, 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  $H_{\epsilon, s_{sec}}^{\text{HILL-rlx}}(W|\text{SS}(W)) \geq \tilde{m}$ , there must be a distribution  $X, Y$  such that  $\tilde{H}_\infty(X|Y) \geq \tilde{m}$  and  $(X, Y)$  is computationally indistinguishable from  $(W, \text{SS}(W))$ . Sample a sketch  $s \leftarrow \text{SS}(W)$ . We know that  $\text{SS}$  followed by  $\text{Rec}$  likely succeeds on  $W|s$  (i.e.,  $\text{Rec}(w', s) = w$  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,

<sup>1</sup> The security in [15, Section 8.2] is expressed in terms of entropy of the error rate; recall that  $\log B_t(\cdot) \approx H_q(t/n)$ , where  $n$  is the number of symbols,  $q$  is the alphabet size, and  $H_q$  is the  $q$ -ary entropy function.

$\text{Rec}(x', y) = x$  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  $\text{Rec}$  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 6.** *We say a metric space  $(\mathcal{M}, \text{dis})$  is  $(s_{\text{neigh}}, t)$ -neighborhood samplable if there exists a randomized circuit  $\text{Neigh}$  of size  $s_{\text{neigh}}$  that for all  $t' \leq t$ ,  $\text{Neigh}(w, t')$  outputs a random point at distance  $t'$  of  $w$ .*

We review the definition of a Shannon code [33]:

**Definition 7.** *Let  $\mathcal{C}$  be a set over space  $\mathcal{M}$ . We say that  $\mathcal{C}$  is an  $(t, \epsilon)$ -Shannon code if there exists an efficient procedure  $\text{Rec}$  such that for all  $t' \leq t$  and for all  $c \in \mathcal{C}$ ,  $\Pr[\text{Rec}(\text{Neigh}(c, t')) \neq c] \leq \epsilon$ . 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<sup>2</sup>. Shannon codes work for all codewords. We can also consider a formulation that works for an ‘‘average’’ codeword.

**Definition 8.** *Let  $\mathcal{C}$  be a distribution over space  $\mathcal{M}$ . We say that  $\mathcal{C}$  is an  $(t, \epsilon)$ -average error Shannon code if there exists an efficient procedure  $\text{Rec}$  such that for all  $t' \leq t$   $\Pr_{c \leftarrow \mathcal{C}}[\text{Rec}(\text{Neigh}(c, t')) \neq c] \leq \epsilon$ .*

An average error Shannon code is one whose average probability of error is bounded by  $\epsilon$ . See [12, 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 [12, Pages 202-204] (we provide a proof in the full version [18]):

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

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 (proof in the full version of this work [18]).

**Theorem 1.** *Let  $(\mathcal{M}, \text{dis})$  be a  $(s_{\text{neigh}}, t)$ -neighborhood samplable metric space. Let  $(\text{SS}, \text{Rec})$  be a HILL-entropy  $(\mathcal{M}, m, \tilde{m}, t)$ -secure sketch that is  $(\epsilon, s_{\text{sec}})$ -secure*

<sup>2</sup> 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.



with error  $\delta$ . Let  $s_{rec}$  denote the size of the circuit that computes  $\text{Rec}$ . 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  $(t, 4(\epsilon + t\delta))$ -Shannon code with recovery procedure  $\text{Rec}(\cdot, s)$ .

For the Hamming metric, any Shannon code (as defined in Definition 7) can be converted into an information-theoretic secure sketch (as described in [15, 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  $\pi$  (which becomes part of the sketch and is applied to  $w'$  during  $\text{Rec}$ ). The formal statement of this result can be expressed in the following Lemma (which is implicit in [15, Section 8.2]).

**Lemma 3.** *For an alphabet  $\mathcal{Z}$ , let  $\mathcal{C}$  over  $\mathcal{Z}^n$  be a  $(t, \delta)$  Shannon code. Then there exists a  $(\mathcal{Z}^n, m, m - (n \log |\mathcal{Z}| - \log |\mathcal{C}|), t)$  secure sketch with error  $\delta$  for the Hamming metric over  $\mathcal{Z}^n$ .*

Putting together Theorem 1 and Lemma 3 gives us the negative result for the Hamming metric: a HILL-entropy secure sketch (for the uniform distribution) implies an information-theoretic one with similar parameters:

**Corollary 1.** *Let  $\mathcal{Z}$  be an alphabet. Let  $(\text{SS}', \text{Rec}')$  be an  $(\epsilon, s_{sec})$ -HILL-entropy  $(\mathcal{Z}^n, n \log |\mathcal{Z}|, \tilde{m}, t)$ -secure sketch with error  $\delta$  for the Hamming metric over  $\mathcal{Z}^n$ , with  $\text{Rec}'$  of circuit size  $s_{rec}$ . If  $s_{sec} \geq t(s_{rec} + n \log |\mathcal{Z}|)$ , then there exists a  $(\mathcal{Z}^n, n \log |\mathcal{Z}|, \tilde{m} - 2, t)$  (information-theoretic) secure sketch with error  $4(\epsilon + t\delta)$ .*

**Note.** In Corollary 1 we make no claim about the efficiency of the resulting  $(\text{SS}, \text{Rec})$ , because the proof of Theorem 1 is not constructive.

Corollary 1 extends to non-uniform distributions: if there exists a distribution whose HILL sketch retains  $\tilde{m}$  bits of entropy, then for all distributions  $W$ , there is an information theoretic sketch that retains  $H_\infty(W) - (n \log |\mathcal{Z}| - \tilde{m}) - 2$  bits of entropy.

### 3.2 Bounds on Secure Sketches Using Unpredictability Entropy

In the previous section, we showed that any sketch that retained HILL entropy could be transformed into an information theoretic sketch. However, HILL entropy is a strong notion. In this section, we therefore ask whether it is useful to consider a sketch that satisfies a minimal requirement: the value of the input is computationally hard to guess given the sketch. We begin by recalling the definition of conditional unpredictability entropy [22, Definition 7], which captures the notion of “hard to guess” (we relax the definition slightly, similarly to the relaxation of HILL entropy described in the previous section).

**Definition 9.** *Let  $(W, S)$  be a pair of random variables.  $W$  has relaxed unpredictability entropy at least  $k$  conditioned on  $S$ , denoted by  $H_{\epsilon, s_{sec}}^{\text{unp-rlx}}(W|S) \geq k$ , if there exists a pair of distributions  $(X, Y)$  such that  $\delta^{\mathcal{D}^{s_{sec}}}((W, S), (X, Y)) \leq \epsilon$ , and for all circuits  $\mathcal{I}$  of size  $s_{sec}$ ,*

$$\Pr[\mathcal{I}(Y) = X] \leq 2^{-k}.$$

A pair of procedures  $(\text{SS}, \text{Rec})$  is a *unpredictability-entropy*  $(\mathcal{M}, m, \tilde{m}, t)$  *secure sketch* that is  $(\epsilon, s_{\text{sec}})$ -hard with error  $\delta$  if it satisfies Definition 2, with the security requirement replaced by  $H_{\epsilon, s_{\text{sec}}}^{\text{unp-r1x}}(W|\text{SS}(W)) \geq \tilde{m}$ . Note this notion is quite natural: combining such a secure sketch in a sketch-and-extract construction of Lemma 1 with a particular type of extractor (called a *reconstructive* extractor [4]), would yield a computational fuzzy extractor (per [22, Lemma 6]).

Unfortunately, the conditional unpredictability entropy  $\tilde{m}$  must decrease as  $t$  increases, as the following theorem states. (The proof of the theorem, generalized to more metric spaces, is in the full version [18].)

**Theorem 2.** *Let  $\mathcal{Z}$  be an alphabet. Let  $(\text{SS}, \text{Rec})$  be an unpredictability-entropy  $(\mathcal{Z}^n, m, \tilde{m}, t)$ -secure sketch that is  $(\epsilon, s_{\text{sec}})$ -secure with error  $\delta$ , if  $s_{\text{sec}} \geq t(|\text{Rec}| + n \log |\mathcal{Z}|)$ , then  $\tilde{m} \leq n \log |\mathcal{Z}| - \log |B_t(\cdot)| + \log(1 - \epsilon - t\delta)$ .*

In particular, if the input is uniform, the entropy loss is about  $\log |B_t(\cdot)|$ . As mentioned at the beginning of Section 3, essentially the same entropy loss can be achieved with information-theoretic secure sketches, by using the randomized code-offset construction. However, it is conceivable that unpredictability entropy secure sketches could achieve lower entropy loss with greater efficiency for some parameter settings.

### 3.3 Avoiding Sketch Entropy Upper Bounds

The lower bounds of Corollary 1 and Theorem 2 are strongest for high entropy sources. This is necessary, if a source contains only codewords (of an error correcting code), no sketch is needed, and thus there is no (computational) entropy loss. This same situation occurs when considering lower bounds for information-theoretic sketches [15, Appendix C].

Both of lower bounds arise because  $\text{Rec}$  must function as an error-correcting code for many points of any indistinguishable distribution. It may be possible to avoid these bounds if  $\text{Rec}$  outputs a fresh random variable<sup>3</sup>. Such an algorithm is called a computational fuzzy conductor. See [25] for the definition of a fuzzy conductor. To the best of our knowledge, a computational fuzzy conductor has not been defined in the literature, the natural definition is to replace the pseudorandomness condition in Definition 4 with a HILL entropy requirement.

Our construction (in Section 4) has pseudorandom output and immediately satisfies definition of a computational fuzzy extractor (Definition 4). It may be possible to achieve significantly better parameters with a construction that is a computational fuzzy conductor (but not a computational fuzzy extractor) and then applying an extractor. We leave this as an open problem.

---

<sup>3</sup> If some efficient algorithm can take the output of  $\text{Rec}$  and efficiently transform it back to the source  $W$ , the bounds of Corollary 1 and Theorem 2 both apply. This means that we need to consider constructions that are hard to invert (either information-theoretically or computationally).

## 4 Computational Fuzzy Extractor Based on LWE

In this section we describe our main construction. Security of our construction depends on the source  $W$ . We first consider a uniform source  $W$ ; we consider other distributions in Section 5. Our construction uses the code-offset construction [23], [15, Section 5] instantiated with a random linear code over a finite field  $\mathbb{F}_q$ . Let  $\text{Decode}_t$  be an algorithm that decodes a random linear code with at most  $t$  errors (we will present such an algorithm later, in Section 4.2).

**Construction 1.** *Let  $n$  be a security parameter and let  $m \geq n$ . Let  $q$  be a prime. Define  $\text{Gen}, \text{Rep}$  as follows:*

Gen	Rep
<ol style="list-style-type: none"> <li>1. <u>Input</u>: <math>w \leftarrow W</math> (where <math>W</math> is some distribution over <math>\mathbb{F}_q^m</math>).</li> <li>2. Sample <math>\mathbf{A} \in \mathbb{F}_q^{m \times n}</math>, <math>\mathbf{x} \in \mathbb{F}_q^n</math> uniformly.</li> <li>3. Compute <math>p = (\mathbf{A}, \mathbf{A}\mathbf{x} + w)</math>, <math>r = \mathbf{x}_{1, \dots, n/2}</math>.</li> <li>4. Output <math>(r, p)</math>.</li> </ol>	<ol style="list-style-type: none"> <li>1. <u>Input</u>: <math>(w', p)</math> (where the Hamming distance between <math>w'</math> and <math>w</math> is at most <math>t</math>).</li> <li>2. Parse <math>p</math> as <math>(\mathbf{A}, \mathbf{c})</math>; let <math>\mathbf{b} = \mathbf{c} - w'</math>.</li> <li>3. Let <math>x = \text{Decode}_t(\mathbf{A}, \mathbf{b})</math></li> <li>4. Output <math>r = x_{1, \dots, n/2}</math>.</li> </ol>

Intuitively, security comes from the computational hardness of decoding random linear codes with a high number of errors (introduced by  $w$ ). In fact, we know that decoding a random linear code is NP-hard [6]; however, this statement is not sufficient for our security goal, which is to show

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

Furthermore, this construction is only useful if  $\text{Decode}_t$  can be efficiently implemented.

The rest of this section is devoted to making these intuitive statements precise. We describe the LWE problem and the security of our construction in Section 4.1. We describe one possible polynomial-time  $\text{Decode}_t$  (which corrects more errors than is possible by exhaustive search) in Section 4.2. In Section 4.3, we describe parameter settings that allow us to extract as many bits as the input entropy, resulting in a lossless construction. In Section 4.4, we compare Construction 1 to using a sketch-and-extract approach (Lemma 1) instantiated with a computational extractor.

### 4.1 Security of Construction 1

The LWE problem was introduced by Regev [30,31] as a generalization of “learning parity with noise.” For a complete description of the LWE problem and related lattices problems (which we do not define here) see [30]. We now recall the decisional version of the problem.

**Definition 10 (Decisional LWE).** *Let  $n$  be a security parameter. Let  $m = m(n) = \text{poly}(n)$  be an integer and  $q = q(n) = \text{poly}(n)$  be a prime<sup>4</sup>. Let  $\mathbf{A}$  be the uniform distribution over  $\mathbb{F}_q^{m \times n}$ ,  $X$  be the uniform distribution over  $\mathbb{F}_q^n$  and  $\chi$  be an arbitrary distribution on  $\mathbb{F}_q^m$ . The decisional version of the LWE problem, denoted  $\text{dist-LWE}_{n,m,q,\chi}$ , is to distinguish the distribution  $(\mathbf{A}, \mathbf{A}X + \chi)$  from the uniform distribution over  $(\mathbb{F}_q^{m \times n}, \mathbb{F}_q^m)$ .*

*We say that  $\text{dist-LWE}_{n,m,q,\chi}$  is  $(\epsilon, s_{\text{sec}})$ -secure if no (probabilistic) distinguisher of size  $s_{\text{sec}}$  can distinguish the LWE instances from uniform except with probability  $\epsilon$ . If for any  $s_{\text{sec}} = \text{poly}(n)$ , there exists  $\epsilon = \text{ngl}(n)$  such that  $\text{dist-LWE}_{n,m,q,\chi}$  is  $(\epsilon, s_{\text{sec}})$ -secure, then we say it is secure.*

Regev [30] and Peikert [29] show that  $\text{dist-LWE}_{n,m,q,\chi}$  is secure when the distribution  $\chi$  of errors is Gaussian, as follows. 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  $\text{dist-LWE}_{n,m,q,\bar{\Psi}_\rho^m}$  is secure. (A recent result of Brakerski et al. [8] shows security of LWE based on hardness of approximating lattices problems for classical algorithms. We have not considered how this result can be integrated into our analysis.)

The above formulation of LWE requires the error term to come from the discretized Gaussian distribution, which makes it difficult to use it for constructing fuzzy extractors (because using  $w$  and  $w'$  to sample Gaussian distributions will increase the distance between the error terms and/or reduce their entropy). Fortunately, recent work Döttling and Müller-Quade [17] shows the security of LWE, under the same assumptions, when errors come from the uniform distribution over a small interval<sup>5</sup>. 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:

**Lemma 4.** *[17, Corollary 1] Let  $n$  be a security parameter. Let  $q = q(n) = \text{poly}(n)$  be a prime and  $m = m(n) = \text{poly}(n)$  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  $\rho q \geq 2n^{1/2+\sigma}m$ . If the approximate decision-version of the shortest vector problem (GAPSVP) and the shortest independent vectors problem (SIVP) are hard within a factor of  $\tilde{O}(n^{1+\sigma}m/\rho)$  for quantum algorithms in the worst case, then, for  $\chi$  the uniform distribution over  $[-\rho q, \rho q]^m$ ,  $\text{dist-LWE}_{n,m,q,\chi}$  is secure.*

To extract pseudorandom bits, we use a result of Akavia, Goldwasser, and Vaikuntanathan [1] to show that  $X$  has simultaneously many hardcore bits. The result says that if  $\text{dist-LWE}_{(n-k,m,q,\chi)}$  is secure then any  $k$  variables of  $X$  in a  $\text{dist-LWE}_{(n,m,q,\chi)}$  instance are hardcore. We state their result for a general error distribution (noting that their proof does not depend on the error distribution):

---

<sup>4</sup> Unlike in common formulations of LWE, where  $q$  can be any integer, we need  $q$  to be prime for decoding.

<sup>5</sup> Micciancio and Peikert provide a similar formulation in [27]. The result Döttling and Müller-Quade provides better parameters for our setting.

**Lemma 5.** [1, Lemma 2] If  $\text{dist-LWE}_{(n-k,m,q,\chi)}$  is  $(\epsilon, s_{\text{sec}})$  secure, then

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

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

The security of Construction 1 follows from Lemmas 4 and 5 when parameters are set appropriately (see Theorem 3), because we use the hardcore bits of  $X$  as our key.

## 4.2 Efficiency of Construction 1

Construction 1 is useful only if  $\text{Decode}_t$  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  $\text{Decode}_t$  that runs in polynomial time and can correct correcting  $O(\log n)$  errors (note that this corresponds to a superpolynomial number of possible error patterns). This algorithm is a proof of concept, and neither the algorithm nor its analysis have been optimized for constants. An improved decoding algorithm can replace our algorithm, which will increase our correcting capability and improve Construction 1.

**Construction 2.** We consider a setting of  $(n, m, q, \chi)$  where  $m \geq 3n$ . We describe  $\text{Decode}_t$ :

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

Each step is computable in time  $O(n^3)$ . For  $\text{Decode}_t$  to be efficient, we need  $t$  to be small enough so that with probability at least  $\frac{1}{\text{poly}(n)}$ , 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  $\mathbf{A}_{i_1, \dots, i_{2n}}$  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  $\mathbf{b} - \mathbf{A}\mathbf{x}' = \mathbf{A}(\mathbf{x} - \mathbf{x}') + (w - w')$  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 distance at least  $O(\frac{m}{n} \log n)$  with probability  $1 - e^{-\Omega(n)}$  (the exact statement is in Corollary 2), so this also ensures correctness. The formal statement is below (proof in the full version of this work [18]):

**Lemma 6 (Efficiency of  $\text{Decode}_t$  when  $t \leq d(m/n - 2) \log n$ ).** *Let  $d$  be a positive constant and assume that  $\text{dis}(W, W') \leq t$  where  $t \leq d(\frac{m}{n} - 2) \log n$ . Then  $\text{Decode}_t$  runs in expected time  $O(n^{4d+3})$  operations in  $\mathbb{F}_q$  (this expectation is over the choice of random coins of  $\text{Decode}_t$ , regardless of the input, as long as  $\text{dis}(w, w') \leq t$ ). 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  $\text{Decode}_t$ ).*

### 4.3 Lossless Computational Fuzzy Extractor

We now state a setting of parameters that yields a lossless construction. The intuition is as follows. We are splitting our source into  $m$  blocks each of size  $\log \rho q$  (from Lemma 4) for a total input entropy of  $m \log \rho q$ . Our key is derived from hardcore bits of  $X$ :  $X_{1, \dots, k}$  and is of size  $k \log q$  (from Lemma 5). Thus, to achieve a lossless construction we need  $k \log q = m \log \rho q$ . In other words, in order to decode a meaningful number of errors, the vector  $w$  is of higher dimension than the vector  $X$ , but each coordinate of  $w$  is sampled using fewer bits than each coordinate of  $X$ . Thus, by increasing the size of  $q$  (while keeping  $\rho q$  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 3.** *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  $\text{Rep}$  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  $m = (c/d + 2)n = O(n)$ . If GAPSVP and SIVP are hard to approximate within polynomial factors using quantum algorithms, then there is a setting of  $q = \text{poly}(n)$  such that for any polynomial  $s_{\text{sec}} = \text{poly}(n)$  there exists  $\epsilon = \text{ngl}(n)$  such that the following holds: Construction 1 is a  $(\mathcal{M}, W, m \log |\mathcal{Z}|, t)$ -computational fuzzy extractor that is  $(\epsilon, s_{\text{sec}})$ -hard with error  $\delta = e^{-\Omega(n)}$ . The generate procedure  $\text{Gen}$  takes  $O(n^2)$  operations over  $\mathbb{F}_q$ , and the reproduce procedure  $\text{Rep}$  takes expected time  $O(n^{4d+3})$  operations over  $\mathbb{F}_q$ .*

*Proof.* Security follows by combining Lemmas 4 and 5; efficiency follows by Lemma 6. For a detailed explanation of the various parameters and constraints see the full version of this work [18].

Theorem 3 shows that a computational fuzzy extractor can be built without incurring any entropy loss. We can essentially think of  $\mathbf{A}X + W$  as an encryption of  $X$  that where decryption works from any close  $W'$ .

#### 4.4 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., [26,3,13]) 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 [2].)

This approach (specifically, its analysis via Lemma 1) 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, and it is conceivable that, if  $W$  has barely enough entropy to begin with, it will have 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. Instead, we require that  $w$  is not computationally recoverable given  $p$ . This requirement is weaker—in particular, in our construction,  $w$  may have no information-theoretic entropy conditioned on  $p$ . 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 private randomness [26, Definition 3].

The main advantage of our analysis (instead of *sketch-and-comp-extract*) is that security need not depend on the error-tolerance  $t$ . In our construction, the error-tolerance depends only on the best available decoding algorithm for random linear codes, because decoding algorithms will not reach the information-theoretic decoding radius.

Unfortunately, LWE parameter sizes require relatively long  $w$ . Therefore, in practice, *sketch-then-comp-extract* will beat our construction if the computational extractor is instantiated efficiently based on assumptions other than LWE (for example, a cryptographic hash function for an extractor and a block cipher for a PRG). However, we believe that our conceptual framework can lead to better constructions. Of particular interest are other codes that are easy to decode up to  $t$  errors but become computationally hard as the number of errors increases.

To summarize, the advantage of Construction 1 is that the security of our construction does not depend on the decoding radius  $t$ . The disadvantages of Construction 1 are that it supports a limited number of errors and only a uniformly distributed source. We begin to address this second problem in the next section.

## 5 Computational Fuzzy Extractor for Nonuniform Sources

While showing the security of Construction 1 for arbitrary high-min-entropy distributions is an open problem, in this section we show it for a particular class

of distributions called symbol-fixing. First we recall the notion of a symbol fixing source (from [24, Definition 2.3]):

**Definition 11.** *Let  $W = (W_1, \dots, W_{m+\alpha})$  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  $\alpha$  indices  $i_1, \dots, i_\alpha$ , the symbols  $W_{i_\alpha}$  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 1 to such a class is not obvious. Although symbol-fixing sources are deterministically extractible [24], we cannot first run a deterministic extractor before using Construction 1. This is because we need to preserve distance between  $w$  and  $w'$  and an extractor must not preserve distance between input points. We present an alternative approach, showing security of LWE directly with symbol-fixing sources.

The following theorem states the main technical result of this section, which is of potential interest outside our specific setting. The result is that  $\text{dist-LWE}$  with symbol-fixing sources is implied by standard  $\text{dist-LWE}$  (but for  $n$  and  $m$  reduced by the amount of fixed symbols).

**Theorem 4.** *Let  $n$  be a security parameter,  $m, \alpha$  be polynomial in  $n$ , and  $q = \text{poly}(n)$  be a prime and  $\beta \in \mathbb{Z}^+$  be such that  $q^{-\beta} = \text{ngl}(n)$ . Let  $U$  denote the uniform distribution over  $\mathcal{Z}^m$  for an alphabet  $\mathcal{Z} \subset \mathbb{F}_q$ , and let  $W$  denote an  $(m + \alpha, m, |\mathcal{Z}|)$  symbol fixing source over  $\mathcal{Z}^{m+\alpha}$ . If  $\text{dist-LWE}_{n,m,q,U}$  is secure, then  $\text{dist-LWE}_{n+\alpha+\beta, m+\alpha, q, W}$  is also secure.*

Theorem 4 also holds for an arbitrary error distribution (not just uniform error) in the following sense. Let  $\chi'$  be an arbitrary error distribution. Define  $\chi$  as the distribution where  $m$  dimensions are sampled according to  $\chi'$  and the remaining dimensions have some fixed error. Then, security of  $\text{dist-LWE}_{n,m,q,\chi'}$  implies security of  $\text{dist-LWE}_{n+\alpha+\beta, m+\alpha, q, \chi}$ . We prove this stronger version of the theorem in the full version of this work [18].

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  $\text{dist-LWE}$  is still hard. The nontrivial part of the reduction is using the additional  $\alpha + \beta$  variables to “explain” a random value for the last  $\alpha$  samples, without knowing the other variables. The  $\beta$  parameter is the slack needed to ensure that the “free” variables have influence on the last  $\alpha$  samples. A similar theorem for the case of a single fixed dimension was shown in concurrent work by Brakerski et al. [8, Lemma 4.3]. The proof techniques of Brakerski et al. can be extended to our setting with multiple fixed dimensions, improving the parameters of Theorem 4 (specifically, removing the need for  $\beta$ ).

Theorem 4 allows us to construct a lossless computational fuzzy extractor from block-fixing sources:

**Theorem 5.** *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. If GAPSVP and SIVP are hard to approximate within polynomial factors using quantum algorithms, then there is a setting of  $q = \text{poly}(n)$  such that for any polynomial  $s_{\text{sec}} = \text{poly}(n)$  there exists  $\epsilon = \text{ngl}(n)$  such that the following holds: Construction 1 is a  $(\mathcal{M}, \mathcal{W}, m \log |\mathcal{Z}|, t)$ -computational fuzzy extractor that is  $(\epsilon, s_{\text{sec}})$ -hard with error  $\delta = e^{-\Omega(n)}$ . The generate procedure  $\text{Gen}$  takes  $O(n^2)$  operations over  $\mathbb{F}_q$ , and the reproduce procedure  $\text{Rep}$  takes expected time  $O(n^{4d+3} \log n)$  operations over  $\mathbb{F}_q$ .

*Proof.* Security follows by Lemmas 4 and 5 and Theorem 4. Efficiency follows by Lemma 6. For a more detailed explanation of parameters see the full version of this work [18].

**Acknowledgements.** The authors are grateful to Jacob Alperin-Sheriff, Ran Canetti, Yevgeniy Dodis, Nico Döttling, Danielle Micciancio, Jörn Müller-Quade, Christopher 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 supported in part by National Science Foundation grants 0831281, 1012910, and 1012798. 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. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009), [http://dx.doi.org/10.1007/978-3-642-00457-5\\_28](http://dx.doi.org/10.1007/978-3-642-00457-5_28)
2. Applebaum, B., Ishai, Y., Kushilevitz, E.: On pseudorandom generators with linear stretch in NC 0. Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, pp. 260–271 (2006)
3. Barak, B., Dodis, Y., Krawczyk, H., Pereira, O., Pietrzak, K., Standaert, F.-X., Yu, Y.: Leftover hash lemma, revisited. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 1–20. Springer, Heidelberg (2011)
4. Barak, B., Shaltiel, R., Wigderson, A.: Computational analogues of entropy. In: 11th International Conference on Random Structures and Algorithms, pp. 200–215 (2003)
5. Bennett, C.H., Brassard, G., Robert, J.M.: Privacy amplification by public discussion. SIAM Journal on Computing 17(2), 210–229 (1988)
6. Berlekamp, E., McEliece, R., van Tilborg, H.: On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory 24(3), 384–386 (1978)

7. Boyen, X., Dodis, Y., Katz, J., Ostrovsky, R., Smith, A.: Secure remote authentication using biometric data. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 147–163. Springer, Heidelberg (2005)
8. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the 45th Annual ACM Symposium on Symposium on Theory of Computing, pp. 575–584. ACM (2013)
9. Brostoff, S., Sasse, M.: Are passfaces more usable than passwords?: A field trial investigation. *People and Computers*, 405–424 (2000)
10. Castelluccia, C., Mutaf, P.: 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, pp. 51–64. ACM (2005)
11. Chandran, N., Kanukurthi, B., Ostrovsky, R., Reyzin, L.: Privacy amplification with asymptotically optimal entropy loss. In: Proceedings of the 42nd ACM Symposium on Theory of Computing, pp. 785–794. ACM, New York (2010), <http://doi.acm.org/10.1145/1806689.1806796>
12. Cover, T.M., Thomas, J.A.: Elements of information theory, 2nd edn. Wiley Interscience (2006)
13. Dachman-Soled, D., Gennaro, R., Krawczyk, H., Malkin, T.: Computational extractors and pseudorandomness. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 383–403. Springer, Heidelberg (2012)
14. Daugman, J.: How iris recognition works. *IEEE Transactions on Circuits and Systems for Video Technology* 14(1), 21–30 (2004)
15. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing* 38(1), 97–139 (2008)
16. Dodis, Y., Wichs, D.: Non-malleable extractors and symmetric key cryptography from weak secrets. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp. 601–610. ACM, New York (2009), <http://doi.acm.org/10.1145/1536414.1536496>
17. Döttling, N., Müller-Quade, J.: Lossy codes and a new variant of the learning-with-errors problem. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 18–34. Springer, Heidelberg (2013)
18. Fuller, B., Meng, X., Reyzin, L.: Computational fuzzy extractors. *Cryptology ePrint Archive* (2013), <http://eprint.iacr.org/2013/416>
19. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: STOC, pp. 99–108. ACM, New York (2011)
20. Guruswami, V.: Introduction to coding theory - lecture 2: Gilbert-Varshamov bound. *University Lecture* (2010)
21. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. *SIAM Journal on Computing* 28(4), 1364–1396 (1999)
22. Hsiao, C.-Y., Lu, C.-J., Reyzin, L.: Conditional computational entropy, or toward separating pseudoentropy from compressibility. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 169–186. Springer, Heidelberg (2007)
23. Juels, A., Wattenberg, M.: A fuzzy commitment scheme. In: Sixth ACM Conference on Computer and Communication Security, pp. 28–36. ACM (November 1999)
24. Kamp, J., Zuckerman, D.: Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing* 36(5), 1231–1247 (2007)
25. Kanukurthi, B., Reyzin, L.: Key agreement from close secrets over unsecured channels. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 206–223. Springer, Heidelberg (2009)

26. Krawczyk, H.: Cryptographic extraction and key derivation: The HKDF scheme. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 631–648. Springer, Heidelberg (2010)
27. Micciancio, D., Peikert, C.: Hardness of SIS and LWE with Small Parameters. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 21–39. Springer, Heidelberg (2013)
28. Nisan, N., Zuckerman, D.: Randomness is linear in space. *Journal of Computer and System Sciences*, 43–52 (1993)
29. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pp. 333–342. ACM, New York (2009), <http://doi.acm.org/10.1145/1536414.1536461>
30. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, pp. 84–93. ACM, New York (2005), <http://doi.acm.org/10.1145/1060590.1060603>
31. Regev, O.: The learning with errors problem (invited survey). In: *Annual IEEE Conference on Computational Complexity*, pp. 191–204 (2010)
32. Reyzin, L.: Some notions of entropy for cryptography. In: Fehr, S. (ed.) ICITS 2011. LNCS, vol. 6673, pp. 138–142. Springer, Heidelberg (2011)
33. Shannon, C.E., Weaver, W., Blahut, R.E., Hajek, B.: *The mathematical theory of communication*, vol. 117. University of Illinois press Urbana (1949)
34. Suh, G.E., Devadas, S.: Physical unclonable functions for device authentication and secret key generation. In: *Proceedings of the 44th Annual Design Automation Conference*, pp. 9–14. ACM (2007)
35. Tuyls, P., Schrijen, G.-J., Škorić, B., van Geloven, J., Verhaegh, N., Wolters, R.: Read-proof hardware from protective coatings. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 369–383. Springer, Heidelberg (2006), [http://dx.doi.org/10.1007/11894063\\_29](http://dx.doi.org/10.1007/11894063_29)
36. Vadhan, S.: *Pseudorandomness. Foundations and Trends in Theoretical Computer Science*. Now Publishers (2012)
37. Zviran, M., Haga, W.J.: A comparison of password techniques for multilevel authentication mechanisms. *The Computer Journal* 36(3), 227–237 (1993)

## A Properties of Random Linear Codes

For efficient decoding of Construction 1, we need the LWE instance 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  $H_2(x) = -x \log x - (1-x) \log(1-x)$ . 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 6.** [20, 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  $\mathbf{A} \in \mathbb{F}_q^{m \times n}$  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  $\delta$  with probability at least  $1 - e^{-\Omega(m)}$ .*

Our setting is the case where  $m = \text{poly}(n) \geq 2n$  and  $\delta = O(\log n/n)$ . This setting of parameters satisfies Theorem 6:

**Corollary 2.** *Let  $n$  be a parameter and let  $m = \text{poly}(n) \geq 2n$ . Let  $q$  be a prime and  $\tau = O(\frac{m}{n} \log n)$ . For large enough values of  $n$ , when  $\mathbf{A} \in \mathbb{F}_q^{m \times n}$  is drawn uniformly, the code generated by  $\mathbf{A}$  has distance at least  $\tau$  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 6. 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.