Reusable Fuzzy Extractors for Low-Entropy Distributions

Ran Canetti* Benjamin Fuller† Omer Paneth‡ Leonid Reyzin§ Adam Smith¶

February 25, 2016

Abstract

Fuzzy extractors (Dodis et al., Eurocrypt 2004) convert repeated noisy readings of a secret into the same uniformly distributed key. To eliminate noise, they require an initial enrollment phase that takes the first noisy reading of the secret and produces a nonsecret helper string to be used in subsequent readings. *Reusable* fuzzy extractors (Boyen, CCS 2004) remain secure even when this initial enrollment phase is repeated multiple times with noisy versions of the same secret, producing multiple helper strings (for example, when a single person's biometric is enrolled with multiple unrelated organizations).

We construct the first reusable fuzzy extractor that makes no assumptions about how multiple readings of the source are correlated (the only prior construction assumed a very specific, unrealistic class of correlations). The extractor works for binary strings with Hamming noise; it achieves computational security under assumptions on the security of hash functions or in the random oracle model. It is simple and efficient and tolerates near-linear error rates.

Our reusable extractor is secure for source distributions of linear min-entropy rate. The construction is also secure for sources with much lower entropy rates—lower than those supported by prior (nonreusable) constructions—assuming that the distribution has some additional structure, namely, that random subsequences of the source have sufficient minentropy. We show that such structural assumptions are necessary to support low entropy rates.

We then explore further how different structural properties of a noisy source can be used to construct fuzzy extractors when the error rates are high, building a computationally secure and an information-theoretically secure construction for large-alphabet sources.

Keywords Fuzzy extractors, reusability, key derivation, error-correcting codes, computational entropy, digital lockers, point obfuscation.

1 Introduction

Fuzzy Extractors Cryptography relies on long-term secrets for key derivation and authentication. However, many sources with sufficient randomness to form long-term secrets provide similar but not identical values of the secret at repeated readings. Prominent examples include biometrics and other human-generated data [Dau04, ZH93, BS00, EHMS00, MG09, MRW02], physically unclonable functions (PUFs) [PRTG02, TSS+06, GCVDD02, SD07], and quantum information [BBR88]. Turning similar readings into identical values is known as *information reconciliation*; further converting those values into

*Email: canetti@cs.bu.edu. Boston University and Tel Aviv University.
†Email: bfuller@cs.bu.edu. Boston University and MIT Lincoln Laboratory.
‡Email: paneth@cs.bu.edu. Boston University.
§Email: reyzin@cs.bu.edu. Boston University.
¶Email: asmith@cse.psu.edu. Pennsylvania State University.

uniformly random secret strings is known as *privacy amplification* [BBR88]. Both of these problems have interactive and non-interactive versions. In this paper, we are interested in the non-interactive case, which is useful for a single user trying to produce the same key from multiple noisy readings of a secret at different times. A *fuzzy extractor* [DORS08] is the primitive that accomplishes both information reconciliation and privacy amplification non-interactively.

Fuzzy extractors consist of a pair of algorithms: Gen (used once, at "enrollment") takes a source value w, and produces a key r and a public helper value p. The second algorithm Rep (used subsequently) takes this helper value p and a close w' to reproduce the original key r. The standard correctness guarantee is that r will be correctly reproduced by Rep as long as w' is no farther than t from w according to some notion of distance (specifically, we work with Hamming distance; our primary focus is on binary strings, although we also consider larger alphabets). The security guarantee is that r produced by Gen is indistinguishable from uniform, even given p. In this work, we consider computational indistinguishability [FMR13] rather than the more traditional information-theoretic notion. (Note that so-called "robust" fuzzy extractors [Mau97, MW97, BDK+05, CDF+08, DKK+12] additionally protect against active attackers who modify p; we do not consider them here, except to point out that our constructions can be easily made robust by the random-oracle-based transform of [BDK+05, Theorem 1].)

Reusability A fuzzy extractor is *reusable* (Boyen [Boy04]) if it remains secure even when a user enrolls the same or correlated values multiple times. For example, if the source is a biometric reading, the user may enroll the same biometric with different noncooperating organizations. Reusability is particularly important for biometrics which, unlike passwords, cannot be changed or created. It is also useful in other contexts, for example, to permit a user to reuse the same visual password across many services or to make a single physical token (embodying a PUF) usable for many applications.

Each enrollment process will get a slightly different enrollment reading w_i, and will run $\mathsf{Gen}(w_i)$ to get a key r_i and a helper value p_i. Security for each r_i should hold even when an adversary is given all the values $p_1, \ldots, p_\rho$ and even the keys r_j for $j \neq i$ (because one organization cannot be sure how other organizations will use the derived keys).

As pointed out by Dodis et al. [DKL09, Section 6], reusable extractors for the nonfuzzy case (i.e., without p and Rep) can be constructed using leakage-resilient cryptography. However, adding error-tolerance makes the problem harder. Most constructions of fuzzy extractors are not reusable [Boy04, STP09, BA12, BA13]. In fact, the only known construction of reusable fuzzy extractors [Boy04] requires very particular relationships between w_i values[1], which are unlikely to hold for a practical source.

1.1 Our Contribution

A Reusable Fuzzy Extractor We construct the first reusable fuzzy extractor whose security holds even if the multiple readings w_i used in Gen are *arbitrarily correlated*, as long as the fuzzy extractor is secure for each w_i *individually*. This construction is the first to provide reusability for a realistic class of correlated readings. Our construction is based on *digital lockers*; in the most efficient instantiation, it requires only evaluation of cryptographic hash functions and is secure in the random oracle model or under strong computational assumptions on the hash functions.[2] The construction can output arbitrarily long r.

[1] Specifically, Boyen's construction requires that the exclusive or $w_i \oplus w_j$ of any two secrets not leak any information about w_i.

[2] The term "digital lockers" was introduced by Canetti and Dakdouk [CD08]; the fact that such digital lockers can be built easily out cryptographic hash functions was shown by [LPS04, Section 4].

Our construction handles a wider class of sources than prior work. It is secure if the bits of w are partially independent. Namely, we require that, for some known parameter k, the substring formed by the bits at k randomly chosen positions in w is unguessable (i.e., has minentropy that is superlogarithmic is the security parameter). We call sources with this feature "sources with high-entropy samples." This requirement is in contrast to most constructions of fuzzy extractors that require w to have sufficient minentropy.

All sources of sufficient minentropy have high-entropy samples (because sampling preserves the entropy rate [Vad03]). However, as we now explain, the family of sources with high-entropy samples also includes some low-entropy sources. (Note that, of course, the entropy of a substring never exceeds the entropy of the entire string; the terms "high" and "low" are relative to the length.)

Low-entropy sources with high-entropy samples are easy to construct artificially: for example, we can build a source of length n whose bits are k-wise independent by multiplying (over GF(2)) a fixed $n \times k$ matrix of rank k by a random k-bit vector. In this source, the entropy rate of any substring of length k is 1, while the entropy rate of the entire string is just k/n.

Such sources also arise naturally whenever w exhibits a lot of redundancy. For example, when the binary string w is obtained via signal processing from some underlying reading (such as an image of an iris or an audio recording of a voice), the signal itself is likely to have a lot of redundancy (for example, nearby pixels of an image are highly correlated). By requiring only high-entropy samples rather than a high entropy rate, we free the signal processing designer from the need to remove redundancy when converting the underlying reading to a string w used in the fuzzy extractor. Thus, we enable the use of oversampled signals.

Our construction can tolerate $\frac{n \ln n}{k}$ errors (out of the n bits of w) if we allow the running time of the construction (the number of hash function evaluations) to be linear in n. More generally, we can tolerate $c\frac{n \ln n}{k}$ errors if we allow running time linear in n^c. Note that, since in principle k needs to be only slightly superlogarithmic to ensure the high-entropy condition on the samples, our allowable error rate is only slightly sublinear.

The Advantage of Exploiting the Structure of the Distribution Following the tradition of extractor literature [CG88, NZ93], much work on fuzzy extractors has focused on providing constructions that work for *any* source of a given minentropy m. In contrast, our construction exploits more about the structure of the distribution than just its entropy. As a result, it supports not only all sources of a given (sufficiently high) minentropy, but also many sources with an entropy rate much lower than the error rate. We know of no prior constructions with this property. We now explain why, in order to achieve this property, exploiting the structure of the distribution is necessary.

A fuzzy extractor that supports t errors out of a string of n bits and works for *all* sources of minentropy m must have the entropy rate $\frac{m}{n}$ at least as big as the binary entropy[3] of the error rate, $h_2(\frac{t}{n})$ (to be exact, $m \geq nh_2(\frac{t}{n}) - \frac{1}{2}\log n - \frac{1}{2}$). The reason for this requirement is simple: if m too small, then a single ball of radius t, which contains at least $2^{nh_2(\frac{t}{n})-\frac{1}{2}\log n-\frac{1}{2}}$ points [Ash65, Lemma 4.7.2, Equation 4.7.5, p. 115], may contain the entire distribution of 2^m points inside it. For this distribution, an adversary can run Rep on the center of this ball and always learn the key r. This argument leads to the following proposition, which holds regardless of whether the fuzzy extractor is information-theoretic or computational, and extends even to the interactive setting.

[3]Binary entropy $h_2(\alpha)$ for $0 < \alpha < 1$ is defined as $-\alpha \log_2 \alpha - (1-\alpha)\log_2(1-\alpha)$; it is greater than $\alpha \log_2 \frac{1}{\alpha}$ and, in particular, greater than α for interesting range $\alpha < \frac{1}{2}$.

Proposition 1. *If the security guarantee of a fuzzy extractor holds for* any *source of minentropy m and the correctness guarantees holds for any t errors and $m < \log |B_t|$ (where $|B_t|$ denotes the number of points in a ball of radius t), the fuzzy extractor must provide no security. In particular, for the binary Hamming case, m must exceed $nh_2(\frac{t}{n}) - \frac{1}{2}\log n - \frac{1}{2} \approx nh_2(\frac{t}{n}) > t \log_2 \frac{n}{t}$.*

Thus, in order to correct t errors regardless of the structure of the distribution, we would have to assume a high total minentropy m. In contrast, by taking advantage of the specific properties of the distribution, we can handle all distributions of sufficiently high minentropy, but also some distributions whose minentropy that is much less than $t < nh_2(\frac{t}{n})$.

Beating the bound of Proposition 1 is important. For example, the IrisCode [Dau04], which is the state of the art approach to handling what is believed to be the best biometric [PPJ03], produces a source where m is less than $nh_2(\frac{t}{n})$ [BH09, Section 5]. PUFs with slightly nonuniform outputs suffer from similar problems [KLRW14].

We emphasize that in applications of fuzzy extractors to physical sources, any constraint on the source—whether minentropy-based or more structured—is always, by necessity, an assumption about the physical world. It is no more possible to verify that a source has high minentropy than it is to verify that it has high-entropy samples[4]. Both statements about the source can be derived only by modeling the source—for example, by modeling the physical processes that generate irises or PUFs.

Some prior work on key agreement from noisy data also made assumptions on the structure of the source (often assuming that it consists of independent identically distributed symbols, e.g. [Mau93, MW96, MTV09, YD10, HMSS12]). However, we are not aware of any work that beat the bound of Propostion 1, with the exception of the work by Holenstein and Renner [HR05, Theorem 4]. Their construction supports a uniform length n binary w, with a random selection of $(n-m)$ bits leaked to the adversary and t random bits flipped in w'. They show that it is possible to support any $m > 4t(1 - \frac{t}{n})$, which is lower than $\log |B_t| \approx nh_2(\frac{t}{n})$, but still higher than t.

Constructions Exploiting the Structure of the Distribution for Larger Alphabets In addition to the binary alphabet construction that supports reuse and low entropy rates, as discussed above, we explore how low entropy rates can be supported when symbols of the string w comes from a large, rather than a binary, alphabet. We obtain two additional constructions, both of which allow for distributions whose total minentropy is lower than the volume of the ball of radius t (in the large-alphabet Hamming space). Unfortunately, neither of them provides reusability, but both can tolerate a linear error rate (of course, over the larger alphabet, where errors may be more likely, because each symbol carries more information).

Our second construction for large alphabets works for *sources with sparse high-entropy marginals*: sources for which sufficiently many symbols have high entropy individually, but no independence among symbols is assumed (thus, the total entropy may be as low as the entropy of a single symbol).

Our third construction for large alphabets provides information-theoretic, rather than computational, security. It works for *sparse block sources*. These are sources in which a sufficient fraction of the symbols have entropy conditioned on previous symbols.

Both constructions should be viewed as evidence that assumptions on the source other than total minentropy may provide new opportunities for increasing the error tolerance of fuzzy extractors.

[4]However, standard heuristics for estimating entropy can also be used to indicate whether a source has high-entropy samples. For a corpus of noisy signals, repeat the following a statistically significant number of times: 1) sample k indices 2) run the heuristic entropy test on the corpus which each sample restricted to the k indices.

Our Approach Our approach in all three constructions is different from most known constructions of fuzzy extractors, which put sufficient information in p to recover the original w from a nearby w' during Rep (this procedure is called a *secure sketch*). We deliberately do not recover w, because known techniques for building secure sketches do not work for sources whose entropy rate is lower than its error rate. (This is because they lose at least $\log |B_t|$ bits of entropy regardless of the source. This loss is necessary when the source is uniform [DORS08, Lemma C.1] or when reusability against a sufficiently rich class of correlations is desired [Boy04, Theorem 11]; computational definitions of secure sketches suffer from similar problems [FMR13, Corollary 1].) Instead, in the computational constructions, we lock up a freshly generated random r using parts of w in an error-tolerant way; in the information-theoretic construction, we reduce the alphabet in order to reduce the ball volume while maintaining entropy.

We note that the idea of locking up a random r has appeared in a prior theoretical construction of a computational fuzzy extractor for any source. Namely, Bitansky et al.. [BCKP14] show how to obfuscate a proximity point program that tests if an input w' is within distance t of the value w hidden inside the obfuscated program and, if so, outputs the secret r (such a program would be output by Gen as p and run by Rep). However, such a construction is based on very strong assumptions (semantically secure graded encodings [PST13]) and, in contrast to our construction, is highly impractical in terms of efficiency. Moreover, it is not known to provide reusability, because known obfuscation of proximity point programs is not known to be composable.

2 Definitions

For a random variables X_i over some alphabet $\mathcal{Z}$ we denote by $X = X_1, ..., X_n$ the tuple $(X_1, \ldots, X_n)$. For a set of indices J, X_J is the restriction of X to the indices in J. The set J^c is the complement of J. The *minentropy* of X is $\mathrm{H}_\infty(X) = -\log(\max_x \Pr[X = x])$, and the *average (conditional)* minentropy of X given Y is $\tilde{\mathrm{H}}_\infty(X|Y) = -\log(\mathbb{E}_{y\in Y} \max_x \Pr[X = x|Y = y])$ [DORS08, Section 2.4]. For a random variable W, let $H_0(W)$ be the logarithm of the size of the support of W, that is $H_0(W) = \log |\{w| \Pr[W = w] > 0\}|$. 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 we write the *computational distance* between X and Y as $\delta^D(X, Y) = |\mathbb{E}[D(X)] - \mathbb{E}[D(Y)]|$ (we extend it to a class of distinguishers $\mathcal{D}$ by taking the maximum over all distinguishers $D \in \mathcal{D}$). We denote by $\mathcal{D}_s$ the class of randomized circuits which output a single bit and have size at most s.

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 $|B_t|$ the size of a ball of radius t. We consider the Hamming metric over vectors in $\mathcal{Z}^n$, defined via $\mathsf{dis}(x, y) = |\{i|x_i \neq y_i\}|$. For this metric, $|B_t| = \sum_{i=0}^{t} \binom{n}{i}(|\mathcal{Z}| - 1)^i$. U_n denotes the uniformly distributed random variable on $\{0, 1\}^n$. Unless otherwise noted logarithms are base 2. Usually, we use capitalized letters for random variables and corresponding lowercase letters for their samples.

2.1 Fuzzy Extractors

In this section we define computational fuzzy extractors. Similar definitions for information-theoretic fuzzy extractors can be found in the work of Dodis et al. [DORS08, Sections 2.5–4.1]. The definition of computational fuzzy extractors allows for a small probability of error.

Definition 1. *[FMR13, Definition 4] Let $\mathcal{W}$ be a family of probability distributions over $\mathcal{M}$. A pair of randomized procedures "generate" (Gen) and "reproduce" (Rep) is an* $(\mathcal{M}, \mathcal{W}, \kappa, t)$-computational fuzzy

extractor *that is* $(\epsilon_{sec}, s_{sec})$*-hard with error* δ *if* Gen *and* Rep *satisfy the following properties:*

- *The generate procedure* Gen *on input* $w \in \mathcal{M}$ *outputs an extracted string* $r \in \{0,1\}^\kappa$ *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 if* $\mathsf{dis}(w, w') \le t$ *and* $(r, p) \leftarrow \mathsf{Gen}(w)$*, then* $\Pr[\mathsf{Rep}(w', p) = r] \ge 1 - \delta$*, where the probability is over the randomness of* (Gen, Rep).
- *The* security *property guarantees that for any distribution* $W \in \mathcal{W}$*, the string* r *is pseudorandom conditioned on* p*, that is* $\delta^{\mathcal{D}_{ssec}}((R, P), (U_\kappa, P)) \le \epsilon_{sec}$.

In the above definition, the errors are chosen before P: if the error pattern between w and w' depends on the output of Gen, then there is no guarantee about the probability of correctness. In Constructions 1 and 2 it is crucial that w' is chosen independently of the outcome of Gen.

Information-theoretic fuzzy extractors are obtained by replacing computational distance by statistical distance. We do make a second definitional modification. The standard definition of information-theoretic fuzzy extractors considers $\mathcal{W}$ consisting of all distributions of a given entropy. As described in the introduction, we construct fuzzy extractors for parameter regimes where it is impossible to provide security for all distributions with a particular minentropy. In both the computational and information-theoretic settings we consider a family of distributions $\mathcal{W}$.

2.2 Reusable Fuzzy Extractors

A desirable feature of fuzzy extractors is reusability [Boy04]. Intuitively, it is the ability to support multiple independent enrollments of the same value, allowing users to reuse the same biometric or PUF, for example, with multiple noncooperating providers.[5] More precisely, the algorithm Gen may be run multiple times on correlated readings $w_1, ..., w_\rho$ of a given source. Each time, Gen will produce a different pair of values $(r_1, p_1), ..., (r_\rho, p_\rho)$. Security for each extracted string r_i should hold even in the presence of all the helper strings $p_1, \ldots, p_\rho$ (the reproduction procedure Rep at the ith provider still obtains only a single w_i' close to w_i and uses a single helper string p_i). Because the multiple providers may not trust each other, a stronger security feature (which we satisfy) ensures that each r_i is secure even when all r_j for $j \neq i$ are also given to the adversary.

Our ability to construct reusable fuzzy extractors depends on the types of correlations allowed among $w_1, \ldots, w_\rho$. Boyen [Boy04] showed how to do so when each w_i is a shift of w_1 by a value that is oblivious to the value of w_1 itself (formally, w_i is a result of a transitive isometry applied to w_1). Boyen also showed that even for this weak class of correlations, any secure sketch must lose at least $\log |B_t|$ entropy [Boy04, Theorem 11].

We modify the definition of Boyen [Boy04, Definition 6] for the computational setting. We first present our definition and then compare to the definitions of Boyen.

Definition 2 (Reusable Fuzzy Extractors)**.** *Let* $\mathcal{W}$ *be a family of distributions over* $\mathcal{M}$*. Let* (Gen, Rep) *be a* $(\mathcal{M}, \mathcal{W}, \kappa, t)$*-computational fuzzy extractor that is* $(\epsilon_{sec}, s_{sec})$*-hard with error* δ*. Let* $(W^1, W^2, \ldots, W^\rho)$ *be* ρ *correlated random variables such that each* $W^j \in \mathcal{W}$*. Let* D *be an adversary. Define the following game for all* $j = 1, ..., \rho$*:*

[5] Reusability and unlinkability are two different properties. Unlinkability prevents an adversary from telling if two enrollments correspond to the same physical source [CS08, KBK+11]. We do not consider this property in this work.

- ***Sampling*** *The challenger samples* $w^j \leftarrow W^j$ *and* $u \leftarrow \{0,1\}^\kappa$.
- ***Generation*** *The challenger computes* $(r^j, p^j) \leftarrow \mathsf{Gen}(w^j)$.
- ***Distinguishing*** *The advantage of* D *is*

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

$(\mathsf{Gen}, \mathsf{Rep})$ *is* $(\rho, \epsilon_{sec}, s_{sec})$*-reusable if for all* $D \in \mathcal{D}_{s_{sec}}$ *and for all* $j = 1, ..., \rho$*, the advantage is at most* ϵ_{sec}.

Comparison with the definition of Boyen Boyen considers two versions of reusable fuzzy extractors. In the first version (called "outsider security" [Boy04, Definition 6]), the adversary sees $p^1, ..., p^\rho$ and tries to learn about the values $w^1, ..., w^\rho$ or the keys $r^1, ..., r^\rho$. This version is weaker than our version, because the adversary is not given any r^i values. In the second version (called "insider security" [Boy04, Definition 7]) , the adversary controls some subset of the servers and can run Rep on arbitrary $\tilde{p}^i$. This definition allows the adversary, in particular, to learn a subset of keys r^i (by performing key generation on the valid p^i), just like in our definition. However, it also handles the case when the p^i values are actively compromised. We do not consider such an active compromise attack. As explained in Section 1, protection against such an attack is called "robustness" and can be handled separately—for example, by techniques from [BDK+05, Theorem 1].

In Boyen's definitions, the adversary creates a perturbation function f^i after seeing $p^1, ..., p^{i-1}$ (and generated keys in case of insider security) and the challenger generates $w^i = f^i(w^1)$. The definition is parameterized by the class of allowed perturbation functions. Boyen constructs an outsider reusable fuzzy extractor for unbounded ρ when the perturbation family is a family of transitive isometries; Boyen then adds insider security using random oracles.

In contrast, instead of considering perturbation functions to generate w^i, we simply consider all tuples of distributions as long as each distribution is in $\mathcal{W}$, because we support arbitrary correlations among them.

3 Tools: Digital Lockers, Point Functions, and Hash Functions

Our main construction uses digital lockers, which are computationally secure symmetric encryption schemes that retain security even when used multiple times with correlated and weak (i.e., nonuniform) keys [CKVW10]. In a digital locker, obtaining any information about the plaintext from the ciphertext is as hard as guessing the key. They have the additional feature that the wrong key can be recognized as such (with high probability). We use notation $\mathsf{c} = \mathsf{lock}(\mathsf{key}, \mathsf{val})$ for the algorithm that performs the locking of the value val using the key key, and $\mathsf{unlock}(\mathsf{key}, \mathsf{c})$ for the algorithm that performs the unlocking (which will output val if key is correct and $\perp$ with high probability otherwise).

The following simple and efficient construction of digital lockers was shown to provide the desired security in the random oracle model of [BR93] by Lynn, Prabhakaran, and Sahai [LPS04, Section 4]. Let H be a cryptographic hash function, modeled as a random oracle. The locking algorithm $\mathsf{lock}(\mathsf{key}, \mathsf{val})$ outputs the pair $\mathsf{nonce}, H(\mathsf{nonce}, \mathsf{key}) \oplus (\mathsf{val}||0^{\mathsf{s}})$, where nonce is a nonce, $||$ denotes concatenation, and s is a security parameter. As long as the entropy of key is superlogarithmic, the adversary has negligible probability of finding the correct key; and if the adversary doesn't find the correct key, then the adversarial

knowledge about key and val is not significantly affected by this locker. Concatenation with 0^{s} is used to make sure that unlock can tell (with certainty $1 - 2^{-\mathsf{s}}$) when the correct value is unlocked.

It is seems plausible that in the standard model (without random oracles), specific cryptographic hash functions, if used in this construction, will provide the necessary security [CD08, Section 3.2], [Dak09, Section 8.2.3]. Moreover, Bitansky and Canetti [BC10], building on the work of [CD08, CKVW10], show how to obtain composable digital lockers based on a strong version of the Decisional Diffie-Hellman assumption without random oracles.

The security of digital lockers is defined via virtual-grey-box simulatability [BC10], where the simulator is allowed unbounded running time but only a bounded number of queries to the ideal locker. Intuitively, the definition gives the primitive we need: if the keys to the ideal locker are hard to guess, the simulator will not be able to unlock the ideal locker, and so the real adversary will not be able to, either. Formally, let $\mathsf{idealUnlock}(\mathsf{key}, \mathsf{val})$ be the oracle that returns val when given key, and $\perp$ otherwise.

Definition 3. *The pair of algorithm* $(\mathsf{lock}, \mathsf{unlock})$ *with security parameter* λ *is an* ℓ*-composable secure digital locker with error* γ *if the following hold:*

- ***Correctness*** *For all* key *and* val, $\Pr[\mathsf{unlock}(\mathsf{key}, \mathsf{lock}(\mathsf{key}, \mathsf{val})) = \mathsf{val}] \geq 1 - \gamma$. *Furthermore, for any* $\mathsf{key}' \neq \mathsf{key}$, $\Pr[\mathsf{unlock}(\mathsf{key}', \mathsf{lock}(\mathsf{key}, \mathsf{val})) = \perp] \geq 1 - \gamma$.
- ***Security*** *For every PPT adversary* A *and every positive polynomial* p, *there exists a (possibly inefficient) simulator* S *and a polynomial* $q(\lambda)$ *such that for any sufficiently large* s, *any polynomially-long sequence of values* $(\mathsf{val}_i, \mathsf{key}_i)$ *for* $i = 1, \ldots, \ell$, *and any auxiliary input* $z \in \{0,1\}^*$,

$$\left| \Pr\left[A\left(z, \{\mathsf{lock}\left(\mathsf{key}_i, \mathsf{val}_i\right)\}_{i=1}^{\ell}\right) = 1 \right] - \Pr\left[S\left(z, \{|\mathsf{key}_i|, |\mathsf{val}_i|\}_{i=1}^{\ell}\right) = 1 \right] \right| \leq \frac{1}{p(\mathsf{s})}$$

  where S *is allowed* $q(\lambda)$ *oracle queries to the oracles* $\{\mathsf{idealUnlock}(\mathsf{key}_i, \mathsf{val}_i)\}_{i=1}^{\ell}$.

Point Functions In one of the constructions for large alphabets, we use a weaker primitive: an obfuscated point function. This primitive can be viewed as a digital locker without the plaintext: it simply outputs 1 if the key is correct and 0 otherwise. Such a function can be easily constructed from the digital locker above with the empty ciphertext, or from a strong version of the Decisional Diffie-Hellman assumption [Can97]. We use notation $\mathsf{c} = \mathsf{lockPoint}(\mathsf{key})$ and $\mathsf{unlockPoint}(\mathsf{key}, \mathsf{c})$; security is defined the same way as for digital lockers with a fixed plaintext.

4 Main Result: Reusable Construction for Sources with High-Entropy Samples

Sources with High-Entropy Samples Let the source $W = W_1, \ldots, W_n$ consist of strings of length n over some arbitrary alphabet $\mathcal{Z}$ (the case of greatest interest is that of the binary alphabet $\mathcal{Z} = \{0,1\}$; however, we describe the construction more generally). For some parameters k, α, we say that the source W is a source with α*-entropy* k*-samples* if $\tilde{\mathrm{H}}_\infty(W_{j_1}, \ldots, W_{j_k} \,|\, j_1, \ldots, j_k) \geq \alpha$ for uniformly random $1 \leq j_1, \ldots, j_k \leq n$. See Section 1 for a discussion of how sources with this property come up naturally.

The Sample-then-Lock Construction The construction first chooses a random r to be used as the output of the fuzzy extractor. It then samples a random subset of symbols $v_1 = w_{j_1}, ..., w_{j_k}$ and creates a digital locker that hides r using v_1.[6] This process is repeated to produce some number ℓ of digital lockers all containing r, each unlockable with $v_1, ..., v_\ell$, respectively. The use of the composable digital lockers allows us to sample multiple times, because we need to argue only about individual entropy of V_i. Composability also allows reusability.[7]

Note that the output r can be as long as the digital locker construction can handle (in particular, the constructions discussed in Section 3 allow r to be arbitrarily long). Also note that it suffices to have r that is as long as a seed for a pseudorandom generator, because a longer output can be obtained by running this pseudorandom generator on r.

Construction 1 (Sample-then-Lock)**.** *Let $\mathcal{Z}$ be an alphabet, and let $W = W_1, ..., W_n$ be a source with α-entropy k-samples, where each W_j is over $\mathcal{Z}$. Let ℓ be a parameter, to be determined later. Let* $\mathsf{lock}, \mathsf{unlock}$ *be an ℓ-composable secure digital locker with error γ (for κ-bit values and keys over $\mathcal{Z}^k$). Define* $\mathsf{Gen}, \mathsf{Rep}$ *as:*

Gen

1. *Input:* $w = w_1, ..., w_n$
2. *Sample* $r \xleftarrow{\$} \{0,1\}^\kappa$.
3. *For* $i = 1, ..., \ell$:
   - (i) *Choose uniformly random* $1 \leq j_{i,1}, ..., j_{i,k} \leq n$
   - (ii) *Set* $v_i = w_{j_{i,1}}, ..., w_{j_{i,k}}$.
   - (iii) *Set* $\mathsf{c}_i = \mathsf{lock}(v_i, r)$.
   - (iv) *Set* $p_i = \mathsf{c}_i, (j_{i,1}, ..., j_{i,k})$.
4. *Output* (r, p), *where* $p = p_1 \dots p_\ell$.

Rep

1. *Input:* $(w' = w'_1, ..., w'_n, p = p_1 \dots p_\ell)$
2. *For* $i = 1, ..., \ell$:
   - (i) *Parse* p_i *as* $\mathsf{c}_i, (j_{i,1}, ..., j_{i,k})$.
   - (ii) *Set* $v'_i = w'_{j_{i,1}}, ..., w'_{j_{i,k}}$.
   - (iii) *Set* $r_i = \mathsf{unlock}(v'_i, \mathsf{c}_i)$. *If* $r_i \neq \perp$ *output* r_i.
3. *Output* $\perp$.

How to Set Parameters: Correctness vs. Efficiency Tradeoff To instantiate Construction 1, we need to choose a value for ℓ. Recall we assume that $\mathsf{dis}(w, w') \leq t$. For any given i, the probability that $v'_i = v_i$ is at least $(1 - \frac{t}{n})^k$. Therefore, the probability that no v'_i matches during Rep, causing Rep output to $\perp$, is at most

$$\left(1 - \left(1 - \frac{t}{n}\right)^k\right)^\ell .$$

In addition, Rep may be incorrect due to an error in one of the lockers, which happens with probability at most $\ell \cdot \gamma$. Thus, to make the overall error probability less than fuzzy extractor's allowable error parameter δ we need to set ℓ so that

$$\left(1 - \left(1 - \frac{t}{n}\right)^k\right)^\ell + \ell \cdot \gamma \leq \delta.$$

[6] We present and analyze the construction with uniformly random subsets; however, if necessary, it is possible to substantially decrease the required public randomness and the length of p by using more sophisticated samplers. See [Gol11] for an introduction to samplers.

[7] For the construction to be reusable ρ times the digital locker must be composable $\ell \cdot \rho$ times.

This provides a way to set ℓ to get a desirable δ, given a digital locker with error γ and source parameters n, t, k.

To get a bit more insight, we need to simplify the above expression. We can use the approximation $e^x \approx 1 + x$ to get

$$\left(1 - \left(1 - \frac{t}{n}\right)^k\right)^\ell \approx (1 - e^{-\frac{tk}{n}})^\ell \approx \exp(-\ell e^{-\frac{tk}{n}}).$$

The value γ can be made very small very cheaply in known locker constructions, so let us assume that γ is small enough so that $\ell \cdot \gamma \leq \delta/2$. Then if $tk = cn \ln n$ for some constant c, setting $\ell \approx n^c \log \frac{2}{\delta}$ suffices.

We now provide the formal statement of security for Construction 1; we consider reusability of this construction below, in Theorem 2.

Theorem 1. *Let λ be a security parameter, Let $\mathcal{W}$ be a family of sources over $\mathcal{Z}^n$ with α-entropy k-samples for $\alpha = \omega(\log \lambda)$. Then for any $s_{sec} = \mathtt{poly}(\lambda)$ there exists some $\epsilon_{sec} = \mathtt{ngl}(\lambda)$ such that Construction 1 is a $(\mathcal{Z}^n, \mathcal{W}, \kappa, t)$-computational fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$-hard with error $\delta = (1 - (1 - \frac{t}{n})^k)^\ell + \ell\gamma \approx \exp(-\ell e^{-\frac{tk}{n}}) + \ell\gamma$. (See above for an expression of ℓ as a function the other parameters.)*

Proof. Correctness is already argued above. We now argue security.

Our goal is to show that for all $s_{sec} = \mathtt{poly}(\lambda)$ there exists $\epsilon_{sec} = \mathtt{ngl}(\lambda)$ such that $\delta^{\mathcal{D}_{s_{sec}}}((R, P), (U, P)) \leq \epsilon_{sec}$. Fix some polynomial s_{sec} and let D be a distinguisher of size at most s_{sec}. We want to bound

$$|\,\mathbb{E}[D(R, P)] - \mathbb{E}[D(U, P)]|$$

by a negligible function.

We proceed by contradiction: suppose this difference is not negligible. That is, suppose that there is some polynomial $p(\cdot)$ such that for all λ_0 there exists some $\lambda > \lambda_0$ such that

$$|\,\mathbb{E}[D(R, P)] - \mathbb{E}[D(U, P)]| > 1/p(\lambda).$$

We note that λ is a function of λ_0 but we omit this notation for the remainder of the proof for clarity.

By the security of digital lockers (Definition 3), there is a polynomial q and an unbounded time simulator S (making at most $q(\lambda)$ queries to the oracles $\{\mathsf{idealUnlock}(v_i, r)\}_{i=1}^\ell$) such that

$$\left|\mathbb{E}[D(R, P_1, ..., P_\ell)] - \mathbb{E}\left[S^{\{\mathsf{idealUnlock}(v_i,r)\}_{i=1}^\ell}\left(R, \{j_{i,1}, ..., j_{i,k}\}_{i=1}^\ell, k, \kappa\right)\right]\right| \leq \frac{1}{3p(\lambda)}. \tag{1}$$

The same is true if we replaced R above by an independent uniform random variable U over $\{0,1\}^\kappa$. We now prove the following lemma, which shows that S cannot distinguish between R and U.

Lemma 1. *Let U denote the uniform distribution over $\{0,1\}^\kappa$. Then*

$$\begin{aligned}\Big|\,&\mathbb{E}\left[S^{\{\mathsf{idealUnlock}(v_i,r)\}_{i=1}^\ell}\left(R, \{j_{i,1}, ..., j_{i,k}\}_{i=1}^\ell, k, \kappa\right)\right] \\ -\,&\mathbb{E}\left[S^{\{\mathsf{idealUnlock}(v_i,r)\}_{i=1}^\ell}\left(U, \{j_{i,1}, ..., j_{i,k}\}_{i=1}^\ell, k, \kappa\right)\right]\Big| \\ &\leq \frac{q(q+1)}{2^\alpha} \leq \frac{1}{3p(\lambda)},\end{aligned} \tag{2}$$

where q is the maximum number of queries S can make.

Proof. Fix any $u \in \{0,1\}^\kappa$ (the lemma will follow by averaging over all u). Let r be the correct value of R. The only information about whether the value is r or u can obtained by S through the query responses. First, modify S slightly to quit immediately if it gets a response not equal to $\perp$ (such S is equally successful at distinguishing between r and u, because the first non-$\perp$ response tells S if its input is equal to the locked value r, and subsequent responses add nothing to this knowledge; formally, it is easy to argue that for any S, there is an S' that quits after the first non-$\perp$ response and is just as successful). There are $q+1$ possible values for the view of S on a given input (q of those views consist of some number of $\perp$ responses followed by the first non-$\perp$ response, and one view has all q responses equal to $\perp$). By [DORS08, Lemma 2.2b], $\tilde{\mathrm{H}}_\infty(V_i|View(S), \{j_{ik}\}) \geq \tilde{\mathrm{H}}_\infty(V_j|\{j_{ik}\}) - \log(q+1) \geq \alpha - \log(q+1)$. Therefore, at each query, the probability that S gets a non-$\perp$ answer (equivalently, guesses V_i) is at most $(q+1)2^{-\alpha}$. Since there are q queries of S, the overall probability is at most $q(q+1)/2^\alpha$. Then since 2^α is $\mathtt{ngl}(\lambda)$, there exists some λ_0 such that for all $\lambda > \lambda_0$, $q(q+1)/2^\alpha \leq 1/(3p(\lambda))$. □

Adding together Equation 1, Equation 2, and Equation 1 in which R is replaced with U, we obtain that

$$\delta^D((R,P),(U,P)) \leq \frac{1}{p(\lambda)}.$$

This is a contradiction and completes the proof of Theorem 1. □

Reusability of Construction 1 The reusability of Construction 1 follows from the security of digital clockers. Consider any ρ number of reuses. For each fixed $i \in \{1,...,\rho\}$, we can treat the keys $r^1,\ldots,r^{i-1},r^{i+1},\ldots,r^\rho$ and the sampled positions as auxiliary input to the digital locker adversary. The result follows by simulatability of this adversary, using the same argument as the proof of Theorem 1 above. Note that this argument now requires the digital locker to be $\rho \cdot \ell$-composable.

Theorem 2. *Fix ρ and let all the variables be as in Theorem 1, except that* $(\mathsf{lock}, \mathsf{unlock})$ *is an $\ell \cdot \rho$-composable secure digital locker (for κ-bit values and keys over $\mathcal{Z}^k$). Then for all $s_{sec} = \mathtt{poly}(n)$ there exists some $\epsilon_{sec} = \mathtt{ngl}(n)$ such that Construction 1 is $(\rho, \epsilon_{sec}, s_{sec})$-reusable fuzzy extractor.*

Comparison with work of [ST09] The work of Škorić and Tuyls [ST09] can be viewed as a fuzzy extractor that places the entire string into a single digital locker (in their paper, they use the language of hash functions). Their Rec procedure symbol searches for a nearby value that unlocks the digital locker, limiting Rec to a polynomial number of error patterns. We use a subset of symbols to lock and take multiple samples, greatly increasing the error tolerance.

5 Additional Constructions for the Case of Large Alphabets

In this section we provide additional constructions of fuzzy extractors that exploit the structure of the distribution w (instead of working for all distributions of a particular min-entropy). As stated in the introduction, both constructions work for low entropy rates when w comes from a large source alphabet $\mathcal{Z}$.

5.1 Construction for Sources with Sparse High-Entropy Marginals

In this section, we consider an alternative construction that is suited to sources over large alphabets. Intuitively, we use single symbols of w to lock bits of a secret that we then transform into r; we use

error-correcting codes to handle bits of the secret that cannot be retrieved due to errors in w'. Our main technical tool is obfuscated point functions (a weaker primitive than digital lockers; see Section 3 for the definition).

This construction requires enough symbols individually to contain sufficient entropy, but does not require independence of symbols, or even "fresh" entropy from them. Unlike the previous construction, it tolerates a linear fraction of errors (but over a larger alphabet, where errors may be more likely.). However, it cannot work for small alphabets, and is not reusable.

Sources with Sparse High-Entropy Marginals This construction works for distributions $W = W_1, ..., W_n$ over $\mathcal{Z}^n$ in which enough symbols W_j are unpredictable even after adaptive queries to equality oracles for other symbols. This quality of a distribution is captured in the following definition.

Definition 4. *Let* $\mathsf{idealUnlock}(\mathsf{key})$ *be an oracle that returns* 1 *when given* key *and* 0 *otherwise. A source* $W = W_1, ..., W_n$ *has* β*-sparse* α*-entropy* q*-marginals if there exists a set* $J \subset \{1, ..., n\}$ *of size at least* $n - \beta$ *such that for any unbounded adversary* S*,*

$$\forall j \in J, \tilde{\mathrm{H}}_\infty(W_j | View(S(\cdot))) \geq \alpha.$$

where S *is allowed* q *queries to the oracles* $\{\mathsf{idealUnlock}(W_i)\}_{i=1}^n$*.*

We show some examples of such sources in Appendix A.4. In particular, any source W where for all j, $\mathrm{H}_\infty(W_j) \geq \alpha = \omega(\log \lambda)$ (but all symbols may arbitrarily correlated) is a source with sparse high-entropy marginals (Proposition 3).

The Error-Correct-and-Obfuscate Construction This construction is inspired by the construction of Canetti and Dakdouk [CD08]. Instead of having large parts of the string w unlock r, we have individual symbols unlock bits of the output.

Before presenting the construction we provide some definitions from error correcting codes. We use error-correct codes over $\{0,1\}^n$ which correct up to t bit flips from 0 to 1 but no bit flips from 1 to 0 (this is the Hamming analog of the Z-channel [TABB02]).[8]

Definition 5. *Let* $e, c \in \{0,1\}^n$ *be vectors. Let* $x = \mathsf{Err}(c, e)$ *be defined as follows*

$$x_i = \begin{cases} 1 & c_i = 1 \vee e_i = 1 \\ 0 & \textit{otherwise.} \end{cases}$$

Definition 6. *A set* C *(over* $\{0,1\}^n$*) is a* (t, δ_{code})*-*Z *code if there exists an efficient procedure* Decode *such that*

$$\forall e \in \{0,1\}^n | \mathsf{Wgt}(e) \leq t, \Pr_{c \in C}[\mathsf{Decode}(\mathsf{Err}(c, e)) \neq c] \leq \delta_{code}.$$

Construction 2 (Lock-and-Error-Correct)**.** *Let* $\mathcal{Z}$ *be an alphabet and let* $W = W_1, ..., W_n$ *be a distribution over* $\mathcal{Z}^n$*. Let* $C \subset \{0,1\}^n$ *be* (t, δ_{code})*-*Z *code. Let* $\mathsf{lockPoint}, \mathsf{unlockPoint}$ *be an* n*-composable secure obfuscated point function with error* γ *(for keys over* $\mathcal{Z}$*). Define* $\mathsf{Gen}, \mathsf{Rep}$ *as:*

[8] Any code that corrects t Hamming errors also corrects t $0 \to 1$ errors, but more efficient codes exist for this type of error [TABB02]. Codes with $2^{\Theta(n)}$ codewords and $t = \Theta(n)$ over the binary alphabet exist for Hamming errors and suffice for our purposes (first constructed by Justensen [Jus72]). These codes also yield a constant error tolerance for $0 \to 1$ bit flips. The class of errors we support in our source (t Hamming errors over a large alphabet) and the class of errors for which we need codes (t $0 \to 1$ errors) are different.

Gen

1. *Input:* $w = w_1, ..., w_n$
2. *Sample* $c \leftarrow C$.
3. *For* $j = 1, ..., n$:
   (i) *If* $c_j = 0$:
   Let $p_j = \mathsf{lockPoint}(w_j)$.
   (ii) *Else:* $r_j \xleftarrow{\$} \mathcal{Z}$.
   Let $p_j = \mathsf{lockPoint}(r_j)$.
4. *Output* (c, p), *where* $p = p_1 \dots p_n$.

Rep

1. *Input:* (w', p)
2. *For* $j = 1, ..., n$:
   (i) *If* $\mathsf{unlockPoint}(w'_j, p_j) = 1$: *set* $c'_j = 0$.
   (ii) *Else: set* $c'_j = 1$.
3. *Set* $c = \mathsf{Decode}(c')$.
4. *Output* c.

As presented, Construction 2 is not yet a computational fuzzy extractor. The codewords c are not uniformly distributed and it is possible to learn some bits of c (for the symbols of W without much entropy). However, we can show that c looks like it has entropy to a computationally bounded adversary who knows p. Applying a randomness extractor with outputs over $\{0,1\}^\kappa$ (technically, an average-case computational randomness extractor) to c, and adding the extractor seed to p, will give us the desired fuzzy extractor. See Appendix A.1 for the formal details.

Construction 2 is secure if no distinguisher can tell whether it is working with r_j or w_j. By the security of point obfuscation, anything learnable from the obfuscation is learnable from oracle access to the function. Therefore, our construction is secure as long as enough symbols are unpredictable even after adaptive queries to equality oracles for individual symbols, which is exactly the property satisfied by sources with sparse high-entropy marginals.

The following theorem formalizes this intuition (proof in Appendix A.2).

Theorem 3. *Let λ be a security parameter. Let $\mathcal{Z}$ be an alphabet. Let $\mathcal{W}$ be a family of sources with β-sparse $\alpha = \omega(\log \lambda)$-entropy q-marginals over $\mathcal{Z}^n$, for any $q = \mathtt{poly}(n)$. Furthermore, let C be a (t, δ_{code})-Z code over $\mathcal{Z}^n$. Then for any $s_{sec} = \mathtt{poly}(n)$ there exists some $\epsilon_{sec} = \mathtt{ngl}(n)$ such that Construction 2, followed by a κ-bit randomness extractor (whose required input entropy is $\leq H_0(C) - \beta$), is a $(\mathcal{Z}^n, \mathcal{W}, \kappa, t)$-computational fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$-hard with error $\delta_{code} + n(1/|\mathcal{Z}| + \gamma)$.*

Entropy vs. Error Rate The minimum entropy necessary to satisfy Definition 4 is $\omega(\log \lambda)$ (for example, when all symbols are completely dependent but are all individually unguessable). The construction corrects a constant fraction of errors. When $n = \lambda^{1/c}$ then the entropy is smaller than the number of errors $m = \omega(\log \lambda) < \Theta(n) = \lambda^{1/c}$.

Output Length The extractor that follows Construction 2 can output $H_0(C) - \beta - 2\log(1/\epsilon_{sec})$ bits using standard information-theoretic techniques (such as the average-case leftover hash lemma [DORS08, Lemma 2.2b, Lemma 2.4]). To get a longer output, Construction 2 can be run multiple (say, μ) times with the same input and independent randomness to get multiple values c, concatenate them, and extract from the concatenation, to obtain an output of sufficient length $\mu(H_0(C) - \beta) - 2\log(1/\epsilon_{sec})$. The goal is to get an output long enough to use as a pseudorandom generator seed: once the seed is obtained, it can be used to generate arbitrary polynomial-length r, just like Construction 1.

Further Improvement If most codewords have Hamming weight close to $1/2$, we can decrease the error tolerance needed from the code from t to about $t/2$, because roughly half of the mismatches between w and w' occur where $c_j = 1$.

Lack of Reusability Even though Construction 2 uses composable obfuscated point functions, it is *not* reusable. Definition 4 allows sources with some "weak" symbols that can be completely learned by an adversary observing p. If a source is enrolled multiple times this partial information may add up over time to reveal the original value w_1. In contrast, Construction 1, leaks no partial information for the supported sources, allowing reusability.

5.2 Information-Theoretic Construction for Sparse Block Sources

The construction in this section has information-theoretic security, in contrast to only computational security of the previous two constructions. It uses symbol-by-symbol condensers to reduce the alphabet size while preserving most of the entropy, and then applies a standard fuzzy extractor to the resulting string.

This construction requires less entropy from each symbol than the previous construction; however, it places more stringent independence requirements on the symbols. It tolerates a linear number of errors.

Sparse Block Sources This construction works for sources $W = W_1, ..., W_n$ over $\mathcal{Z}^n$ in which enough symbols W_j contribute fresh entropy conditioned on previous symbols. We call this such sources *sparse block sources*, weakening the notion of block sources (introduced by Chor and Goldreich [CG88]), which require every symbol to contribute fresh entropy.

Definition 7. *A distribution $W = W_1, ..., W_n$ is an (α, β)-sparse block source if there exists a set of indices J where $|J| \geq n - \beta$ such that the following holds:*

$$\forall j \in J, \forall w_1, ..., w_{j-1} \in W_1, ..., W_{j-1}, \mathrm{H}_\infty(W_j | W_1 = w_1, ..., W_{j-1} = w_{j-1}) \geq \alpha.$$

The choice of conditioning on the past is arbitrary: a more general sufficient condition is that there exists some ordering of indices where most items have entropy conditioned on all previous items in this ordering (for example, is possible to consider a sparse reverse block source [Vad03]).

The Condense-then-Fuzzy-Extract Construction The construction first condenses entropy from each symbol of the source and then applies a fuzzy extractor to the condensed symbols. We'll denote the fuzzy extractor on the smaller alphabet as $(\mathsf{Gen}', \mathsf{Rep}')$. A condenser is like a randomness extractor but the output is allowed to be slightly entropy deficient. Condensers are known with smaller entropy loss than possible for randomness extractors (e.g. [DPW14]).

Definition 8. *A function $\mathtt{cond} : \mathcal{Z} \times \{0,1\}^d \to \mathcal{Y}$ is a $(m, \tilde{m}, \epsilon)$-randomness condenser if whenever $\mathrm{H}_\infty(W) \geq m$, then there exists a distribution Y with $\mathrm{H}_\infty(Y) \geq \tilde{m}$ and $(\mathtt{cond}(W, seed), seed) \approx_\epsilon (Y, seed)$.*

The main idea of the construction is that errors are "corrected" on the large alphabet (before condensing) while the entropy loss for the error correction is incurred on a smaller alphabet (after condensing).

Construction 3. *Let $\mathcal{Z}$ be an alphabet and let $W = W_1, ..., W_n$ be a distribution over $\mathcal{Z}^n$. We describe $\mathsf{Gen}, \mathsf{Rep}$ as follows:*

Gen

1. *Input:* $w = w_1, ..., w_n$
2. *For* $j = 1, ..., n$:
   (i) *Sample* $seed_i \leftarrow \{0,1\}^d$.
   (ii) *Set* $v_i = \mathtt{cond}(w_i, seed_i)$.
3. *Set* $(r, p') \leftarrow \mathsf{Gen}'(v_1, ..., v_n)$.
4. *Set* $p = (p', seed_1, ..., seed_n)$.
5. *Output* (r, p).

Rep

1. *Input:* $(w', p = (p', seed_1, ..., seed_n))$
2. *For* $j = 1, ..., n$:
   (i) *Set* $v'_i = \mathtt{cond}(w'_i, seed_i)$.
3. *Output* $r = \mathsf{Rep}'(v', p')$.

The following theorem shows the security of this construction (proof in Appendix B).

Theorem 4. *Let $\mathcal{W}$ be a family of $(\alpha = \Omega(1), \beta \leq n(1 - \Theta(1)))$-sparse block sources over $\mathcal{Z}^n$ and let $\mathtt{cond} : \mathcal{Z} \times \{0,1\}^d \to \mathcal{Y}$ be a $(\alpha, \tilde{\alpha}, \epsilon_{cond})$-randomness conductor. Define $\mathcal{V}$ as the family of all distributions with minentropy at least $\tilde{\alpha}(n - \beta)$ and let $(\mathsf{Gen}', \mathsf{Rep}')$ be $(\mathcal{Y}^n, \mathcal{V}, \kappa, t, \epsilon_{fext})$-fuzzy extractor with error δ.*[9] *Then $(\mathsf{Gen}, \mathsf{Rep})$ is a $(\mathcal{Z}^n, \mathcal{W}, \kappa, t, n\epsilon_{cond} + \epsilon_{fext})$-fuzzy extractor with error δ.*

Overcoming Proposition 1 Proposition 1 shows that no fuzzy extractor can be secure for all sources of a given minentropy $m < \log |B_t|$. Construction 3 supports sparse block sources whose overall entropy is less than $\log |B_t|$. The structure of a sparse block source implies that $\mathrm{H}_\infty(W) \geq \alpha(n - \beta) = \Theta(n)$. We assume that $\mathrm{H}_\infty(W) = \Theta(n)$. Using standard fuzzy extractors (for $\mathsf{Gen}', \mathsf{Rep}'$) it is possible to correct $t = \Theta(n)$ errors, yielding $\log |B_t| > \Theta(n)$ when $|\mathcal{Z}| = \omega(1)$.

Acknowledgements

The authors are grateful to Nishanth Chandran, Nir Bitansky, Sharon Goldberg, Gene Itkis, Bhavana Kanukurthi, and Mayank Varia for helpful discussions, creative ideas, and important references. The authors also thank the anonymous referees for useful feedback on the paper.

The work of A.S. was performed while at Boston University's Hariri Institute for Computing and RISCS Center, and Harvard University's "Privacy Tools" project.

Ran Canetti is supported by the NSF MACS project, an NSF Algorithmic foundations grant 1218461, the Check Point Institute for Information Security, and ISF grant 1523/14. Omer Paneth is additionally supported by the Simons award for graduate students in theoretical computer science. The work of Benjamin Fuller is sponsored in part by US NSF grants 1012910 and 1012798 and 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. Leonid Reyzin is supported in part by US NSF grants 0831281, 1012910, 1012798, and 1422965. Adam Smith is supported in part by NSF awards 0747294 and 0941553.

[9] We actually need $(\mathsf{Gen}', \mathsf{Rep}')$ to be an average case fuzzy extractor (see [DORS08, Definition 4] and the accompanying discussion). Most known constructions of fuzzy extractors are average-case fuzzy extractors. For simplicity we refer to $\mathsf{Gen}', \mathsf{Rep}'$ as simply a fuzzy extractor.

References

[Ash65] Robert Ash. *Information Theory*. Interscience Publishers, 1965.

[BA12] Marina Blanton and Mehrdad Aliasgari. On the (non-) reusability of fuzzy sketches and extractors and security improvements in the computational setting. *IACR Cryptology ePrint Archive*, 2012:608, 2012.

[BA13] Marina Blanton and Mehrdad Aliasgari. Analysis of reusability of secure sketches and fuzzy extractors. *IEEE transactions on information forensics and security*, 8(9-10):1433–1445, 2013.

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

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

[BCKP14] Nir Bitansky, Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On virtual grey box obfuscation for general circuits. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, 2014.

[BDK+05] Xavier Boyen, Yevgeniy Dodis, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Secure remote authentication using biometric data. In *EUROCRYPT*, pages 147–163. Springer, 2005.

[BH09] Marina Blanton and William MP Hudelson. Biometric-based non-transferable anonymous credentials. In *Information and Communications Security*, pages 165–180. Springer, 2009.

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

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[BS00] Sacha Brostoff and M.Angela Sasse. Are passfaces more usable than passwords?: A field trial investigation. *People and Computers*, pages 405–424, 2000.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *Advances in Cryptology-CRYPTO'97*, pages 455–469. Springer, 1997.

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

[CDF+08] Ronald Cramer, Yevgeniy Dodis, Serge Fehr, Carles Padró, and Daniel Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In *Advances in Cryptology–EUROCRYPT 2008*, pages 471–488. Springer, 2008.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2), 1988.

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

[CS08] F Carter and A Stoianov. Implications of biometric encryption on wide spread use of biometrics. In *EBF Biometric Encryption Seminar (June 2008)*, 2008.

[Dak09] Ramzi Ronny Dakdouk. *Theory and Application of Extractable Functions*. PhD thesis, Yale University, 2009. `http://www.cs.yale.edu/homes/jf/Ronny-thesis.pdf`.

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

[DKK+12] Yevgeniy Dodis, Bhavana Kanukurthi, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. *IEEE Transactions on Information Theory*, 58(9):6207–6222, 2012.

[DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 621–630. ACM, 2009.

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

[DPW14] Yevgeniy Dodis, Krzysztof Pietrzak, and Daniel Wichs. Key derivation without entropy waste. In *Advances in Cryptology–EUROCRYPT 2014*, pages 93–110. Springer, 2014.

[EHMS00] Carl Ellison, Chris Hall, Randy Milbert, and Bruce Schneier. Protecting secret keys with personal entropy. *Future Generation Computer Systems*, 16(4):311–318, 2000.

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

[GCVDD02] Blaise Gassend, Dwaine Clarke, Marten Van Dijk, and Srinivas Devadas. Silicon physical random functions. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 148–160. ACM, 2002.

[Gol11] Oded Goldreich. A sample of samplers: A computational perspective on sampling. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 302–332. Springer, 2011.

[HILL99] 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.

[HLR07] Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In *EUROCRYPT*, pages 169–186, 2007.

[HMSS12] Matthias Hiller, Dominik Merli, Frederic Stumpf, and Georg Sigl. Complementary ibs: Application specific error correction for PUFs. In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 1–6. IEEE, 2012.

[HR05] Thomas Holenstein and Renato Renner. One-way secret-key agreement and applications to circuit polarization and immunization of public-key encryption. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 478–493. Springer, 2005.

[Jus72] Jørn Justesen. Class of constructive asymptotically good algebraic codes. *Information Theory, IEEE Transactions on*, 18(5):652–656, 1972.

[KBK+11] Emile JC Kelkboom, Jeroen Breebaart, Tom AM Kevenaar, Ileana Buhan, and Raymond NJ Veldhuis. Preventing the decodability attack based cross-matching in a fuzzy commitment scheme. *Information Forensics and Security, IEEE Transactions on*, 6(1):107–121, 2011.

[KLRW14] Patrick Koeberl, Jiangtao Li, Anand Rajan, and Wei Wu. Entropy loss in PUF-based key generation schemes: The repetition code pitfall. In *Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on*, pages 44–49. IEEE, 2014.

[KR09] Bhavana Kanukurthi and Leonid Reyzin. Key agreement from close secrets over unsecured channels. In *EUROCRYPT*, pages 206–223, 2009.

[Kra10] Hugo Krawczyk. Cryptographic extraction and key derivation: The HKDF scheme. In *Advances in Cryptology–CRYPTO 2010*, pages 631–648. Springer, 2010.

[KZ07] Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, 2007.

[LPS04] Benjamin Lynn, Manoj Prabhakaran, and Amit Sahai. Positive results and techniques for obfuscation. In *Advances in Cryptology–EUROCRYPT 2004*, pages 20–39. Springer, 2004.

[Mau93] Ueli M. Maurer. Secret key agreement by public discussion from common information. *IEEE Transactions on Information Theory*, 39(3):733–742, 1993.

[Mau97] Ueli M. Maurer. Information-theoretically secure secret-key agreement by NOT authenticated public discussion. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 209–225. Springer, 1997.

[MG09] Rene Mayrhofer and Hans Gellersen. Shake well before use: Intuitive and secure pairing of mobile devices. *IEEE Transactions on Mobile Computing*, 8(6):792–806, 2009.

[MRW02] Fabian Monrose, Michael K Reiter, and Susanne Wetzel. Password hardening based on keystroke dynamics. *International Journal of Information Security*, 1(2):69–83, 2002.

[MTV09] Roel Maes, Pim Tuyls, and Ingrid Verbauwhede. Low-overhead implementation of a soft decision helper data algorithm for SRAM PUFs. In *Cryptographic Hardware and Embedded Systems-CHES 2009*, pages 332–347. Springer, 2009.

[MW96] Ueli M. Maurer and Stefan Wolf. Towards characterizing when information-theoretic secret key agreement is possible. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology - ASIACRYPT '96, International Conference on the Theory and Applications of Cryptology and Information Security, Kyongju, Korea, November 3-7, 1996, Proceedings*, volume 1163 of *Lecture Notes in Computer Science*, pages 196–209. Springer, 1996.

[MW97] Ueli M. Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 307–321. Springer, 1997.

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

[PPJ03] Salil Prabhakar, Sharath Pankanti, and Anil K Jain. Biometric recognition: Security and privacy concerns. *IEEE Security & Privacy*, 1(2):33–42, 2003.

[PRTG02] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. Physical one-way functions. *Science*, 297(5589):2026–2030, 2002.

[PST13] Rafael Pass, Karn Seth, and Sidharth Telang. Obfuscation from semantically-secure multilinear encodings. Cryptology ePrint Archive, Report 2013/781, 2013. `http://eprint.iacr.org/`.

[SD07] 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.

[ST09] B. Skoric and P. Tuyls. An efficient fuzzy extractor for limited noise. Cryptology ePrint Archive, Report 2009/030, 2009. `http://eprint.iacr.org/`.

[STP09] Koen Simoens, Pim Tuyls, and Bart Preneel. Privacy weaknesses in biometric sketches. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 188–203. IEEE, 2009.

[TABB02] Luca G Tallini, Sulaiman Al-Bassam, and Bella Bose. On the capacity and codes for the Z-channel. In *IEEE International Symposium on Information Theory*, page 422, 2002.

[TSS+06] Pim Tuyls, Geert-Jan Schrijen, Boris Skoric, Jan Geloven, Nynke Verhaegh, and Rob Wolters. Read-proof hardware from protective coatings. In *Cryptographic Hardware and Embedded Systems - CHES 2006*, pages 369–383. 2006.

[Vad03] Salil P Vadhan. On constructing locally computable extractors and cryptosystems in the bounded storage model. In *Advances in Cryptology-CRYPTO 2003*, pages 61–77. Springer, 2003.

[YD10] Meng-Day Mandel Yu and Srinivas Devadas. Secure and robust error correction for physical unclonable functions. *IEEE Design & Test*, 27(1):48–65, 2010.

[ZH93] Moshe Zviran and William J. Haga. A comparison of password techniques for multilevel authentication mechanisms. *The Computer Journal*, 36(3):227–237, 1993.

A Analysis of Construction 2

A.1 Computational Fuzzy Conductors and Computational Extractors

In this section we introduce tools necessary to convert Construction 2 to a computation fuzzy extractor. We first define an object weaker than a computational fuzzy extractor: it outputs a key with computational entropy (instead of a pseudorandom key). We call this object a computational fuzzy conductor. It is the computational analogue of a fuzzy conductor (introduced by Kanukurthi and Reyzin [KR09]). Before defining this object, we define conditional computational "HILL" ([HILL99]) entropy.

Definition 9. *[HLR07, Definition 3] Let (W, S) be a pair of random variables. W has* HILL entropy *at least m conditioned on S, denoted $H^{\mathtt{HILL}}_{\epsilon_{sec}, s_{sec}}(W|S) \geq m$ if there exists a joint distribution (X, S), such that $\tilde{\mathrm{H}}_\infty(X|S) \geq m$ and $\delta^{\mathcal{D}_{s_{sec}}}((W, S), (X, S)) \leq \epsilon_{sec}$.*

Definition 10. *A pair of randomized procedures "generate" (*Gen*) and "reproduce" (*Rep*) is an $(\mathcal{M}, \mathcal{W}, \tilde{m}, t)$-computational fuzzy conductor that is $(\epsilon_{sec}, s_{sec})$-hard with error δ if* Gen *and* Rep *satisfy Definition 1, except the last condition is replaced with the following weaker condition:*

- *for any distribution $W \in \mathcal{W}$, the string r has high HILL entropy conditioned on P. That is $H^{\mathtt{HILL}}_{\epsilon_{sec}, s_{sec}}(R|P) \geq \tilde{m}$.*

Computational fuzzy conductors can be converted to computational fuzzy extractors (Definition 1) using standard techniques, as follows. The transformation uses a computational extractor. A computational extractor is the adaption of a randomness extractor to the computational setting. Any information-theoretic randomness extractor is also a computational extractor; however, unlike information-theoretic extractors, computational extractors can expand their output arbitrarily via pseudorandom generators once a long-enough output is obtained. We adapt the definition of Krawczyk [Kra10] to the average case:

Definition 11. *A function* $\mathtt{cext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\kappa$ *a $(m, \epsilon_{sec}, s_{sec})$*-average-case computational extractor *if for all pairs of random variables X, Y (with X over $\{0,1\}^n$) such that $\tilde{H}_\infty(X|Y) \geq m$, we have $\delta^{\mathcal{D}_{s_{sec}}}((\mathtt{cext}(X; U_d), U_d, Y), U_\kappa \times U_d \times Y) \leq \epsilon_{sec}$.*

Combining a computational fuzzy conductor and a computational extractor yields a computational fuzzy extractor:

Lemma 2. *Let* $(\mathsf{Gen}', \mathsf{Rep}')$ *be a $(\mathcal{M}, \mathcal{W}, \tilde{m}, t)$-computational fuzzy conductor that is $(\epsilon_{cond}, s_{cond})$-hard with error δ and outputs in $\{0,1\}^n$. Let* $\mathtt{cext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\kappa$ *be a $(\tilde{m}, \epsilon_{ext}, s_{ext})$-average case computational extractor. Define* $(\mathsf{Gen}, \mathsf{Rep})$ *as:*

- $\mathsf{Gen}(w; seed)$ *(where $seed \in \{0,1\}^d$): run $(r', p') = \mathsf{Gen}'(w)$ and output $r = \mathtt{cext}(r'; seed)$, $p = (p', seed)$.*
- $\mathsf{Rep}(w', (p', seed))$ *: run $r' = \mathsf{Rep}'(w'; p')$ and output $r = \mathtt{cext}(r'; seed)$.*

Then $(\mathsf{Gen}, \mathsf{Rep})$ *is a* $(\mathcal{M}, \mathcal{W}, \kappa, t)$*-computational fuzzy extractor that is* $(\epsilon_{cond} + \epsilon_{ext}, s')$*-hard with error* δ *where* $s' = \min\{s_{cond} - |\mathtt{cext}| - d, s_{ext}\}$.

Proof. It suffices to show if there is some distinguisher D' of size s' where

$$\delta^{D'}((\mathtt{cext}(X; U_d), U_d, P'), (U_\kappa, U_d, P')) > \epsilon_{cond} + \epsilon_{ext}$$

then there is an distinguisher D of size s_{cond} such that for all Y with $\tilde{\mathrm{H}}_\infty(Y|P') \geq \tilde{m}$,

$$\delta^{D}((X, P'), (Y, P')) \geq \epsilon_{cond}.$$

Let D' be such a distinguisher. That is,

$$\delta^{D'}(\mathtt{cext}(X, U_d) \times U_d \times P', U_\kappa \times U_d \times P') > \epsilon_{ext} + \epsilon_{cond}.$$

Then define D as follows. On input (y, p') sample $seed \leftarrow U_d$, compute $r \leftarrow \mathtt{cext}(y; seed)$ and output $D(r, seed, p')$. Note that $|D| \approx s' + |\mathtt{cext}| + d = s_{cond}$. Then we have the following:

$$\begin{aligned}
\delta^{D}((X, P'), (Y, P')) &= \delta^{D'}((\mathtt{cext}(X, U_d), U_d, P'), \mathtt{cext}(Y, U_d), U_d, P') \\
&\geq \delta^{D'}((\mathtt{cext}(X, U_d), U_d, P'), (U_\kappa \times U_d \times P')) \\
&\quad - \delta^{D'}((U_\kappa \times U_d \times P'), (\mathtt{cext}(Y, U_d), U_d, P')) \\
&> \epsilon_{cond} + \epsilon_{ext} - \epsilon_{ext} = \epsilon_{cond}.
\end{aligned}$$

Where the last line follows by noting that D' is of size at most s_{ext}. Thus D distinguishes X from all Y with sufficient conditional minentropy. This is a contradiction. □

A.2 Security of Construction 2

It suffices to prove that Construction 2 is a $(\mathcal{Z}^n, \mathcal{W}, \tilde{m} = H_0(C) - \beta, t)$-comp. fuzzy conductor, i.e., that C has HILL entropy $H_0(C) - \beta$ conditioned on P. The final extraction step will convert it to a computational fuzzy extractor (see Lemma 2).

The security proof of Construction 2 is similar to the security proof of Construction 1. However, it is made more complicated by the fact that the definition of sources with sparse high-entropy marginals (Definition 4) allows for certain weak symbols that can easily be guessed. This means we must limit our indistinguishable distribution to symbols that are difficult to guess. Security is proved via the following lemma:

Lemma 3. *Let all variables be as in Theorem 3. For every* $s_{sec} = \mathtt{poly}(n)$ *there exists some* $\epsilon_{sec} = \mathtt{ngl}(n)$ *such that* $H^{\mathtt{HILL}}_{\epsilon_{sec}, s_{sec}}(C|P) \geq H_0(C) - \beta$.

We give a brief outline of the proof, followed by the proof of the new statement. It is sufficient to show that there exists a distribution C' with conditional minentropy and $\delta^{\mathcal{D}_{s_{sec}}}((C, P), (C', P)) \leq \mathtt{ngl}(n)$. Let J be the set of indices that exist according to Definition 4. Define the distribution C' as a uniform codeword conditioned on the values of C and C' being equal on all indices outside of J. We first note that C' has sufficient entropy, because $\tilde{\mathrm{H}}_\infty(C'|P) = \tilde{\mathrm{H}}_\infty(C'|C_{J^c}) \geq \mathrm{H}_\infty(C', C_{J^c}) - H_0(C_{J^c}) = H_0(C) - |J^c|$ (the second step is by [DORS08, Lemma 2.2b]). It is left to show $\delta^{\mathcal{D}_{ssec}}((C, P), (C', P)) \leq \mathtt{ngl}(n)$. The outline for the rest of the proof is as follows:

- Let D be a distinguisher between (C, P) and (C', P). By the security of obfuscated point functions,

$$\left|\mathbb{E}[D(C, P_1, ..., P_n)] - \mathbb{E}\left[S^{\{\mathsf{idealUnlock}(\cdot)\}_{i=1}^n}(C, n \cdot |\mathcal{Z}|)\right]\right|$$

  is small.

- Show that even an unbounded S making a polynomial number of queries to the stored points cannot distinguish between C and C'. That is,

$$\left|\mathbb{E}\left[S^{\{\mathsf{idealUnlock}(\cdot)\}_{i=1}^n}(C, n \cdot |\mathcal{Z}|)\right] - \mathbb{E}\left[S^{\{\mathsf{idealUnlock}(\cdot)\}_{i=1}^n}(C', n \cdot |\mathcal{Z}|)\right]\right|$$

  is small.

- By the security of obfuscated point functions,

$$\left|\mathbb{E}\left[S^{\{\mathsf{idealUnlock}(\cdot)\}_{i=1}^n}(C', n \cdot |\mathcal{Z}|)\right] - \mathbb{E}[D(C', P_1, ..., P_n)]\right|$$

  is small.

Proof of Lemma 3. The overall approach and the proof of the first and third bullet as in Theorem 1. We only prove the second bullet. Define the distribution X as follows:

$$X_j = \begin{cases} W_j & C_j = 0 \\ R_j & C_j = 1. \end{cases}$$

Lemma 4. $\Delta\left(S^{\{\mathsf{idealUnlock}(X_i)\}_{i=1}^n}(C, n \cdot |\mathcal{Z}|), S^{\{\mathsf{idealUnlock}(X_i)\}_{i=1}^n}(C', n \cdot |\mathcal{Z}|)\right)$
$\leq (n - \beta)2^{-(\alpha+1)}$.

Proof. It suffices to show that for any two codewords that agree on J^c, the statistical distance is at most $(n - \beta)2^{-(\alpha+1)}$.

Lemma 5. *Let c^* be true value encoded in X and let c' a codeword in C'. Then,*

$$\Delta\left(S^{\{\mathsf{idealUnlock}(X_i)\}_{i=1}^n}(c^*, n \cdot |\mathcal{Z}|), S^{\{\mathsf{idealUnlock}(X_i)\}_{i=1}^n}(c', n \cdot |\mathcal{Z}|)\right) \leq (n - \beta)2^{-(\alpha+1)}.$$

Proof. Recall that for all $j \in J$, $\tilde{\mathrm{H}}_\infty(W_j | View(S)) \geq \alpha$. The only information about the correct value of c_j^* is contained in the query responses. When all responses are 0 the view of S is identical when presented with c^* or c'. We now show that for any value of c^* all queries on $j \in J$ return 0 with probability $1 - 2^{-\alpha+1}$. Suppose not. That is, suppose the probability of at least one nonzero response on index j is $> 2^{-(\alpha+1)}$. Since w, w' are independent of r_j, the probability of this happening when $c_j^* = 1$ is at most $q/\mathcal{Z}$ or equivalently $2^{-\log|\mathcal{Z}|+\log q}$. Thus, it must occur with probability:

$$\begin{aligned} 2^{-\alpha+1} &< \Pr[\text{non zero response location } j] \\ &= \Pr[c_j^* = 1]\Pr[\text{non zero response location } j \wedge c_j^* = 1] \\ &+ \Pr[c_j^* = 0]\Pr[\text{non zero response location } j \wedge c_j^* = 0] \\ &\leq 1 \times 2^{-\log|\mathcal{Z}|+\log q} + 1 \times \Pr[\text{non zero response location } j \wedge c_j^* = 0] \end{aligned} \tag{3}$$

We now show that for $\alpha \leq \log|\mathcal{Z}| - \log q$:

Claim 1. *If W is a source with β-sparse α-entropy q-marginals over $\mathcal{Z}$, then $\alpha \leq \log|\mathcal{Z}| - \log q$.*

Proof. Let $J \subset \{1, ..., n\}$ the set of good indices. It suffices to show that there exists an S making q queries such that for some

$$j \in J, \tilde{\mathrm{H}}_\infty(W_j | S^{\{\mathsf{idealUnlock}(X_i)\}_{i=1}^n}) \leq \log|\mathcal{Z}| - \log q.$$

Let $j \in J$ be some arbitrary element of J and denote by $w_{j,1}, ..., w_{j,q}$ the q most likely outcomes of W_j (breaking ties arbitrarily). Then $\sum_{i=1}^{q} \Pr[W_j = w_{j,i}] \geq q/|\mathcal{Z}|$. Suppose not. This means that there is some $w_{j,i}$ with probability $\Pr[W_j = w_{j,i}] < 1/|\mathcal{Z}|$. Since there are $\mathcal{Z} - q$ remaining possible values of W_j for their total probability to be at least $1 - q/|\mathcal{Z}|$ at least of these values has probability at least $1/\mathcal{Z}$. This contradicts the statement $w_{j,1}, ..., w_{j,q}$ are the most likely values. Consider S that queries the jth oracle on $w_{j,1}, .., w_{j,q}$. Denote by Bad the random variable when $W_j \in \{w_{j,1}, .., w_{j,q}\}$ After these queries the remaining minentropy is at most:

$$\begin{aligned}
&\tilde{\mathrm{H}}_\infty(W_j | S^{J_W(\cdot,\cdot)}) \\
&\quad = -\log\left(\Pr[Bad = 1] \times 1 + \Pr[Bad = 0] \times \max_w \Pr[W_j = w | Bad = 0]\right) \\
&\quad \leq -\log\left(\Pr[Bad = 1] \times 1\right) \\
&\quad = -\log\left(\frac{q}{|\mathcal{Z}|}\right) = \log|\mathcal{Z}| - \log q
\end{aligned}$$

This completes the proof of Claim 1. □

Rearranging terms in Equation 3, we have:

$$\Pr[\text{non zero response location } j \wedge c_j = 0] > 2^{-\alpha+1} - 2^{-(\log|\mathcal{Z}| - \log q)} = 2^{-\alpha}$$

When there is a 1 response and $c_j = 0$ this means that there is no remaining minentropy. If this occurs with over $2^{-\alpha}$ probability this violates the condition on W (Definition 4). By the union bound over the indices $j \in J$ the total probability of a 1 in J is at most $(n - \beta)2^{-\alpha+1}$. Recall that c^*, c' match on all indices outside of J. Thus, for all c^*, c' the statistical distance is at most $(n - \beta)2^{-\alpha+1}$. This concludes the proof of Lemma 5. □

Lemma 4 follows by averaging over all points in C'. □

□

A.3 Correctness of Construction 2

We now argue correctness of Construction 2. We first assume ideal functionality of the obfuscated point functions. Consider a coordinate j for which $c_j = 1$. Since w' is chosen independently of the points r_j, and r_j is uniform, $\Pr[r_j = w'_j] = 1/|\mathcal{Z}|$. Thus, the probability of at least one $1 \to 0$ bit flip (the random choice r_i being the same as w'_i) is $\leq n(1//|\mathcal{Z}|)$. Since there are most t locations for which $w_j \neq w'_j$ there are at most t $0 \to 1$ bit flips in c, which the code will correct with probability $1 - \delta_{code}$, because c was chosen uniformly. Finally, since each obfuscated point function is correct with probability $1 - \gamma$, Construction 2 is correct with error at most $\delta_{code} + n(1/|\mathcal{Z}| + \gamma)$.

A.4 Characterizing sources with sparse high-entropy marginals

Definition 4 is an inherently adaptive definition and a little unwieldy. In this section, we partially characterize sources that satisfy Definition 4. The majority of the difficulty in characterizing Definition 4 is that different symbols may be dependent, so an equality query on symbol i may reshape the distribution of symbol j. In the examples that follow we denote the adversary by S as the simulator in Definition 3. We first show some sources that have sparse high-entropy marginals (Section A.4.1) and then show sources with high overall entropy that do not have sparse high-entropy marginals (Section A.4.2).

A.4.1 Positive Examples

We begin with the case of independent symbols.

Proposition 2. *Let $W = W_1, ..., W_n$ be a source in which all symbols W_j are mutually independent. Let α be a parameter. Let $J \subset \{1, ..., n\}$ be a set of indices such that for all $j \in J$, $\mathrm{H}_\infty(W_j) \geq \alpha$. Then for any q, W is a source with $(n - |J|)$-sparse $(\alpha - \log(q+1))$-entropy q-marginals. In particular, when $\alpha = \omega(\log n)$ and $q = \mathtt{poly}(n)$, then W is a source with $(n - |J|)$-sparse $\omega(\log n)$-entropy q-marginals.*

Proof. It suffices to show that for all $j \in J, \tilde{\mathrm{H}}_\infty(W_j | View(S(\cdot))) = \alpha - \log(q+1)$ where S is allowed q queries to the oracles $\{\mathsf{idealUnlock}(W_i)\}_{i=1}^n$. We can ignore queries for all symbols but the jth, as the symbols are independent. Furthermore, without loss of generality, we can assume that no duplicate queries are asked, and that the adversary is deterministic (S can calculate the best coins). Let $A_1, A_2, \ldots A_q$ be the random variables representing the oracle answers for an adversary S making q queries about the ith symbol. Each A_k is just a bit, and at most one of them is equal to 1 (because duplicate queries are disallowed). Thus, the total number of possible responses is $q+1$. Thus, we have the following,

$$\begin{aligned}\tilde{\mathrm{H}}_\infty(W_j | View(S(\cdot))) &= \tilde{\mathrm{H}}_\infty(W_j | A_1, \ldots, A_q) \\ &= \mathrm{H}_\infty(W_j) - |A_1, \ldots, A_q| \\ &= \alpha - \log(q+1)\,,\end{aligned}$$

where the second line follows from the first by [DORS08, Lemma 2.2]. □

In their work on computational fuzzy extractors, Fuller, Meng, and Reyzin [FMR13] show a construction for symbol-fixing sources, where each symbol is either uniform or a fixed symbol (symbol-fixing sources were introduced by Kamp and Zuckerman [KZ07]). Proposition 2 shows that Definition 4 captures, in particular, this class of distributions. However, Definition 4 captures more distributions. We now consider more complicated distributions where symbols are not independent.

Proposition 3. *Let $f : \{0,1\}^e \to \mathcal{Z}^n$ be a function. Furthermore, let f_j denote the restriction of f's output to its jth coordinate. If for all j, f_j is injective then $W = f(U_e)$ is a source with 0-sparse $(e - \log(q+1))$-entropy q-marginals.*

Proof. f is injective on each symbol, so

$$\tilde{\mathrm{H}}_\infty(W_j | View(S)) = \tilde{\mathrm{H}}_\infty(U_e | View(S))\,.$$

Consider a query q_k on symbol j. There are two possibilities: either q_k is not in the image of f_j, or q_k can be considered a query on the preimage $f_j^{-1}(q_k)$. Then (by assuming S knows f) we can eliminate queries which correspond to the same value of U_e. Then the possible responses are strings with Hamming weight at most 1 (like in the proof of Claim 2), and by [DORS08, Lemma 2.2] we have for all j, $\tilde{\mathrm{H}}_\infty(W_j | View(S)) \geq \mathrm{H}_\infty(W_j) - \log(q+1)$. □

Note the total entropy of a source in Proposition 3 is e, so there is a family of distributions with total entropy $\omega(\log n)$ for which Construction 2 is secure. For these distributions, all the coordinates are as dependent as possible: one determines all others. We can prove a slightly weaker claim when the correlation between the coordinates W_j is arbitrary:

Proposition 4. *Let $W = W_1, ..., W_n$. Suppose that for all j, $\mathrm{H}_\infty(W_j) \geq \alpha$, and that $q \leq 2^\alpha/4$ (this holds asymptotically, in particular, if q is polynomial and α is super-logarithmic). Then W is a source with 0-sparse $(\alpha - 1 - \log(q+1))$-entropy q-marginals.*

Proof. Intuitively, the claim is true because the oracle is not likely to return 1 on any query. Formally, we proceed by induction on oracle queries, using the same notation as in the proof of Proposition 2. Our inductive hypothesis is that $\Pr[A_1 \neq 0 \vee \cdots \vee A_{i-1} \neq 0] \leq (i-1)2^{1-\alpha}$. If the inductive hypothesis holds, then, for each j,

$$\mathrm{H}_\infty(W_j | A_1 = \cdots = A_{i-1} = 0) \geq \alpha - 1\,. \tag{4}$$

This is true for $i = 1$ by the condition of the theorem. It is true for $i > 1$ because, as a consequence of the definition of H_∞, for any random variable X and event E, $\mathrm{H}_\infty(X|E) \geq \mathrm{H}_\infty(X) + \log \Pr[E]$; and $(i-1)2^{1-\alpha} \leq 2q2^{-\alpha} \leq 1/2$.

We now show that $\Pr[A_1 \neq 0 \vee \cdots \vee A_i \neq 0] \leq i2^{1-\alpha}$, assuming that $\Pr[A_1 \neq 0 \vee \cdots \vee A_{i-1} \neq 0] \leq (i-1)2^{1-\alpha}$.

$$\begin{aligned}
&\Pr[A_1 \neq 0 \vee \cdots \vee A_{i-1} \neq 0 \vee A_i \neq 0] \\
&\quad = \Pr[A_1 \neq 0 \vee \cdots \vee A_{i-1} \neq 0] + \Pr[A_1 = \cdots = A_{i-1} = 0 \wedge A_i = 1] \\
&\quad \leq (i-1)2^{1-\alpha} + \Pr[A_i = 1 \,|\, A_1 = \cdots = A_{i-1} = 0] \\
&\quad \leq (i-1)2^{1-\alpha} + \max_j 2^{-\mathrm{H}_\infty(W_j | A_1 = \cdots = A_{i-1} = 0)} \\
&\quad \leq (i-1)2^{1-\alpha} + 2^{1-\alpha} \\
&\quad = i2^{1-\alpha}
\end{aligned}$$

(where the third line follows by considering that to get $A_i = 1$, the adversary needs to guess some W_j, and the fourth line follows by (4)). Thus, using $i = q+1$ in (4), we know $\mathrm{H}_\infty(W_j | A_1 = \cdots = A_q = 0) \geq \alpha - 1$. Finally this means that

$$\begin{aligned}
\tilde{\mathrm{H}}_\infty(W_j | A_1, \ldots, A_q) &\geq -\log\left(2^{-\mathrm{H}_\infty(W_j | A_1 = \cdots = A_q = 0)} \Pr[A_1 = \cdots = A_q = 0] + 1 \cdot \Pr[A_1 \neq 0 \vee \cdots \vee A_q \neq 0]\right) \\
&\geq -\log\left(2^{-\mathrm{H}_\infty(W_j | A_1 = \cdots = A_q = 0)} + q2^{1-\alpha}\right) \\
&\geq -\log\left((q+1)2^{1-\alpha}\right) = \alpha - 1 - \log(q+1)\,.
\end{aligned}$$

□

A.4.2 Negative Examples

Propositions 2 and 3 rest on there being no easy "entry" point to the distribution. This is not always the case. Indeed it is possible for some symbols to have very high entropy but lose all of it after equality queries.

Proposition 5. *Let $p = (\mathtt{poly}(\lambda))$ and let $f_1, ..., f_n$ be injective functions where $f_j : \{0,1\}^{j \times \log p} \to \mathcal{Z}$.*[10] *Then define the distribution U_n and consider $W_1 = f_1(U_{1,...,\log p})$, $W_2 = f_2(U_{1,...,2\log p}), ..., W_n = f_n(U)$. There is an adversary making $p \times n$ queries such that $\tilde{\mathrm{H}}_\infty(W|View(S(\cdot))) = 0$.*

Proof. Let x be the true value for $U_{p \times n}$. We present an adversary S that completely determines x. S computes $y_1^1 = f_1(x_1^1), ..., y_1^p = f(x_1^p)$. Then S queries on $(y_1), ..., (y_p)$ to the first oracle, exactly one answer returns 1. Let this value be y_1^* and its preimage x_1^*. Then S computes $y_2^1 = f_2(x_1^*, x_2^1), ..., y_2^p = f_2(x_1^*, x_2^p)$ and queries $y_2^1, ..., y_2^p$. Again, exactly one of these queries returns 1. This process is repeated until all of x is recovered (and thus w). □

The previous example relies on an adversary's ability to determine a symbol from the previous symbols. We formalize this notion next. We define the entropy jump of a source as the remaining entropy of a symbol when previous symbols are known:

Definition 12. *Let $W = W_1, ..., W_n$ be a source under ordering $i_1, ..., i_n$. The* jump *of a symbol i_j is $\mathtt{Jump}(i_j) = \max_{w_{i_1}, ..., w_{i_{j-1}}} H_0(W_{i_j} | W_{i_1} = w_{i_1}, ..., W_{i_{j-1}} = w_{i_{j-1}})$.*

An adversary who can learn symbols in succession can eventually recover the entire secret. In order for a source to have sparse high-entropy marginals, the adversary must get "stuck" early enough in this recovery process. This translates to having a super-logarithmic jump early enough.

Proposition 6. *Let W be a distribution and let q be a parameter, if there exists an ordering $i_1, ..., i_n$ such that for all $j \leq n - \beta + 1$, $\mathtt{Jump}(i_j) = \log q/(n - \beta + 1)$, then W is not a source with β-sparse high-entropy q-marginals.*

Proof. For convenience relabel the ordering that violates the condition as $1, ..., n$. We describe an unbounded adversary S that determines $W_1, ..., W_{n-\beta+1}$. As before S queries the q/n possible values for W_1 and determines W_1. Then S queries the (at most) $q/(n - \beta + 1)$ possible values for $W_2|W_1$. This process is repeated until $W_{n-\beta+1}$ is learned. □

Presenting a sufficient condition for security is more difficult as S may interleave queries to different symbols. It seems like the optimum strategy for S is to focus on a single symbol at a time, but it is unclear how to formalize this intuition.

B Analysis of Construction 3

Proof. Let $W \in \mathcal{W}$. It suffices to argue correctness and security. We first argue correctness.

Correctness: When $w_i = w_i'$, then $\mathtt{cond}(w_i, seed_i) = \mathtt{cond}(w_i', seed_i)$ and thus $v_i = v_i'$. Thus, for all w, w' where $\mathsf{dis}(w, w') \leq t$, then $\mathsf{dis}(v, v') \leq t$. Then by correctness of $(\mathsf{Gen}', \mathsf{Rep}')$, $\Pr[(r, p) \leftarrow \mathsf{Gen}'(v) \wedge r' \leftarrow \mathsf{Rep}(v', p) \wedge r' = r] \geq 1 - \delta$.

Security: We now argue security. Denote by *seed* the random variable consisting of all n seeds and V the entire string of generated $V_1, ..., V_n$. To show that

$$R|P, seed \approx_{n\epsilon_{cond} + \epsilon_{fext}} U|P, seed,$$

it suffices to show that $\tilde{\mathrm{H}}_\infty(V|seed)$ is $n\epsilon_{cond}$ close to a distribution with average minentropy $\tilde{\alpha}(n - \beta)$. The lemma then follows by the security of $(\mathsf{Gen}', \mathsf{Rep}')$.[11]

[10] Here we assume that $|\mathcal{Z}| \geq n \times \log p$, that is the source has a small number of symbols.

[11] Note, again, that $(\mathsf{Gen}', \mathsf{Rep}')$ must be an average-case fuzzy extractor. Most known constructions are average-case and we omit this notation.

We now argue that there exists a distribution Y where $\tilde{\mathrm{H}}_\infty(Y|seed) \geq \tilde{\alpha}(n-\beta)$ and $(V, seed_1, ..., seed_n) \approx (Y, seed_1, .., seed_n)$. First note since W is (α, β) sparse block source that there exists a set of indices J where $|J| \geq n - \beta$ such that the following holds:

$$\forall j \in J, \forall w_1, ..., w_{j-1} \in W_1, ..., W_{j-1}, \mathrm{H}_\infty(W_j | W_1 = w_1, ..., W_{j-1} = w_{j-1}) \geq \alpha.$$

Then consider the first element of $j_1 \in J$, $\forall w_1, ..., w_{j_1-1} \in W_1, ..., W_{j_1-1}$,

$$\mathrm{H}_\infty(W_{j_1} | W_1 = w_1, ..., W_{j_1-1} = w_{j_1-1}) \geq \alpha.$$

Thus, there exists a distribution Y_{j_1} with $\tilde{\mathrm{H}}_\infty(Y_{j_1}|seed_{j_1}) \geq \tilde{\alpha}$ such that

$$(\mathtt{cond}(W_{j_1}, seed_{j_1}), seed_{j_1}, W_1, ..., W_{j_1-1}) \approx_{\epsilon_{cond}} (Y_{j_1}, seed_{j_1}, W_1, ..., W_{j_1-1})$$

and since $(seed_1, ..., seed_{j_1})$ are independent of these values

$$(\mathtt{cond}(W_{j_1}, seed_{j_1}), W_{j_1-1}, ..., W_1, seed_{j_1}, ..., seed_1) \approx_{\epsilon_{cond}} (Y_{j_1}, W_{j_1-1}, ..., W_1, seed_{j_1}, , ..., seed_1)\,.$$

Consider the random variable

$$Z_{j_1} = (Y_{j_1}, \mathtt{cond}(W_{j_1-1}, seed_{j_1-1}), ..., \mathtt{cond}(W_1, seed_1))$$

and note that

$$\tilde{\mathrm{H}}_\infty(Z_{j_1}|seed_1, ..., seed_{j_1}) \geq \alpha'.$$

Applying a deterministic function does not increase statistical distance and thus,

$$\begin{aligned}(\mathtt{cond}(W_{j_1}, seed_{j_1}), \mathtt{cond}(W_{j_1-1}, seed_{j_1-1}), .., \mathtt{cond}(W_1, seed_1), seed_{j_1}, ..., seed_1)& \\ \approx_{n\epsilon_{cond}} (Z_{j_1}, seed_{j_1}, ..., seed_1)&\end{aligned}$$

By a hybrid argument there exists a distribution Z with $\tilde{\mathrm{H}}_\infty(Z|seed) \geq \tilde{\alpha}(n-\beta)$ where

$$\begin{aligned}(\mathtt{cond}(W_n, seed_n), ..., \mathtt{cond}(W_1, seed_1), seed_n, ..., seed_1)& \\ \approx_{n\epsilon_{cond}} (Z, seed_n, ..., seed_1).&\end{aligned}$$

This completes the proof of Theorem 4. □