Key Derivation From Noisy Sources With More Errors Than Entropy

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

December 1, 2014

Abstract

Fuzzy extractors (Dodis et al., Eurocrypt 2004) convert repeated noisy readings of a high-entropy 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. This helper string reduces the entropy of the original secret—in the worst case, by as much as the logarithm of the number of tolerated error patterns. For many practical sources of secrets, reliability demands that the number of tolerated error patterns is large, making this loss greater than the original entropy of the secret. We say that such sources have *more errors than entropy*. Most known approaches for building fuzzy extractors cannot be used for such sources.

We provide constructions of fuzzy extractors for large classes of sources with more errors than entropy. Our constructions exploit the structural properties of a source in addition to its entropy guarantees. Some are made possible by relaxing the security requirement from information-theoretic to computational.

Reusable fuzzy extractors (Boyen, CCS 2004) remain secure even when the initial enrollment phase is repeated multiple times with the same or correlated secrets, producing multiple helper strings. By relying on computational security, we construct the first reusable fuzzy extractors that make no assumption about how multiple readings of the source are correlated.

Keywords Fuzzy extractors, reusability, key derivation, error-correcting codes, computational entropy, 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, TSv+06, GCVDD02, SD07], and quantum information [BBR88]. Turning similar readings into identical values is known as *information reconciliation*; further converting those values into 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,

*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; work performed while at Boston University's Hariri Institute for Computing and RISCS Center, and Harvard University's "Privacy Tools" project.

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* is the primitive that accomplishes both information reconciliation and privacy amplification non-interactively; fuzzy extractors are defined in [DORS08].

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 correctness guarantee is that r will be correctly reproduced by Rep as long as w' is no farther than t from w in some metric space. In this work, we consider the Hamming metric. The security guarantee is that r produced by Gen is close to uniform (information-theoretically [DORS08] or computationally [FMR13]), even given p. This guarantee holds as long as w comes from a high-quality distribution, which traditionally has been defined as *any* distribution with sufficient min-entropy m.

Reusable Fuzzy Extractors An additional desirable security property of fuzzy extractors, introduced by Boyen [Boy04], is called reusability. This property is necessary if 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 organizations. Each of them 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 $p_1, \ldots, p_q$ (and, in case some organizations turn out to compromised or adversarial, a stronger security notion requires security for r_i even in the presence of r_j for $j \neq i$). Many traditional fuzzy extractors are not reusable [Boy04, STP09, BA12, BA13].

Limitations of Known Approaches Constructions of fuzzy extractors are limited by the tension between security and correctness guarantees: if we allow for higher error tolerance t, then we also need higher starting entropy m. The reason for this tension is simple: if an adversary who knows p can guess some w' within distance t of w, then it will be able to easily obtain the true r by running Rep. In fact, if t is high enough that there are 2^m points in a ball of radius t, then there exists a distribution of w of min-entropy m *contained entirely in a single ball*. For this distribution, an adversary can run Rep on the center of this ball and always learn the key r. Thus, if the security guarantee of a given fuzzy extractor holds for *any* source of a given min-entropy m and the correctness guarantees holds for any t errors, then m must be greater than $\log |B_t|$, where $|B_t|$ denotes the number of points in a ball of radius t. This condition on the source holds regardless of whether the fuzzy extractor is information-theoretic or computational, and extends even to the interactive setting. If a source fails this condition, we will says that it has *more errors than entropy*.

Unfortunately, sources that have been proposed as prime candidates for authentication have more errors than entropy. 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 that more errors than entropy [BH09, Section 5]. PUFs with slightly nonuniform outputs suffer from similar problems [KLRW14].

The situation with reusability is even worse: the only known construction of reusable fuzzy extractors [Boy04] requires very particular relationships between w_i values, which are unlikely to hold in any practical source.

Our Contributions We provide the first constructions of fuzzy extractors that can be used for large classes of sources that have more errors than entropy. Our constructions work for Hamming errors for strings w of length γ over some alphabet $\mathcal{Z}$. Naturally, as argued above, these constructions cannot work for all sources of a given entropy; each construction comes with a constraint on the sources for which it

	Security	Source Structure	Error-Tolerance	Alphabet Size
Cons. 3.2	Not reusable Info-theoretic	Most symbols contribute entropy	Constant fraction	Super constant
Cons. 4.1	Reusable Computational	Most symbols contribute entropy	Sub constant fraction	Super constant
Cons. 5.3	Not reusable Computational	Symbols correlated but hard to guess	Constant fraction	Super polynomial

Table 1: Summary of new constructions. All constructions support families of distributions with more errors than entropy.

is secure. Table 1 summarizes our constructions. Our first construction provides information-theoretic security. It can correct a constant fraction of errors, but requires that a constant fraction of the symbols contribute fresh entropy, even conditioned on previous symbols (Definition 3.3).

We switch to computational security to obtain constructions with additional features. Our second construction provides reusability (against computationally bounded adversaries). The reusability we obtain is very strong: security holds even if the multiple readings w_i used in Gen are *arbitrarily correlated*, as long as each w_i *individually* comes from an allowed distribution. The allowed distributions include those that are supported in the first construction, as well as other distributions, such as those with k-wise independence among symbols for superlogarithmic k. This construction requires that the fraction of errors is subconstant.

Our third construction removes the need for fresh entropy in the symbols and allows a constant fraction of symbols of errors, at the cost of requiring a large alphabet size (super-polynomial in the security parameter). It is secure if symbols in w each have individual super-logarithmic min-entropy, even if they are arbitrarily correlated. Moreover, a constant fraction of symbols in w may have little or no entropy, as long as knowledge of their values does not reduce the entropy of the high-entropy symbols too much (see Definition 5.4).

Our Approach Most known constructions of fuzzy extractors put sufficient information in p to recover the original w from a nearby w' during Rep (this procedure is called a *secure sketch*), and then apply a randomness extractor to w to get r. Unfortunately, the current techniques for building secure sketches do not work for sources with more errors than entropy, because they lose at least $\log |B_t|$ bits of entropy regardless of the source. Moreover, 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].

Additionally, computational definitions of security suffer from similar problems [FMR13, Corollary 3.8, Theorem 3.10]. Thus, we take a different approach and do not attempt to recover w.

Our first construction reduces the alphabet size by hashing each input symbol (which comes from a large alphabet) into a much smaller set, so that the resulting hash value has low entropy deficiency. The intuition behind this approach is that it reduces the size of B_t by reducing the alphabet size, but preserves a sufficient portion of the input entropy. The resulting string no longer has more errors than entropy. We then apply an information-theoretic fuzzy extractor to the resulting string.

Our second construction, which is computationally secure, is based on obfuscated digital lockers [CD08]. Digital lockers output a secret value only when given the correct input to "unlock" the secret. An obfuscated digital locker does not provide information about the locked value or how to unlock it. The main idea of the construction is to pick a random r and lock r in a digital locker that is unlocked by a random

subset of the symbols of w. To tolerate errors in the input, this process is repeated several times, so that at least one digital locker can be unlocked using w'. We use obfuscation in a way that does not leak partial information; this is crucial to arguing reusability.

Finally, our third construction tolerates more errors than the second because it uses digital lockers that are unlocked by single symbols of w. Since we do not assume that every symbol has high individual entropy, hiding an entire r in every locker then becomes too risky, Instead, we hide a single bit per locker. To tolerate errors, these bits come from an error correcting code. To ensure an adversary who learns some bits doesn't learn anything useful about r, we don't encode r in the error-correcting code, but rather extract r (using an information-theoretic [NZ93] or computational [Kra10] extractor) from the decoded string.

The Required Notion of Obfuscation Our constructions use simulation-secure obfuscation of digital lockers, however, we do not require full-fledged virtual black-box obfuscation [BGI$^+$01]. Instead, we rely on the relaxed notion of *virtual grey-box* obfuscation [BC10]. We also require that the obfuscation remains secure even when several digital lockers of correlated points are composed. Bitanski and Canetti constructed composable digital lockers with virtual grey-box security under particular number-theoretic assumptions [BC10].

Connection to General Obfuscation We note that fuzzy extractors for sources with more errors than entropy can be trivially constructed from virtual grey-box obfuscation for the class of *proximity point programs*. A proximity point program $I_w(x)$ tests if x is within distance t of w. Recently, Bitansky et al. [BCKP14] constructed such obfuscation based on the strong assumption of *semantically secure graded encodings* [PST13]. The construction of Bitansky et al. is based on multilinear encoding and is highly impractical. Our constructions use obfuscated digital lockers. Obfuscated digital lockers are instantiable under significantly weaker assumptions and can be implemented quite efficiently. Additionally, the known obfuscation for proximity point programs is not known to be composable and therefore does not yield a reusable fuzzy extractor.

Open Problems All of our constructions support more errors than entropy by using two metrics spaces. Errors are tolerated in one metric space and corrected in a second. To handle more errors than entropy, we map to a metric space where multiple error patterns are grouped together. All our constructions require the first metric space to have a super-constant size alphabet. An alternative approach to the problem may support constant size alphabets.

In this work we restrict the distribution of the original reading w and allow w' to be an arbitrary point within distance t. An alternative approach is to restrict the set of w' where Gen produces the correct key.

Organization The remainder of this paper is organized as follows: we cover notation and fuzzy extractors in Section 2. We present our information-theoretic construction in Section 3 and our two computational constructions in Sections 4 and 5.

2 Preliminaries

For a random variables X_i over some alphabet $\mathcal{Z}$ we denote by $X = X_1, ..., X_\gamma$ the tuple $(X_1, \ldots, X_\gamma)$. 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 *min-entropy* of X is $\mathrm{H}_\infty(X) = -\log(\max_x \Pr[X = x])$, and the *average (conditional)* min-entropy 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 $\mathcal{Z}^\gamma$, defined via $\mathsf{dis}(x, y) = \{i|x_i \neq y_i\}$. For this metric, $|B_t| = \sum_{i=0}^{t} \binom{\gamma}{i}(|\mathcal{Z}| - 1)^i$. 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. 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 2.1. *[FMR13, Definition 2.5] 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 $(\mathsf{Gen}, \mathsf{Rep})$. If $\mathsf{dis}(w, w') > t$, then no guarantee is provided about the output of Rep.*
- *The* security *property guarantees that for any distribution $W \in \mathcal{W}$, the string r is pseudorandom conditioned on p, that is $\delta^{\mathcal{D}_{s_{sec}}}((R, P), (U_\kappa, P)) \le \epsilon_{sec}$.*

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 4.1 and 5.3 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, it is impossible to provide security for all distributions with more errors than entropy. In both the computational and information-theoretic settings we consider a family of distributions $\mathcal{W}$.

2.1.1 Reusable Fuzzy Extractors

An additional desirable feature of fuzzy extractors is reusability [Boy04]. Inuitively, 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 noncommunicating providers. More precisely, the algorithm Gen may be run multiple times on correlated readings $w_1, ..., w_q$ of a given source. Each time, Gen will produce a different pair of values $(r_1, p_1), ..., (r_q, p_q)$. Security for each extracted string r_i should hold even in the presence of all the helper strings $p_1, \ldots, p_q$ (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 $w_1, \ldots, w_q$. 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 discuss the our definition and definitions due to Boyen in Appendix A.2.

Definition 2.2 (Reusable Fuzzy Extractors)**.** *Let $\mathcal{W}$ be a family of distributions over $\mathcal{M}$. Let $(\mathsf{Gen}, \mathsf{Rep})$ be a $(\mathcal{M}, \mathcal{W}, \kappa, t)$-computational fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$-hard with error δ. Fix some $W_1 \in \mathcal{W}$. Let $f_2, .., f_q, D$ be a split adversary. Define the following game for all $j = 1, ..., q$:*

- ***Sampling*** *The challenger samples $w_1 \leftarrow W_1, u \leftarrow \{0,1\}^\kappa$.*
- ***Perturbation*** *For $i = 2, ..., q$: the challenger computes $(r_i, p_i) \leftarrow \mathsf{Gen}(w_i)$. Set $w_{i+1} = f_i(w_1, p_1, ..., p_i)$.*
- ***Distinguishing*** *The advantage of D is*

$$\begin{aligned} Adv(D) \stackrel{def}{=} &\Pr[D(r_1, ..., r_{j-1}, r_j, r_{j+1}, ..., r_q, p_1, ..., p_q) = 1] \\ &- \Pr[D(r_1, ..., r_{j-1}, u, r_{j+1}, ..., r_q, p_1, ..., p_q) = 1]. \end{aligned}$$

$(\mathsf{Gen}, \mathsf{Rep})$ is $(q, \epsilon_{sec}, s_{sec}, f_2, ..., f_q)$-reusable if for all $D \in \mathcal{D}_{s_{sec}}$ the advantage is at most ϵ_{sec}.

The definition is parameterized by $f_2, ..., f_q$. This adversary implicitly defines distributions $W_2, ..., W_q$ (which depend on W_1 and the public values $P_1, ...P_i$). Security seems hopeless if fuzzy extractor is not secure on each of these distributions on their own. This is the only requirement we make on these functions. We call these types of functions admissible:

Definition 2.3. *Let $(\mathsf{Gen}, \mathsf{Rep})$ be a $(\mathcal{M}, \mathcal{W}, \kappa, t)$-computational fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$-hard with error δ. In the reusability game above, we say a set of functions $f_2, ..., f_q$ are* admissible *if for all $W_1 \in \mathcal{W}$ for all $w_1 \in W_1$ and $\forall p_1, ..., p_q$ that are the public outputs of Gen the distribution $W_{i,w_1,p_1,...,p_{i-1}} = f_i(w_1, p_1, ..., p_{i-1})$ is a member of $\mathcal{W}$.*

2.2 Obfuscation

Our constructions will use obfuscation for two types of circuits: point functions and digital lockers. The family of point functions $\mathtt{I}_n = \{I_w\}_{w \in \{0,1\}^n}$ defined as follows:

$$I_w(x) : \begin{cases} 1 & x = w \\ 0 & \text{otherwise} \end{cases}.$$

and the class of digital lockers is $\mathtt{I}_n = \{I_{w,r}\}_{w\in\{0,1\}^n, r\in\{0,1\}^\kappa}$ defined as follows:

$$I_{w,r}(x) : \begin{cases} r & x = w \\ \perp & \text{otherwise} \end{cases}.$$

The required notion of obfuscation is virtual grey-box (VGB) introduced in [BC10]. This notion is weaker then the standard notion of virtual black-box ([BGI+01]), as it allows the simulator to run in unbounded time while making at a polynomial number of oracle queries to the function. VGB obfuscators for point functions and digital lockers are constructible from specific number-theoretic assumptions or by strong assumptions on hash functions. We provide more details and a formal definition in Appendix A.3.

3 Supporting more errors than entropy

In this section we show an information-theoretic fuzzy extractor that supports more errors than entropy. The construction first condenses entropy from each block of the source and then applies a different fuzzy extractor to the condensed blocks. 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 3.1. *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 3.2. *Let* $\mathcal{Z}$ *be an alphabet and let* $W = W_1, ..., W_\gamma$ *be a distribution over* $\mathcal{Z}^\gamma$*. We describe* $\mathsf{Gen}, \mathsf{Rep}$ *as follows:*

Gen

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

Rep

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

For Construction 3.2 to be secure we need most blocks to contribute some entropy to the output. We call this notion a partial block source.

Definition 3.3. *A distribution* $W = W_1, ..., W_\gamma$ *is an* (α, β)*-partial block source if there exists a set of indices* J *where* $|J| \geq \gamma - \beta$ *such that the following holds:*

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

Definition 3.3 is a weakening of block sources (introduced by Chor and Goldreich [CG88]), as only some blocks are required to have entropy conditioned on the past. 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, a "partial" reverse block source [Vad03]). This construction is secure and it supports distributions with more errors than entropy (proof is in Appendix C).

Lemma 3.4. *Let $\mathcal{W}$ be the family of $(\alpha = \Omega(1), \beta \leq \gamma(1 - \Theta(1)))$-partial block sources over $\mathcal{Z}^\gamma$ 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 min-entropy at least $\tilde{\alpha}(\gamma - \beta)$ and let $(\mathsf{Gen}', \mathsf{Rep}')$ be $(\mathcal{Y}^\gamma, \mathcal{V}, \kappa, t, \epsilon_{fext})$-fuzzy extractor with error δ.*[1] *Then $(\mathsf{Gen}, \mathsf{Rep})$ is a $(\mathcal{Z}^\gamma, \mathcal{W}, \kappa, t, \gamma\epsilon_{cond} + \epsilon_{fext})$-fuzzy extractor with error δ.*

3.1 More errors than entropy

In this section we show that Construction 3.2 supports partial block sources with more errors than entropy. The structure of a partial block source implies that $\mathrm{H}_\infty(W) \geq \alpha(\gamma - \beta) = \Theta(\gamma)$. We assume that $\mathrm{H}_\infty(W) = \Theta(\gamma)$. The condenser of Dodis et al [DPW14] has a constant entropy loss, so $\alpha - \tilde{\alpha} = \Theta(1)$. This means that the input entropy to $(\mathsf{Gen}', \mathsf{Rep}')$ is $\Theta(\gamma)$. We assume that the new alphabet $\mathcal{Y}$ is of constant size. Standard fuzzy extractors on constant size alphabets correct a constant fraction of errors at a entropy loss of $\Theta(\gamma)$, yielding $\kappa = \Theta(\gamma)$. Thus, our construction is secure for distributions with more errors than entropy whenever $|\mathcal{Z}| = \omega(1)$. More formally:

$$\#\text{ Errors} - \text{Entropy} = \log|B_t| - \mathrm{H}_\infty(W) \geq t\log|\mathcal{Z}| - \Theta(\gamma) - = \Theta(\gamma)\log|\mathcal{Z}| - \Theta(\gamma) > 0$$

That is, there exists a super-constant alphabet size for which Construction 3.2 is secure with more errors than entropy.

4 Adding reusability

In the previous section, we showed it was possible to construct a fuzzy extractor for a family of distributions with more errors than entropy. Using computational techniques we are able to retain many of the advantages of Construction 3.2 and achieve a reusable fuzzy extractor.

The construction samples a random subset of blocks $W_{j_1}, ..., W_{j_\eta}$ and obfuscates the concatenation of these blocks. Denote this concatenated value by V_1. This process is repeated to produce $V_1, ..., V_\ell$ where at least one V_i should be correct to "unlock" the correct key. Let $\mathsf{Sample}_{\gamma,\eta}(\cdot)$ be an algorithm that outputs a random subset of $\{1, ..., \gamma\}$ of size η given let r_{sam} bits of randomness.

Construction 4.1 (Sample-then-Obfuscate)**.** *Let $\mathcal{Z}$ be an alphabet, and let $W = W_1, ..., W_\gamma$ be a source where each W_j is over $\mathcal{Z}$. Let η be a parameter, and $\mathcal{O}$ be an obfuscator for the family of digital lockers with κ-bit outputs. Define $\mathsf{Gen}, \mathsf{Rep}$ as:*

[1] 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.

Gen

1. *Input:* $w = w_1, ..., w_\gamma$
2. *Sample* $r \xleftarrow{\$} \{0,1\}^\kappa$.
3. *For* $i = 1, ..., \ell$:
   (i) *Select* $\lambda_i \xleftarrow{\$} \{0,1\}^{r_{sam}}$.
   (ii) *Set* $j_{i,1}, ..., j_{i,\eta} \leftarrow \mathsf{Sample}_{\gamma,\eta}(\lambda_i)$
   (iii) *Set* $v_i = w_{j_{i,1}}, ..., w_{j_{i,\eta}}$.
   (iv) *Set* $\rho_i = \mathcal{O}(I_{v_i,r})$.
   (v) *Set* $p_i = \rho_i, \lambda_i$.
4. *Output* (r, p), *where* $p = p_1 \dots p_\ell$.

Rep

1. *Input:* $(w' = w'_1, ..., w'_\gamma, p)$
2. *For* $i = 1, ..., \ell$:
   (i) *Parse* p_i *as* ρ_i, λ_i.
   (ii) *Set* $j_{i,1}, ..., j_{i,\eta} \leftarrow \mathsf{Sample}_{\gamma,\eta}(\lambda_i)$.
   (iii) *Set* $v'_i = w'_{j_{i,1}}, ..., w'_{j_{i,\eta}}$.
   (iv) *Set* $\rho_i(v'_i) = r_i$. *If* $r_i \neq \perp$ *output* r_i.
3. *Output* $\perp$.

The use of a computational primitive (obfuscation of digital lockers) allows us to sample multiple times, because we need to argue only about individual entropy of V_i, as opposed to the information-theoretic setting, where it would be necessary to argue about the entropy of the joint variable V. This is the property that allows reusability.

This construction uses a naïve sampler that takes truly random samples, but the public randomness may be substantially decreased by using more sophisticated samplers. (See Goldreich [Gol11] for an introduction to samplers.)

Theorem 4.2. *Let $\mathcal{Z}$ be an alphabet. Let n be a security parameter. Let $\mathcal{W}$ be the family of $(\alpha = \Omega(1), \beta \leq \gamma(1 - \Theta(1)))$-partial block sources over $\mathcal{Z}^\gamma$ where $\gamma = \Omega(n)$. Let η be such that $\eta = \omega(\log n)$ and $\eta = o(\gamma)$, and let $c > 1$ be a constant and ℓ be such that $\ell = n^c$. Let $\mathcal{O}$ be an ℓ-composable VGB obfuscator for digital lockers (with κ bit outputs) with auxiliary inputs. Then for every $s_{sec} = \mathtt{poly}(n)$ there exists some $\epsilon_{sec} = \mathtt{ngl}(n)$ such that Construction 4.1 is a $(\mathcal{Z}^\gamma, \mathcal{W}, \kappa, t)$-computational fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$-hard with error δ for*

$$t \leq -\frac{(c-1)}{2}\frac{(\gamma - \eta)\log n}{\eta} = o(\gamma)$$
$$\delta = e^{-n}$$

4.1 Security of Construction 4.1

In this section we outline security of Construction 4.1. A proof of security appears in Appendix D. With overwhelming probability, at each of the ℓ iterations, the sampler will choose enough coordinates of W that have high entropy, making V_i have sufficient entropy. Once each of the $V_1, ..., V_\ell$ have high entropy the obfuscations are unlikely to return a value other than $\perp$ to an adversary. We begin by showing that each V_i is statistically close to a high entropy distribution. Let Λ represent the random variable of all the coins used by Sample and $\lambda = \lambda_1 \dots \lambda_\ell$ be some particular outcome.

Lemma 4.3. *Let all variables be as in Theorem 4.2. There exists $\epsilon_{sam} = O(e^{-\eta}) = \mathtt{ngl}(n)$ and $\alpha' = \alpha\eta(\gamma - \beta - \eta)/\gamma = \omega(\log n)$ such that for each i,*

$$\Pr_{\lambda \leftarrow \Lambda}[\mathrm{H}_\infty(V_i | \Lambda = \lambda) \geq \alpha'] \geq 1 - \epsilon_{sam}.$$

We can then argue that all V_i simultaneously have individual entropy with good probability (by union bound):

Corollary 4.4. *Let ϵ_{sam}, α' be as in Lemma 4.3, and all the other variables be as in Theorem 4.2. Then* $\Pr_{\lambda \leftarrow \Lambda}[\forall i, \mathrm{H}_\infty(V_i|\Lambda = \lambda) \geq \alpha'] \geq 1 - \ell\epsilon_{sam}$.

Once all V_i all simultaneously have good entropy, the adversary only sees $\perp$ as an output from the obfuscations (with overwhelming probability). If the adversary only sees $\perp$ from the obfuscations, they have no information about the key.

Lemma 4.5 (Proof in Appendix D)**.** *Let all the variables be as in Theorem 4.2. For every $s_{sec} = \mathtt{poly}(n)$ there exists $\epsilon_{sec} = \mathtt{ngl}(n)$ such that $\delta^{\mathcal{D}_{s_{sec}}}((R, P), (U_\kappa, P)) < \epsilon_{sec}$.*

4.2 Correctness of Construction 4.1

We encode the entire key in each obfuscation. For correctness, at least one of the repeated readings must be correct with overwhelming probability. Let V_i represent one of the initial readings and V_i' represent a repeated reading. For showing correctness we must show that $\Pr[\forall i, V_i \neq V_i'] < \mathtt{ngl}(n)$.

Lemma 4.6 (Proof in Appendix D)**.** *Let all the variables be as in Theorem 4.2. Then* $\Pr[\forall i, v_i \neq v_i'] < \mathtt{ngl}(n)$*, where the probability is over the coins of* Gen.

4.3 Reusability of Construction 4.1

The reusability of Construction 4.1 follows from the security of the VGB obfuscator with auxiliary input. We consider a bounded $q = \mathtt{poly}(n)$ number of reuses. For some fixed $i \in \{1, ..., q\}$ we will treat the remaining keys as auxiliary input to the adversary, and the simulator still performs comparably to a distinguisher with access to the obfuscations. Thus, given sufficiently strong reusability we achieve the following result:

Theorem 4.7. *Let $q = \mathtt{poly}(n)$, and let all the variables be as in Theorem 4.2, except that $\mathcal{O}$ be an $\ell \times q$-composable VGB obfuscator for digital lockers (with κ bit outputs) with auxiliary inputs. For any admissible $f_2, ..., f_q$, for all $s_{sec} = \mathtt{poly}(n)$ there exists some $\epsilon_{sec} = \mathtt{ngl}(n)$ such that* $(\mathsf{Gen}, \mathsf{Rep})$ *is $(q, \epsilon_{sec}, s_{sec}, f_2, ..., f_q)$-reusable fuzzy extractor.*

Proof. The only modification to the outline presented in Section 4.1 is in Lemma 4.5 with the other keys $R_1, ..., R_{i-1}, R_{i+1}, ..., R_q$ treated as additional auxiliary input to the adversary/simulator. The simulator in the definition of composable obfuscation is required to function for arbitrary circuits in the family even if the choice of these circuits depends on the previous obfuscations. Thus allows reading w_i to be chosen depending on public values $p_1, ..., p_{i-1}$. □

4.4 More errors than entropy?

In this section, we show when Construction 4.1 supports partial block sources with more errors than entropy. The structure of the partial block source implies that $\mathrm{H}_\infty(W) \geq \alpha(\gamma - \beta) = \Theta(\gamma)$. We assume that $\mathrm{H}_\infty(W) = \Theta(\gamma)$. We are able to correct $o(\gamma)$ errors. This yields:

$$\#\text{ Errors} - \text{Entropy} = \log|B_t| - \mathrm{H}_\infty(W) \geq t \log|\mathcal{Z}| - \Theta(\gamma) = o(\gamma)\log|\mathcal{Z}| - \Theta(\gamma)$$

That is, there exists a super-constant alphabet size for which Construction 4.1 is secure with more errors than entropy.

Notes: Construction 4.1 works for an arbitrary size alphabet; however, for a constant size alphabet, the required entropy is greater than the number of corrected error patterns. However, Construction 4.1 is reusability for an arbitrary size alphabet.

In the analysis of Construction 4.1 we restricted our attention to partial block sources, to allow for an easy comparison with Construction 3.2. However, in fact Construction 4.1 is secure for any source where sampling produces a high entropy string (entropy $\omega(\log n)$) with overwhelming probability. For example, it is secure for sources with symbols that are $\omega(\log n)/\log|\mathcal{Z}|$-wise independent.

5 Allowing Correlated Symbols

In the previous two sections, we presented constructions that supported restricted classes of distributions with more errors than entropy. Unfortunately, both Constructions 3.2 and 4.1 required each symbol to contribute "fresh" entropy. In this section, we present a computational construction that allows for correlation between symbols while still supporting more errors than entropy and correcting a constant fraction of errors. This construction is inspired by the construction of digital lockers from point obfuscation by 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 $\{0,1\}^\gamma$ 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]).[2]

Definition 5.1. *For a point $c \in \{0,1\}^\gamma$ define $\mathsf{Neigh}_t(c)$ as the set of all points where at most t bits c_i are changed from 0 to 1.*

Definition 5.2. *Let $\mathsf{Neigh}_t(c)$ be as in Definition 5.1. Then a set C (over $\{0,1\}^\gamma$) is a $(\mathsf{Neigh}_t, \delta_{code})$-code if there exists an efficient procedure Decode such that $\Pr_{c\in C}[\exists c' \in \mathsf{Neigh}_t(c)$ s.t. $\mathsf{Decode}(c') \neq c] \leq \delta_{code}$.*

Construction 5.3. *Let $\mathcal{Z}$ be an alphabet and let $W = W_1, ..., W_\gamma$ be a distribution over $\mathcal{Z}^\gamma$. Let $\mathcal{O}$ be an obfuscator for point functions with points from $\mathcal{Z}$. Let $C \subset \{0,1\}^\gamma$ be an error-correcting code. We describe $\mathsf{Gen}, \mathsf{Rep}$ as follows:*

[2] 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 $2^{\Theta(\gamma)}$ codewords and $t = \Theta(\gamma)$ 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:* $w = w_1, ..., w_\gamma$
2. *Sample* $c \leftarrow C$.
3. *For* $j = 1, ..., \gamma$:
   (i) *If* $c_j = 0$: $p_j = \mathcal{O}(I_{w_j})$.
   (ii) *Else:* $r_j \xleftarrow{\$} \mathcal{Z}$.
   Let $p_j = \mathcal{O}(I_{r_j})$.
4. *Output* (c, p), *where* $p = p_1 \dots p_\gamma$.

Rep

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

Construction 5.3 is secure if no distinguisher can tell whether it is working with random obfuscations or obfuscations of 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 blocks are unpredictable even after adaptive queries to equality oracles for individual symbols. This restriction on the distribution is captured in the following definition.

Definition 5.4. *Let* $I_w(\cdot, \cdot)$ *be an oracle that returns*

$$I_w(j, w'_j) = \begin{cases} 1 & w_j = w'_j \\ 0 & otherwise. \end{cases}$$

A source $W = W_1|...|W_\gamma$ *is a* (q, α, β)-unguessable block source *if there exists a set* $J \subset \{1, ..., \gamma\}$ *of size at least* $\gamma - \beta$ *such that for any unbounded adversary* S *with oracle access to* I_w *making at most* q *queries*

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

We show some examples of unguessable block sources in Appendix B. In particular, any source W where for all j, $\mathrm{H}_\infty(W_j) \geq \omega(\log n)$ (but all blocks may arbitrarily correlated) is an unguessable block source (Claim B.3).

Unfortunately, Construction 5.3 is not 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. We use the notion of HILL entropy [HILL99] extended to the conditional case:

Definition 5.5. *[HLR07, Definition 3] Let* (W, S) *be a pair of random variables.* W *has* HILL entropy *at least* k *conditioned on* S, *denoted* $H^{\mathtt{HILL}}_{\epsilon_{sec}, s_{sec}}(W|S) \geq k$ *if there exists a joint distribution* (X, S), *such that* $\tilde{\mathrm{H}}_\infty(X|S) \geq k$ *and* $\delta^{\mathcal{D}_{s_{sec}}}((W, S), (X, S)) \leq \epsilon_{sec}$.

We now define a weaker object that outputs 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]).

Definition 5.6. *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 2.1, 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 using standard techniques (see Appendix A.1). The following theorem states the construction is a computational fuzzy conductor (proof in Appendix E).

Theorem 5.7. *Let n be a security parameter. Let $\mathcal{Z}$ be an alphabet where $|\mathcal{Z}| \geq 2^{\omega(\log(n))}$. Let $\mathcal{W}$ be a family of $(q, \alpha = \omega(\log n), \beta)$-unguessable block sources over $\mathcal{Z}^\gamma$, for any $q = \mathtt{poly}(n)$. Furthermore, let C be a $(\mathsf{Neigh}_t, \delta_{code})$-code over $\mathcal{Z}^\gamma$. Let $\mathcal{O}$ be an γ-composable VGB obfuscator for point functions with auxiliary inputs. Then for any $s_{sec} = \mathtt{poly}(n)$ there exists some $\epsilon_{sec} = \mathtt{ngl}(n)$ such that Construction 5.3 is a $(\mathcal{Z}^\gamma, \mathcal{W}, \tilde{m} = H_0(C) - \beta, t)$-computational fuzzy conductor that is $(\epsilon_{sec}, s_{sec})$-hard with error $\delta_{code} + \gamma/|\mathcal{Z}|$.*

5.1 More errors than entropy?

In this section, we show that Construction 5.3 can support distributions with more errors than entropy. We first calculate the size of the Hamming ball.

$$\log |B_t| = \log \sum_{i=0}^{t} \binom{\gamma}{i} (|\mathcal{Z}| - 1)^i > \log \binom{\gamma}{t} (|\mathcal{Z}| - 1)^t = \Theta(t \log |\mathcal{Z}|) + \log \binom{\gamma}{t}$$

The simplest type of unguessable block source is where each block is independent and has super-logarithmic entropy (Claim B.1). For this type of source the entropy is $\mathrm{H}_\infty(W) = \gamma\omega(\log n)$. This yields:

$$\#\text{ errors} - \text{entropy} = \log |B_t| - \mathrm{H}_\infty(W) > \left(\Theta(t \log |\mathcal{Z}|) + \log \binom{\gamma}{t}\right) - \gamma\omega(\log n).$$

When $t = \Theta(\gamma)$ and the entropy of each block is $o(\log |\mathcal{Z}|)$, then the construction supports more errors than entropy. Furthermore, the output entropy is $H_0(C) - \beta$ (if C is a constant rate code, this is $\Theta(\gamma)$).

Improvements 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 $c_j = 1$.

If γ is not long enough to get a sufficiently long output, the construction can be run multiple times with the same input and independent randomness.

Acknowledgements

The authors are grateful to Nishanth Chandran, Sharon Goldberg, Gene Itkis, Bhavana Kanukurthi, and Mayank Varia for helpful discussions, creative ideas, and important references.

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.

References

[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.

[BGI+01] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. In *Advances in Cryptology-CRYPTO 2001*, pages 1–18. Springer, 2001.

[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.

[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.

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

[Chv79] Vašek Chvátal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 25(3):285–287, 1979.

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

[DKRS06] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 232–250. Springer Berlin Heidelberg, 2006.

[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.

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

[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.

[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.

[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/`.

[Sca09] Matthew Scala. Hypergeometric tail inequalities: ending the insanity, 2009.

[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.

[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.

[TSv+06] Pim Tuyls, Geert-Jan Schrijen, Boris Škoriá, 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.

[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 Definitions

A.1 Computational Fuzzy Conductors and Computational Extractors

In this section, we show that a computational fuzzy conductor Definition 5.6 can be transformed to a computational fuzzy extractor Definition 2.1. 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 A.1. *A function* $\mathtt{cext} : \{0,1\}^\gamma \times \{0,1\}^d \to \{0,1\}^\kappa$ *a* $(m, \epsilon_{sec}, s_{sec})$-average-case computational extractor *if for all pairs of random variables* X, Y *(with* X *over* $\{0,1\}^\gamma$*) 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 A.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* $\{0,1\}^\gamma$*. Let* $\mathtt{cext} : \{0,1\}^\gamma \times \{0,1\}^d \to \{0,1\}^\kappa$ *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 min-entropy. This is a contradiction. □

A.2 Reusability Fuzzy Extractors

The goal of a reusable fuzzy extractor is to allow enrollment of a source across multiple services. A service i sees an reading of the source w_i. Boyen considers two versions of reusable fuzzy extractors, first where the adversary sees $p_1, ..., p_q$ (outsider security [Boy04, Definition 6]) and tries to learn about the values $w_1, ..., w_q$ or the keys $r_1, ..., r_q$. Second, where the adversary controls some subset of the servers and can key generation on arbitrary p'_i (insider security [Boy04, Definition 7]). This allows the adversary

to learn a subset of keys r_i (by performing key generation on the valid p_i). This definition makes sense when servers are compromised (after enrollment) and act maliciously. In both definitions, the adversary creates a perturbation function $_i$ after seeing $p_1, ..., p_{i-1}$ (and generated keys for outsider security) and the challenger generates $w_i = f_i(w_1)$. The definition is parameterized by the class of allowed perturbation functions.

Boyen constructs a outsider reusable cryptographic fuzzy extractor for unbounded q when the perturbation family is a transitive isometric permutation groups. Boyen transforms this construction to insider security using random oracles.

Insider security strengthens outsider security in two ways. First, it allows the adversary to see some subset of keys, second it allows the adversary to perform key generation on arbitrary p_i. This mixes two properties of a fuzzy extractor: reusability and robustness [DKRS06]. Robust fuzzy extractors provide security against modified p. In this work, we show reusability when r_i are observed but do not handle the issue of robustness. That is, we assume keys may be exposed but servers keep honest state. Our definition lies between outsider and insider security.

We adapt the definition of Boyen to the computational setting (Definition 2.2). The definition of Boyen considers a single adversary. We split the adversary into two parts, one of which is information-theoretic and another that is computationally bounded. The functions $f_2, ..., f_q$ can be thought of as a single adversary that sees all prior state. However, to provide meaningful security in the computational setting, we cannot have communication between these adversaries.[3] Because these two adversaries do not communicate we strengthen the definition by allowing the perturbation functions, f_i, to see the original sample w_1. This was not allowed in the definition of Boyen as it would make security impossible.

A.3 Obfuscation

In this section, we give a formal definition of the required notion of obfuscation. We require that the obfuscation is composable and secure with respect to auxiliary input. Composable auxiliary-input VGB obfuscators for point functions and digital lockers are constructed in [BC10, Theorem 6.1] from the Strong Vector Decision Diffie-Hellman assumption, which is a generalization of the strong DDH assumption of [Can97] for tuples of points. They can also be constructed by assuming strong properties of cryptographic hash functions [Can97].

Definition A.3 (composable obfuscation VGB obfuscation with auxiliary input [BC10])**.** *A PPT algorithm $\mathcal{O}$ is an ℓ-composable VGB obfuscator for $\mathtt{I}_n$ (resp. $\mathtt{I}_{n+\kappa}$) with auxiliary-input if the following conditions are met:*

1. Functionality: *for every n and $I \in \mathtt{I}_n$, $\mathcal{O}(I)$ is a circuit that computes the same function as I.*
2. Virtual grey-box: *For every PPT adversary A and polynomial p, there exists a (possibly inefficient) simulator S and a polynomial q such that for all sufficiently large n, any sequence of circuits $I^1, \ldots, I^\ell \in \mathtt{I}_n$, (where $\ell = \mathtt{poly}(n)$) and for all auxiliary inputs $z \in \{0,1\}^*$:*

$$|\Pr_{A,\mathcal{O}}[A(z, \mathcal{O}(I^1), \ldots, \mathcal{O}(I^\ell)) = 1] - \Pr_{S}[S^{(I^1,\ldots,I^\ell)[q(n)]}(z, 1^{|I^1|}, \ldots, 1^{|I^\ell|}) = 1]| < \frac{1}{p(n)} \ ,$$

where $(I^1, \ldots, I^\ell)[q(n)]$ is an oracle that answers at most $q(n)$ queries, and where every query of the form (i, x) is answered by $I^i(x)$.

[3] An alternative would be to have a single computationally bounded adversary. Construction 4.1 satisfies this alternative adaption as well.

For notational convenience, when we use point function obfuscation, we denote the oracle provided to the simulator as $I_w(\cdot,\cdot)$ where $w = w_1, ..., w_\gamma$ is the vector of obfuscated points. When we use digital lockers we denote the oracle provided to the simulator as $I_{v,r}(\cdot,\cdot)$ where $v = v_1, ..., v_\ell$ is the vector of obfuscated points and r is the hidden value (we will hide the same value in each obfuscation).

B Characterizing unguessable block sources

Definition 5.4 is an inherently adaptive definition and a little unwieldy. In this section, we partially characterize sources that satisfy Definition 5.4. The majority of the difficulty in characterizing Definition 5.4 is that different blocks may be dependent, so an equality query on block i may reshape the distribution of block j. In the examples that follow we denote the adversary by S as we consider security against computationally unbounded adversaries defined in VGB obfuscation (Definition A.3). We first show some sources that are unguessable block sources (Section B.1) and then show distributions with high overall entropy that are not unguessable block sources (Section B.2).

B.1 Positive Examples

We begin with the case of independent blocks.

Claim B.1. *Let $W = W_1, ..., W_\gamma$ be a source in which all blocks W_j are mutually independent. Let α be a parameter. Let $J \subset \{1, ..., \gamma\}$ be a set of indices such that for all $j \in J$, $\mathrm{H}_\infty(W_j) = \alpha$. Then for any q, W is a $(q, \alpha - \log(q+1), \gamma - |J|)$-unguessable block source. In particular, when $\alpha = \omega(\log n)$ and $q = \mathtt{poly}(n)$, then W is a $(q, \omega(\log n), \gamma - |J|)$-unguessable block source.*

Proof. It suffices to show that for all $j \in J, \tilde{\mathrm{H}}_\infty(W_j | View(S^{I_W(\cdot,\cdot)}) = \alpha - \log(q+1)$. We can ignore queries for all blocks but the jth, as the blocks 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 block. 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,

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

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 block-fixing sources, where each block is either uniform or a fixed symbol (block fixing sources were introduced by Kamp and Zuckerman [KZ07]). Claim B.1 shows that Definition 5.4 captures, in particular, this class of distributions. However, Definition 5.4 captures more distributions. We now consider more complicated distributions where blocks are not independent.

Claim B.2. *Let $f : \{0,1\}^e \to \mathcal{Z}^\gamma$ 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 $(q, e - \log(q+1), 0)$-unguessable block source.*

Proof. Since f is injective on each block, $\tilde{\mathrm{H}}_\infty(W_j|View(S^{I_W(\cdot,\cdot)})) = \tilde{\mathrm{H}}_\infty(U_e|View(S^{I_W(\cdot,\cdot)}))$. Consider a query q_k on block 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 B.1), and by [DORS08, Lemma 2.2] we have for all j, $\tilde{\mathrm{H}}_\infty(W_j|View(S^{I_W(\cdot,\cdot)})) \geq \mathrm{H}_\infty(W_j) - \log(q+1)$. □

Note the total entropy of a source in Claim B.2 is e, so there is a family of distributions with total entropy $\omega(\log n)$ for which Construction 5.3 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:

Claim B.3. *Let $W = W_1, ..., W_\gamma$ be a source. 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 $(q, \alpha - 1 - \log(q+1), 0)$-unguessable block source.*

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 Claim B.1. Our inductive hypothesis is that $\Pr[A_1 \neq 0 \vee \cdots \vee A_{k-1} \neq 0] \leq (k-1)2^{1-\alpha}$. If the inductive hypothesis holds, then, for each j,

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

This is true for $k = 1$ by the condition of the theorem. It is true for $k > 1$ 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 $(k-1)2^{1-\alpha} \leq 2q2^{-\alpha} \leq 1/2$.

We now show that $\Pr[A_1 \neq 0 \vee \cdots \vee A_k \neq 0] \leq k2^{1-\alpha}$, assuming that $\Pr[A_1 \neq 0 \vee \cdots \vee A_{k-1} \neq 0] \leq (k-1)2^{1-\alpha}$.

$$\begin{aligned}
\Pr[A_1 \neq 0 \vee \cdots \vee A_{k-1} \neq 0 \vee A_k \neq 0] &= \Pr[A_1 \neq 0 \vee \cdots \vee A_{k-1} \neq 0] + \Pr[A_1 = \cdots = A_{k-1} = 0 \wedge A_k = 1] \\
&\leq (k-1)2^{1-\alpha} + \Pr[A_k = 1 \,|\, A_1 = \cdots = A_{k-1} = 0] \\
&\leq (k-1)2^{1-\alpha} + \max_j 2^{-\mathrm{H}_\infty(W_j|A_1=\cdots=A_{k-1}=0)} \\
&\leq (k-1)2^{1-\alpha} + 2^{1-\alpha} \\
&= k2^{1-\alpha}
\end{aligned}$$

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

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

□

B.2 Negative Examples

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

Claim B.4. *Let $p = (\mathtt{poly}(n))$ and let $f_1, ..., f_\gamma$ be injective functions where $f_j : \{0,1\}^{j \times \log p} \to \{0,1\}^n$.*[4] *Then define the distribution $W_1 = f_1(U_{1,...,\gamma}), W_2 = f_2(U_{1,...,2\gamma}), ..., W_\gamma = f_\gamma(U)$. There is an adversary making $p \times \gamma = \mathtt{poly}(n)$ queries such that $\mathrm{H}_\infty(W|View(S^{I_W(\cdot,\cdot)})) = 0$.*

Proof. Let x be the true value for $U_{p\times\gamma}$. 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 $(1, y_1), ..., (1, y_p)$, 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 adversaries ability to determine a block from the previous blocks. We formalize this notion next. We define the entropy jump of a block source as the remaining entropy when other blocks are known:

Definition B.5. *Let $W = W_1, ..., W_\gamma$ be a source under ordering $i_1, ..., i_\gamma$. The* jump *of a block i_j is $\mathtt{Jump}(i_j) = \max_{w_{i_1},...,w_{i_{j-1}}} H_0(W_{i_j}|W_{i_1} = w_{i_1}, ..., W_{i_{j-1}} = w_{i_{j-1}})$.*

If an adversary can learn blocks in succession they can eventually recover the entire secret. In order for a source to be block unguessable the adversary must get "stuck" early enough in their recovery process. This translates to having a super-logarithmic jump early enough.

Claim B.6. *Let W be a distribution and let q be a parameter, if there exists an ordering $i_1, ..., i_\gamma$ such that for all $j \leq \gamma - \beta + 1$, $\mathtt{Jump}(i_j) = \log q/(\gamma - \beta + 1)$, then W is not $(q, 0, \beta)$-unguessable block source.*

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

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

C Analysis of Construction 3.2

Proof of Lemma 3.4. Let $W \in \mathcal{W}$. It suffices to argue correctness and security. We first argue 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 $(\mathsf{Gen}', \mathsf{Rep}')$, $\Pr[(r, p) \leftarrow \mathsf{Gen}'(v) \wedge r' \leftarrow \mathsf{Rep}(v', p) \wedge r' = r] \geq 1 - \delta$.

We now argue security. Denote by *seed* the random variable consisting of all γ seeds and V the entire string of generated $V_1, ..., V_\gamma$. To show that $R|P, seed \approx_{\gamma\epsilon_{cond}+\epsilon_{fext}} U|P, seed$, it suffices to show that

[4] Here we assume that $n \geq \gamma \times \log p$, that is the source has a small number of blocks.

$\tilde{\mathrm{H}}_\infty(V|seed)$ is $\gamma\epsilon_{cond}$ close to a distribution with average min-entropy $\tilde{\alpha}(\gamma-\beta)$. The lemma then follows by the security of $(\mathsf{Gen}', \mathsf{Rep}')$.[5]

We now argue that there exists a distribution Y where $\tilde{\mathrm{H}}_\infty(Y|seed) \geq \tilde{\alpha}(\gamma-\beta)$ and $(V, seed_1, ..., seed_\gamma) \approx (Y, seed_1, .., seed_\gamma)$. First note since W is (α, β)-partial block distribution that there exists a set of indices J where $|J| \geq \gamma - \beta$ 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,

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

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

$$(\mathtt{cond}(W_\gamma, seed_\gamma), ..., \mathtt{cond}(W_1, seed_1), seed_\gamma, ..., seed_1) \approx_{\gamma\epsilon_{cond}} (Z, seed_\gamma, ..., seed_1).$$

This completes the proof. □

D Analysis of Construction 4.1

D.1 Security

The proof of security for Construction 4.1 uses the definition of block unguessable sources (Definition 5.4). This definition is adaptive and discussed in Appendix B. We show the security of Construction 4.1:

- Lemma 4.3: Show that sampling is successful with overwhelming probability.
- Corollary 4.4: The outputs $V_1, .., V_\ell$ have high individual entropy with good probability.
- The outputs $V_1, ..., V_\ell$ are a block unguessable source. This is made formal in the following corollary:

[5] 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.

Corollary D.1. *Let ϵ_{sam}, α' be as in Lemma 4.3, and all the other variables be as in Theorem 4.2. Take any $q = \mathtt{poly}(n)$. For $\alpha'' = \alpha' - 1 - \log(q+1) = \omega(\log n)$, with probability $1 - \ell\epsilon_{sam}$ over the choice of $\Lambda = \lambda$, the distribution $V|\Lambda = \lambda$ is a $(q, \alpha'', 0)$-unguessable block source.*

- Lemma 4.5: An adversary is unlikely to receive any information about the key for a block unguessable source.

We now present the proofs of Lemmas 4.3 and 4.5.

Proof of Lemma 4.3. Consider some fixed i. Recall that there a set J of size $\gamma - \beta = \Theta(\gamma)$ such that each w and block $j \in J$, $\mathrm{H}_\infty(W_j|W_1 = w_1, ..., W_{j-1} = w_{j-1}, W_{j+1} = w_{j+1}, ..., W_\gamma = w_\gamma) \geq \alpha$. Since this is a worst case guarantee, the entropy of V_i can be deduced from the number of symbols in V_i that come from J. Namely, Denote by $X = |\{j_{i,1}, ..., j_{i,\eta}\} \cap J|$.

Claim D.2.

$$\mathrm{H}_\infty(V_i|\Lambda = \lambda) \geq \alpha X.$$

Proof. Denote by $j_1, ..., j_\eta$ the indices selected by the randomness λ_i. We begin by noting that $\mathrm{H}_\infty(V_i|\Lambda = \lambda) = -\log \max_{v \in V_i} \Pr[V_i = v|\Lambda = \lambda] = -\log \max_{w_{j_1}, ..., w_{j_\eta}} \Pr[W_{j_1} = w_{j_1} \wedge \cdots \wedge W_{j_\eta} w_{j_\eta}]$. Then

$$\begin{aligned}
\max_{w_{j_1}, ..., w_{j_\eta}} \Pr[W_{j_1} = w_{j_1} \wedge \cdots \wedge W_{j_\eta} = w_{j_\eta}] &= \max_{w_{j_1}, ..., w_{j_\eta}} \prod_{k=1}^{\eta} \Pr[W_{j_k} = w_{j_k} | W_{j_{k-1}} = w_{j_{k-1}} \wedge ... \wedge W_{j_1} = w_{j_1}] \\
&\leq \prod_{k=1}^{\eta} \max_{w_{j_1}, ..., w_{j_\eta}} \Pr[W_{j_k} = w_{j_k} | W_{j_{k-1}} = w_{j_{k-1}} \wedge ... \wedge W_{j_1} = w_{j_1}] \\
&\leq \prod_{k=1}^{\eta} \max_{w_1, ..., w_\gamma} \Pr[W_{j_k} = w_{j_k} | W_1 = w_1 \wedge ... \wedge W_{j_k - 1} = w_{j_k - 1}]
\end{aligned}$$

Taking the negative logarithm of both sides we have that

$$\begin{aligned}
\mathrm{H}_\infty(V_i|\Lambda = \lambda) &\geq \sum_{k=1}^{\eta} \min_{w_1, ..., w_\gamma} \mathrm{H}_\infty(W_{j_k} | W_1 = w_1 \wedge ... \wedge W_{j_k - 1} = w_{j_k - 1}) \\
&\geq \sum_{j_k \in J} \alpha = \alpha X
\end{aligned}$$

This completes the proof of Claim D.2. □

We note that X is distributed according to the hypergeometric distribution, and that $\mathbb{E}[X] = \eta(\gamma - \beta)/\gamma$. Using the tail bounds from [Chv79, Sca09], we can conclude that $\Pr[X \leq \mathbb{E}[X]/2] \leq e^{-2((\gamma-\beta)/2\gamma)^2 \eta} = O(e^{-\eta})$.

Thus, setting $\alpha' = \frac{\alpha\eta(\gamma-\beta)}{2\gamma}$ and applying Claim D.2, we conclude that

$$\Pr[\mathrm{H}_\infty(V_i) \geq \alpha'] \geq 1 - O(e^{-\eta}).$$

□

Proof of Lemma 4.5. Let $\mathcal{O}$ be a ℓ-composable VGB obfuscator with auxiliary input for digital lockers over $\mathcal{Z}$[6] . Let W be a $(q, \alpha'' = \omega(\log n), 0)$-unguessable block source. Our goal is to show that for all $s_{sec} = \mathtt{poly}(n)$ there exists $\epsilon_{sec} = \mathtt{ngl}(n)$ such that $\delta^{\mathcal{D}_{s_{sec}}}((R,P),(U,P)) \leq \epsilon_{sec}$.

Suppose not, that is suppose there is some $s_{sec} = \mathtt{poly}(n)$ such that exists $\epsilon_{sec} = \mathtt{poly}(n)$ and $\delta^{\mathcal{D}_{s_{sec}}}((R,P),(U,P)) > \epsilon_{sec}$. Let D be such a distinguisher of size at most s_{sec}. That is,

$$|\,\mathbb{E}[D(R,P)] - \mathbb{E}[D(U,P)] > \epsilon_{sec} = 1/\mathtt{poly}(n).$$

Define the oracle $I_{v_1,...,v_\ell,r}(\cdot,\cdot)$ as follows:

$$I_{v_1,...,v_\ell,r}(x,i) = \begin{cases} r & v_i = x \\ \perp & \text{otherwise.} \end{cases}$$

By the security of obfuscation (Definition A.3), there exists a unbounded time simulator S (making at most q queries) such that

$$|\,\mathbb{E}[D(R,P_1,...,P_\ell)] - \mathbb{E}[S^{I_{v_1,...,v_\ell,r}(\cdot,\cdot)}(R,1^{\ell\log|Z|})]| \leq \epsilon_{sec}/3. \tag{2}$$

We now prove S cannot distinguish between R and U.

Lemma D.3. $\Delta(S^{I_{v_1,...,v_\ell,r}(\cdot,\cdot)}(R,1^{\ell\log|Z|}), S^{I_{v_1,...,v_\ell,r}(\cdot,\cdot)}(U,1^{\ell\log|Z|})) \leq \ell 2^{-\alpha''}$.

Proof. It suffices to show that for any two values in $\{0,1\}^\kappa$, the statistical distance is at most $\ell 2^{-\alpha''}$.

Lemma D.4. *Let r be true value encoded in I and let $u \in \{0,1\}^\kappa$. Then,*

$$\Delta(S^{I_{v_1,...,v_\ell,r}(\cdot,\cdot)}(r,1^{\ell\log|Z|}), S^{I_{v_1,...,v_\ell,r}(\cdot,\cdot)}(u,1^{\ell\log|Z|})) \leq \ell 2^{-\alpha''}.$$

Proof. Recall that for all j, $\tilde{\mathrm{H}}_\infty(V_j|View(S)) \geq \alpha''$. The only information about the correct value of r is contained in the query responses. When all responses are $\perp$ the view of S is identical when presented with r or u. We now show that for any value of r all queries return $\perp$ with probability $1-2^{-\alpha''}$. Suppose not, that is suppose, the probability of at least one nonzero response is $> 2^{-(\alpha'')}$.

When there is a response other than $\perp$ for some j this means that there is no remaining min-entropy in V_j. If this occurs with over $2^{-\alpha''}$ probability this violates the block unguessability of V (Definition 5.4). By the union bound over the indices j the total probability of a response other than $\perp$ is at most $\ell 2^{-\alpha''}$. Thus, for all r, u the statistical distance is at most $\ell 2^{-\alpha''}$. This concludes the proof of Lemma D.4. □

By averaging over all points in $\{0,1\}^\kappa$ we conclude that

$$\Delta(S^{I_{v_1,...,v_\ell,r}X(\cdot,\cdot)}(R,1^{\ell\log|Z|}), S^{I_{v_1,...,v_\ell,r}(\cdot,\cdot)}(U,1^{\ell\log|Z|})) < \ell 2^{-\alpha''}.$$

This completes the proof of Lemma D.3. □

Now by the security of obfuscation we have that

$$|\,\mathbb{E}[D(R,P_1,...,P_\ell)] - \mathbb{E}[S^{I_{v_1,...,v_\ell,r}(\cdot,\cdot)}(R,1^{\ell\log|Z|})]| \leq \epsilon_{sec}/3. \tag{3}$$

[6] In this proof we only consider the case where the sampling has produced a block unguessable source. The negligible portion of the time when this does not happen in included in the security of Theorem 4.2

Combining Equations 4 and 6 and Lemma D.3, we have

$$\begin{aligned}\delta^D((R,P),(U,P)) &\leq |\,\mathbb{E}[D(R,P_1,...,P_\ell)] - \mathbb{E}[S^{I_{v_1},...,v_\ell,r(\cdot,\cdot)}(R,1^{\ell\log|Z|})]| \\ &+ |\,\mathbb{E}[S^{I_{v_1},...,v_\ell,r(\cdot,\cdot)}(R,1^{\ell\log|Z|})] - \mathbb{E}[S^{I_{v_1},...,v_\ell,r(\cdot,\cdot)}(U,1^{\ell\log|Z|})]| \\ &+ |\,\mathbb{E}[S^{I_{v_1},...,v_\ell,r(\cdot,\cdot)}(U,1^{\ell\log|Z|})] - \mathbb{E}[D(U,P_1,...,P_\ell)]| \\ &\leq \epsilon_{sec}/3 + \ell 2^{-\alpha''} + \epsilon_{sec}/3 \\ &\leq 2\epsilon_{sec}/3 + \mathtt{ngl}(n) < \epsilon_{sec}.\end{aligned}$$

This is a contradiction and completes the proof of Lemma 4.5. □

D.2 Correctness

Proof of Lemma 4.6. Recall that $\mathsf{dis}(w,w') \leq t$ and that the locations of the errors is independent of the selected locations. Denote by $\mu = -\frac{(c-1)\log n}{2}$. Since $\eta = \omega(\log n)$, we will assume $\eta \geq 2\mu$. We begin by computing the probability that a single $v_i = v'_i$.

$$\begin{aligned}\Pr[v_i = v'_i] &= \Pr[w \text{ and } w' \text{ agree on positions } j_{i,1},...,j_{i,\eta}] \\ &\geq \prod_{j=0}^{\eta-1}\left(1-\frac{t}{\gamma-j}\right) \geq \prod_{j=0}^{\eta-1}\left(1-\frac{\mu(\gamma-\eta)/\eta}{\eta-j}\right) \\ &\geq \prod_{j=0}^{\eta-1}\left(1-\frac{\mu}{\eta}\left(\frac{\gamma-\eta}{\gamma-j}\right)\right) \geq \prod_{j=0}^{\eta-1}\left(1-\frac{\mu}{\eta}\right) \\ &= \left(1-\frac{\mu}{\eta}\right)^{\eta} = \left(\left(1-\frac{\mu}{\eta}\right)^{\eta/\mu}\right)^{\mu} \geq \left(\frac{1}{2}\right)^{2\mu} \\ &\geq \left(\frac{1}{2}\right)^{(c-1)\log n} = \frac{1}{n^{c-1}}.\end{aligned}$$

We then have the probability that all $v_i \neq v'_i$ as:

$$\begin{aligned}\Pr[\forall i, v_i \neq v'_i] &= \left(1-\Pr[v_i = v'_i]\right)^{\ell} \\ &= \left(1-\frac{1}{n^{c-1}}\right)^{\ell} = \left(\left(1-\frac{1}{n^{c-1}}\right)^{n^{c-1}}\right)^{\ell/n^{c-1}} \\ &\leq \left(\frac{1}{e}\right)^{n^c/n^{c-1}} = \frac{1}{e^n}.\end{aligned}$$

This completes the proof of Lemma 4.6. □

E Analysis of Construction 5.3

E.1 Security

Security of Construction 5.3 is similar to the security of Construction 4.1. However, security is more complicated, the main difficulty is that the definition of block unguessable sources (Definition 5.4) allows

for certain weak blocks that can easily be guessed. This means we must limit our indistinguishable distribution to blocks that are difficult to guess. Security is proved via the following lemma:

Lemma E.1. *Let all variables be as in Theorem 5.7. 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. It is sufficient to show that there exists a distribution C' with conditional min-entropy and $\delta^{\mathcal{D}_{s_{sec}}}((C,P),(C',P)) \leq \mathtt{ngl}(n)$. Let J be the set of indices that exists according to Definition 5.4. 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 $\delta^{\mathcal{D}_{s_{sec}}}((C,P),(C',P)) \leq \mathtt{ngl}(n)$. The outline for the rest of the proof is as follows:

- Let D be a distinguisher between (C,P) and (C',P). Since P is a collection of obfuscated programs, there exists a simulator S (outputting a single bit), such that $\Pr[D(C,P)=1]$ is close to $\Pr[S^{\mathcal{O}}(C)=1]$.
- Show that even an unbounded S making a polynomial number of queries to the stored points cannot distinguish between C and C'. That is, $\Delta(S^{\mathcal{O}}(C), S^{\mathcal{O}}(C'))$ is small.
- By the security of obfuscation, $\Pr[S^{\mathcal{O}}(C')=1]$ is close to $\Pr[D(C',P)=1]$.

Proof of Lemma E.1. Let $\mathcal{O}$ be a γ-composable VGB obfuscator with auxiliary input for point programs over $\mathcal{Z}$. Let W be a $(q, \alpha = \omega(\log n), \beta)$-unguessable block source. Our goal is to show that for all $s_{sec} = \mathtt{poly}(n)$ there exists $\epsilon_{sec} = \mathtt{ngl}(n)$ such that $H^{\mathtt{HILL}}_{\epsilon_{sec},s_{sec}}(C|P) \geq H_0(C) - \beta$. Suppose not, that is suppose there is some $s_{sec} = \mathtt{poly}(n)$ such that exists $\epsilon_{sec} = \mathtt{poly}(n)$ and $H^{\mathtt{HILL}}_{\epsilon_{sec},s_{sec}}(C|P) < H_0(C) - \beta$. By Definition 5.4 there exists a set of indices J such that all blocks within J are unguessable. Define by C' the distribution of sampling a uniform codeword where all locations outside J are fixed. Then $\tilde{\mathrm{H}}_\infty(C'|C_{J^c}) \geq \mathrm{H}_\infty(C', C_{J^c}) - H_0(C_{J^c}) = H_0(C) - \beta$ (by [DORS08, Lemma 2.2b]).

Let D a distinguisher of size at most s_{sec} such that

$$|\,\mathbb{E}[D(C,P)] - \mathbb{E}[D(C',P)] > \epsilon_{sec} = 1/\mathtt{poly}(n).$$

Define the distribution X as follows:

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

By the security of obfuscation (Definition A.3), there exists a unbounded time simulator S (making at most q queries) such that

$$|\,\mathbb{E}[D(P_1, ..., P_\gamma, C)] - \mathbb{E}[S^{I_X(\cdot,\cdot)}(C, 1^{\gamma \log |Z|})]| \leq \epsilon_{sec}/3. \tag{4}$$

We now prove S cannot distinguish between C and C'.

Lemma E.2. $\Delta(S^{I_X(\cdot,\cdot)}(C, 1^{\gamma \log |Z|}), S^{I_X(\cdot,\cdot)}(C', 1^{\gamma \log |Z|})) \leq (\gamma - \beta)2^{-(\alpha+1)}$.

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

Lemma E.3. *Let c^* be true value encoded in X and let c' a codeword in C'. Then,*

$$\Delta(S^{I_X(\cdot,\cdot)}(c^*, 1^{\gamma \log |Z|}), S^{I_X(\cdot,\cdot)}(c', 1^{\gamma \log |Z|})) \leq (\gamma - \beta)2^{-(\alpha+1)}.$$

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:

$$\begin{aligned}
2^{-\alpha+1} &< \Pr[\text{non zero response location } j] \\
&= \Pr[c_j^* = 1] \Pr[\text{non zero response location } j \wedge c_j^* = 1] \\
&+ \Pr[c_j^* = 0] \Pr[\text{non zero response location } j \wedge c_j^* = 0] \\
&\leq 1 \times 2^{-\log|\mathcal{Z}|+\log q} + 1 \times \Pr[\text{non zero response location } j \wedge c_j^* = 0] \qquad (5)
\end{aligned}$$

We now show that for an unguessable block source the remaining entropy $\alpha \leq \log|\mathcal{Z}| - \log q$:

Claim E.4. *If W is a (q, α, β)-block unguessable source over $\mathcal{Z}$ then $\alpha \leq \log|\mathcal{Z}| - \log q$.*

Proof. Let W be a (q, α, β)-block unguessable source. Let $J \subset \{1, ..., \gamma\}$ the set of good indices. It suffices to show that there exists an S making q queries such that for some $j \in J, \tilde{\mathrm{H}}_\infty(W_j|S^{I_W(\cdot,\cdot)}) \leq \log|\mathcal{Z}| - \log q$. 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 its oracle on $(j, w_{j,1}), .., (j, w_{j,q})$. Denote by Bad the random variable when $W_j \in \{w_{j,1}, .., w_{j,q}\}$ After these queries the remaining min-entropy is at most:

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

This completes the proof of Claim E.4. □

Rearranging terms in Equation 5, 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 min-entropy. If this occurs with over $2^{-\alpha}$ probability this violates the block unguessability of W (Definition 5.4). By the union bound over the indices $j \in J$ the total probability of a 1 in J is at most $(\gamma - \beta)2^{-\alpha+1}$. Recall that c^*, c' match on all indices outside of J. Thus, for all c^*, c' the statistical distance is at most $(\gamma - \beta)2^{-\alpha+1}$. This concludes the proof of Lemma E.3. □

By averaging over all points in C' we conclude that $\Delta(S^{I_X(\cdot,\cdot)}(C, 1^{\gamma \log |Z|}), S^{I_X(\cdot,\cdot)}(C', 1^{\gamma \log |Z|})) < (\gamma - \beta)2^{-(\alpha+1)}$. This completes the proof of Lemma E.2. □

Now by the security of obfuscation we have that

$$|\,\mathbb{E}[D(P_1, ..., P_\gamma, C')] - \mathbb{E}[S^{I_X(\cdot,\cdot)}(C', 1^{\gamma \log |Z|})]| \leq \epsilon_{sec}/3. \tag{6}$$

Combining Equations 4 and 6 and Lemma E.2, we have

$$\begin{aligned}
\delta^D((P, C), (P, C')) &\leq |\,\mathbb{E}[D(P_1, ..., P_\gamma, C)] - \mathbb{E}[S^{I_X(\cdot,\cdot)}(C, 1^{\gamma \log |Z|})]| \\
&+ |\,\mathbb{E}[S^{I_X(\cdot,\cdot)}(C, 1^{\gamma \log |Z|})] - \mathbb{E}[S^{I_X(\cdot,\cdot)}(C', 1^{\gamma \log |Z|})]| \\
&+ |\,\mathbb{E}[S^{I_X(\cdot,\cdot)}(C', 1^{\gamma \log |Z|})] - \mathbb{E}[D(P_1, ..., P_\gamma, C')]| \\
&\leq \epsilon_{sec}/3 + (\gamma - \beta)2^{-(\alpha-1)} + \epsilon_{sec}/3 \\
&\leq 2\epsilon_{sec}/3 + \mathtt{ngl}(n) < \epsilon_{sec}.
\end{aligned}$$

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

E.2 Correctness

We now argue correctness of Construction 5.3. We begin by showing that the probability of a single $1 \to 0$ bit flip in c is negligible.

Lemma E.5. *Let all variables be as in Theorem 5.7. The probability of at least one $1 \to 0$ bit flip (an obfuscation of a random block being interpreted as the obfuscation of the point) is $\leq \gamma/|\mathcal{Z}| = \mathtt{ngl}(n)$.*

Proof. Consider a coordinate j for which $c_j = 1$. Since w' is chosen independently of the points r_j, and r_j is uniform, $\Pr[r_j = w'_j] = 1/|\mathcal{Z}|$. The lemma follows by the union bound, since there are at most γ such coordinates. □

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. Therefore, Construction 5.3 is correct with error at most $\gamma/|\mathcal{Z}|$.