Key Derivation From Noisy Sources With More Errors Than Entropy

Ran Canetti[*] Benjamin Fuller[†] Omer Paneth[‡] Leonid Reyzin[§]

April 6, 2014

Abstract

Fuzzy extractors convert a noisy source of entropy into a consistent uniformly-distributed key. In the process of eliminating noise, they lose some of the entropy of the original source—in the worst case, as much as the logarithm of the number of correctable error patterns. We call what is left after this worst-case loss the *minimum usable entropy*. Unfortunately, this quantity is negative for some sources that are important in practice. Most known approaches for building fuzzy extractors work in the worst case and cannot be used when the minimum usable entropy is negative.

We construct the first fuzzy extractors that work for a large class of distributions that have negative minimum usable entropy. Their security is computational. They correct Hamming errors over a large alphabet. In order to avoid the worst-case loss, they necessarily restrict distributions for which they work.

Our first construction requires high individual entropy of a constant fraction of symbols, but permits symbols to be dependent. Our second construction requires a constant fraction of symbols to have a constant amount of entropy conditioned on prior symbols. The constructions can be implemented efficiently based on number-theoretic assumptions or assumptions on cryptographic hash functions.

Keywords Fuzzy extractors, 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 at repeated readings (prominent examples include biometrics and other human-generated data [Dau04, ZH93, BS00, EHMS00, MG09, MRW02], physically unclonable functions [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, which is useful for a single user trying to produce the same key from multiple readings of a physical source at different times. A *fuzzy extractor* is the primitive that accomplishes both information reconciliation and privacy amplification non-interactively; fuzzy extractors are defined information-theoretically in [DORS08].

[*] 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.

Fuzzy extractors consist of a pair of algorithms: Gen takes a source value w, and produces a key r and a public helper value p. The second algorithm Rep takes this helper value p and a close w' to reproduce the original key r. The security guarantee is that r produced by Gen is close to uniform (information-theoretically [DORS08] or computationally [FMR13]), even given p, as long as w comes from a high-quality distribution (traditionally, any distribution with sufficient min-entropy m). 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 paper, we focus on the Hamming metric on length ℓ strings over some alphabet $\mathcal{Z}$).

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 any w' within distance t of w, 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.

More generally, let B_t denote the number of points in a ball of radius t. For any m and t, there is a distribution of min-entropy m such that the adversary can guess a correct w' with probability $1/\lceil(2^m/B_t)\rceil \approx B_t 2^{-m}$: the distribution consists of the uniform distribution over all points in several non-overlapping balls of radius t (the metric space must be large enough for these balls not to intersect). We thus call $m - \log B_t$ the *minimum usable* entropy, denoted by $\mathrm{H}_{\mathtt{usable}}$. The previous paragraph shows that no fuzzy extractor can handle all distributions of a given min-entropy m if $\mathrm{H}_{\mathtt{usable}} \leq 0$.

Prime candidate sources for authentication have $\mathrm{H}_{\mathtt{usable}} \leq 0$. As an example, the iris is believed to be the best biometric for high security applications [PPJ03]. Daugman estimates irises contain 249 bits of entropy [Dau04]. Daugman uses specialized wavelets to derive a 2048 bit string called an iris code. Let the outcome of this transform (on different irises) define a distribution w. The precise number of errors that must be tolerated depends on the desired false reject rate (how often the correct key is produced). For a false reject rate of $\leq 80\%$, a t of approximately 205 is required. We have the following calculation for $\mathrm{H}_{\mathtt{usable}}$:

$$\mathrm{H}_{\mathtt{usable}} = \mathrm{H}_\infty(W) - \log|B_t| = 249 - \log \sum_{i=0}^{205} \binom{2048}{i} \approx -707.$$

There is considerable subsequent research [CGR09, GRC09, RUW11] to Daugman but it does not affect $\mathrm{H}_{\mathtt{usable}}$ dramatically.

Our Contributions We provide the first constructions of computational fuzzy extractors that can be used for a large class of distributions with $\mathrm{H}_{\mathtt{usable}} \leq 0$ over $\mathcal{Z}^\ell$ for a large alphabet $\mathcal{Z}$. As explained above, such constructions cannot work without some restriction on the distribution. Our first construction is secure when 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 entropy, as long as knowledge of their values does not reduce the entropy of the high-entropy symbols too much (see Definition 4.2).

We improve the entropy requirement in the second construction, which requires only a constant fraction of the symbols w to have constant min-entropy conditioned on the previous symbols. However, this improvement comes at a price to error-tolerance: whereas the first construction tolerates a constant fraction of errors, the second construction tolerates $\ell/\omega(\log \ell)$ errors.

Our Approach Our constructions are computationally secure. Known techniques for proving security of information-theoretic fuzzy extractors work for all distributions for a given m, t, and thus cannot be used with $\mathrm{H}_{\mathtt{usable}} \le 0$ (because they would prove security for distributions contained within a single ball of radius of t).

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. Fuller, Meng, and Reyzin show that replacing secure sketches with a similar computational component is unlikely to be fruitful [FMR13, Corollary 3.8, Theorem 3.10]. Instead, they suggest two alternatives: combine the information-reconciliation and privacy amplification components (their approach) or produce a new consistent secret with computational entropy instead of recovering w. We take the second approach.

Any procedure that converts a high-entropy input to a high-entropy output is known as a *conductor* [CRVW02]; if it's error-tolerant, then it's a *fuzzy conductor* [KR09]. We show two constructions of *computational fuzzy conductors*. These constructions may be converted to computational fuzzy extractors using information-theoretic [NZ93] or computational [Kra10] extractors (Lemma 3.5).

Both constructions are based on obfuscation of point programs [Can97]. A point program $I_w(x)$ outputs 1 if $x = w$ and 0 otherwise. Intuitively, an obfuscated version of a program reveals nothing past its input and output behavior. For a point program, this means hiding all partial information about the point w. We need a strong version of point obfuscation that remains secure even when several obfuscations of correlated points are composed. While the standard definition of obfuscation [BGI⁺01] does not imply security under composition, we can base our construction on the relaxed notion of *virtual grey-box* obfuscation introduced in [BC10]. For this notion, [BC10] construct composable obfuscation of point programs under particular number-theoretic assumptions. Additionally, such obfuscation can be made very efficient under a strong assumption on cryptographic hash functions [Can97].

Both of our constructions are inspired by Canetti and Dakdouk's construction of digital lockers from point obfuscation [CD08]. Let $w = w_1 \dots w_\ell$, for $w_j \in \mathcal{Z}$. In the first construction (Construction 4.1), for each j, Gen flips a coin c_j and either obfuscates I_{w_j} or picks a random point r_j and obfuscates I_{r_j}. This produces ℓ obfuscated programs $P_1, ..., P_\ell$. With a close value w', Rep then runs the obfuscated program P_j on w'_j and checks whether $P_j(w'_j) = 1$. For most locations j, Rep can determine whether w_j or a random value was obfuscated. Thus, most bits of c_j are recoverable. To tolerate errors, the set of coins $c_1 \dots c_\ell$ is chosen at random from the codewords of an error correcting code. This construction conducts entropy from w to c.

Obfuscation of point functions provides no security if a point can be guessed; thus, in order for the first construction to be secure, sufficiently many coordinates of w have to be unguessable (even to an adversary who can make equality queries for the values of other coordinates). We relax this requirement in our second construction (Construction 5.2), called *sample-then-obfuscate*: it transforms w into a string of blocks and then applies the first construction. Gen randomly samples several coordinates of w and concatenates them to form a block. This reduces the entropy requirement on the individual symbols, but lowers the error-tolerance. This approach is similar to the sample-then-extract paradigm for building locally computable extractors [Lu02, Vad03]. Unlike in locally computable extractors, we can form multiple blocks sampling from the same value w and only argue about their individual entropy, because correlations among blocks are allowed in the first construction. Computational, rather than information-theoretic, analysis seems crucial for achieving this property.

Connection to General Obfuscation We note that fuzzy extractors for the setting where $\mathrm{H}_{\mathtt{usable}} \le 0$ can be trivially constructed from a strong form of obfuscation, specifically, virtual black-box obfuscation for the class *proximity point programs*, $I_w(x)$ that tests if x is within Hamming distance t of w. However, currently we do not know if such obfuscation exists, let alone whether it can be made as efficient as our constructions. Recent works constructed candidate indistinguishability obfuscators [GGH$^+$13, PST13] and virtual black-box obfuscators in an idealized model [BR13, BGK$^+$14] for all programs. However, these notions of obfuscation do not imply virtual-black-box obfuscation for proximity point programs in the plain model. The work of [BBC$^+$14] suggests an average-case virtual-black-box obfuscator for the family of *evasive* functions that test if a low-degree multi-variant arithmetic circuit evaluates to zero. However, we do not know if these functions include proximity point programs.

Open Problems Both constructions require a large alphabet $\mathcal{Z}$—one whose size is more than polynomial in the security parameter.[1] It is possible to tweak the sample-then-obfuscate construction for use with a small alphabet. However, distributions with $\mathrm{H}_{\mathtt{usable}} \leq 0$ are supported only for large alphabets. For small alphabets, $\mathrm{H}_{\mathtt{usable}} > 0$ and there are good information-theoretic constructions known [DORS08, Section 5]. Constructing a computational fuzzy extractor when $\mathrm{H}_{\mathtt{usable}} \leq 0$ and a small alphabet is an open problem.

Using information-theoretic fuzzy extractors with additional privacy properties, Dodis and Smith [DS05, Section 5] construct program obfuscators for the program $I_w(x)$ that tests if x is within Hamming distance t of w. The obfuscation is secure as long as w comes from a distribution of sufficient min-entropy; in particular, the entropy must be high enough so that $\mathrm{H}_{\mathtt{usable}} > 0$. Our constructions do not provide obfuscators for proximity queries, because they leak more information than whether x is within distance t of w (for example, they may provide some information about the actual distance or about which coordinates agree). Constructing an efficient obfuscator for proximity queries when $\mathrm{H}_{\mathtt{usable}} \leq 0$ is an open problem.

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. A meaningful restriction of correctable errors is an open problem.

The remainder of this paper is organized as follows: we cover notation and background on obfuscation and error correcting codes in Section 2, describe computational fuzzy extractors in Section 3, and present our two constructions in Sections 4 and 5 respectively.

2 Preliminaries

For a random variables X_i over some alphabet $\mathcal{Z}$ we denote by $X = X_1, ..., X_\ell$ the tuple $(X_1, \ldots, X_\ell)$. 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, $B_t(x) = \{y|\mathsf{dis}(x, y) \leq t\}$. If the size of a ball in a metric space does not depend on x, we denote

[1] Codes over large alphabets are often used to correct burst errors [Gil60].

by $|B_t|$ the size of a ball of radius t. We consider the Hamming metric over vectors in $\mathcal{Z}^\ell$, defined via $\mathsf{dis}(x, y) = \{i | x_i \neq y_i\}$. For this metric, $|B_t| = \sum_{i=0}^{t} \binom{\ell}{i}(|\mathcal{Z}| - 1)^i$. 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 Coding Theory

We will consider slightly nonstandard error-correct codes over $\{0, 1\}^\ell$, 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]).

Definition 2.1. *For a point $c \in \{0, 1\}^\ell$ 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 2.2. *Let $\mathsf{Neigh}_t(c)$ be as in Definition 2.1. Then a set C (over $\{0, 1\}^\ell$) is a $(\mathsf{Neigh}_t, \delta_{code})$-code if there exists an efficient procedure* Decode *such that $\Pr_{c\in C}[\exists c' \in \mathsf{Neigh}_t(c) \text{ s.t. } \mathsf{Decode}(c') \neq c] \leq \delta_{code}$.*

Notes: 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 $2^{\Theta(\ell)}$ codewords and $t = \Theta(\ell)$ 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. See Constructions 4.1 and 5.2 for the translation between the error classes.

2.2 Obfuscation

Our construction uses obfuscation for a family of point functions $\mathtt{I}_n = \{I_w\}_{w\in\{0,1\}^n}$ defined as follows:

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

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 most a polynomial number of oracle queries to the function. In the following definition we also require that the obfuscation is composable and secure with respect to auxiliary input. Composable auxiliary-input VGB obfuscators for point functions 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 2.3 (ℓ-composable obfuscation VGB obfuscation with auxiliary input [BC10])**.** *A PPT algorithm $\mathcal{O}$ is an ℓ-composable VGB obfuscator for point functions 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 $(I^1, \dots, I^\ell)[q(n)]$ is an oracle that answers at most $q(n)$ queries, and where every query of the form (i, x) is answered by $I^i(x)$.

For notational convenience, since we only use point function obfuscation, we denote the oracle provided to the simulator as $I_w(\cdot, \cdot)$ where $w = w_1, ..., w_\ell$ is the vector of obfuscated points.

3 Computational Fuzzy Extractors

In this section we present our paradigm for constructing 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. Let $\mathcal{M}$ be a metric space with distance function dis.

Definition 3.1. *[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 (ϵ, s)-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 $\delta^{\mathcal{D}_s}((R, P), (U_\kappa, P)) \le \epsilon$.*

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 both our constructions it is crucial that w' is chosen independently of the outcome of Gen.

Fuller, Meng, and Reyzin [FMR13] present two approaches for constructing a computational fuzzy extractor: analyzing the information-reconciliation and privacy amplifications components together or using a fuzzy conductor and a privacy amplification component. We follow the second approach. The lower bounds on entropy loss of fuzzy extractors shown by Dodis et al. [DORS08, Section C] extend immediately to fuzzy conductors. To overcome these bounds, we use a computational version of a fuzzy conductor. We use the common notion of HILL entropy [HILL99] extended to the conditional case:

Definition 3.2. *[HLR07, Definition 3] Let (W, S) be a pair of random variables. W has* HILL entropy *at least k conditioned on S, denoted $H^{\mathtt{HILL}}_{\epsilon,s}(W|S) \ge k$ if there exists a joint distribution (X, S), such that $\tilde{\mathrm{H}}_\infty(X|S) \ge k$ and $\delta^{\mathcal{D}_s}((W, S), (X, S)) \le \epsilon$.*

We now define a computational fuzzy conductor and a (computational) randomness extractor. A computational fuzzy conductor is the computational analogue of a fuzzy conductor (introduced by Kanukurthi and Reyzin [KR09]).

Definition 3.3. *A pair of randomized procedures "generate" (Gen) and "reproduce" (Rep) is an $(\mathcal{M}, \mathcal{W}, \tilde{m}, t)$-computational fuzzy conductor that is (ϵ, s)-hard with error δ if Gen and Rep satisfy Definition 3.1, except the last condition is replaced with the following weaker condition:*

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

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 3.4. *Let χ be a finite set. A function $\mathtt{cext} : \{0,1\}^\ell \times \{0,1\}^d \to \{0,1\}^\kappa$ a (m, ϵ, s)-*average-case computational extractor *if for all pairs of random variables X, Y (with X over $\{0,1\}^\ell$) such that $\tilde{H}_\infty(X|Y) \geq m$, we have $\delta^{\mathcal{D}_s}((\mathtt{cext}(X; U_d), U_d, Y), U_\kappa \times U_d \times Y) \leq \epsilon$.*

Combining a computational fuzzy conductor and an appropriate computational extractor yields a computational fuzzy extractor (proof in Appendix B):

Lemma 3.5. *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 $\{0,1\}^\ell$. Let $\mathtt{cext} : \{0,1\}^\ell \times \{0,1\}^d \to \{0,1\}^\kappa$ 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 $s' = \min\{s_{cond} - |\mathtt{cext}| - d, s_{ext}\}$.

4 Tolerating a Constant Fraction of Errors when $\mathrm{H}_{\mathtt{usable}} \leq 0$

For the remainder of this work, we consider the Hamming metric over some alphabet $\mathcal{Z}$. Our goal is to derive strong keys for a large class of sources where $0 \geq \mathrm{H}_{\mathtt{usable}} = \mathrm{H}_\infty(W) - \log |B_t|$. Our first construction is inspired by the construction of digital lockers from point obfuscation by Canetti and Dakdouk [CD08].

Construction 4.1. *Let $\mathcal{Z}$ be an alphabet and let $W = W_1, ..., W_\ell$ be a distribution over $\mathcal{Z}^\ell$. Let $\mathcal{O}$ be an obfuscator for point functions with points from $\mathcal{Z}$. Let $C \subset \{0,1\}^\ell$ be an error-correcting code. We describe $\mathsf{Gen}, \mathsf{Rep}$ as follows:*

```
Gen

    1. Input: $w = w_1, ..., w_\ell$
    2. Sample $c \leftarrow C$.
    3. For $j = 1, ..., \ell$:
        (i) If $c_j = 0$: $p_j = \mathcal{O}(I_{w_j})$.
        (ii) Else: Sample $r_j \xleftarrow{\$} \mathcal{Z}$.
                 Let $p_j = \mathcal{O}(I_{r_j})$.
    4. Output $(c, p)$, where $p = p_1 \dots p_\ell$.
```

```
Rep

    1. Input: $(w', p)$
    2. For $j = 1, ..., \ell$:
        (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 $c$.
```

The input w is hidden in two different ways. In locations where $c_j = 1$, the block w_j is information-theoretically unknown. In locations where $c_j = 0$, it is hard to find w_j given access to the point obfuscation. There are two possible reasons for a bit c'_j to be 1: because the true value was 1 and because $w_j \neq w'_j$. However, if a bit c'_j is 0, this likely means that $w_j = w'_j$ because collisions when $c_j = 0$ are unlikely (occurring with probability $1/|\mathcal{Z}|$). This is the reason for the use of a code that only corrects $0 \to 1$ flips.

Construction 4.1 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 4.2. *Let $I_w(\cdot,\cdot)$ be an oracle that returns*

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

*A source $W = W_1||...|W_\ell$ is a (q, α, β)-*unguessable block distribution *if there exists a set $J \subset \{1, ..., \ell\}$ of size at least $\ell - \beta$ 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 distributions in Appendix A. 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 distribution (Claim A.3).

Theorem 4.3. *Let n be a security parameter. Let $\mathcal{Z} \subseteq \{0,1\}^{\omega(\log n)}$ be an alphabet. Let $\mathcal{W}$ be a family of $(q, \alpha = \omega(\log n), \beta)$-unguessable block distributions over $\mathcal{Z}^\ell$, for any $q = \mathtt{poly}(n)$. Furthermore, let C be a $(\mathsf{Neigh}_t, \delta_{code})$-code over $\mathcal{Z}^\ell$. Let $\mathcal{O}$ be an ℓ-composable VGB obfuscator for point functions with auxiliary inputs. Then for any $s_{sec} = \mathtt{poly}(n)$ there exists some $\epsilon = \mathtt{ngl}(n)$ such that Construction 4.1 is a $(\mathcal{Z}^\ell, \mathcal{W}, \tilde{m} = H_0(C) - \beta, t)$-computational fuzzy conductor that is $(\epsilon_{sec}, s_{sec})$-hard with error $\delta_{code} + \ell/|\mathcal{Z}|$.*

Proof. We first argue security in the following Lemma.

Lemma 4.4. *Let all variables be as in Theorem 4.3. 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 here; the proof is in Appendix C.

Outline. 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 4.2. 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 $\Pr[D(C', P) = 1]$.

□

We now argue correctness of Construction 4.1. We begin by showing that the probability of a single $1 \to 0$ bit flip in c is negligible.

Lemma 4.5. *Let all variables be as in Theorem 4.3. 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* $\leq \ell/|\mathcal{Z}| = \mathtt{ngl}(n)$.

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 ℓ 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 4.1 is correct with error at most $\ell/|\mathcal{Z}|$. □

4.1 Discussion of Construction 4.1

To show that $\mathrm{H}_{\mathtt{usable}}$ can be negative for Construction 4.1, we first calculate the size of the Hamming ball.

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

We consider what entropy is necessary for security. The simplest type of unguessable block distribution is where each block is independent and has super-logarithmic entropy (Claim A.1). For this type of source the required entropy is $\mathrm{H}_\infty(W) = \ell\omega(\log n)$. This yields:

$$\mathrm{H}_{\mathtt{usable}} = \mathrm{H}_\infty(W) - \log |B_t| < \ell\omega(\log n) - \left(\Theta(t \log |\mathcal{Z}|) + \log \binom{\ell}{t}\right).$$

When $t = \Theta(\ell)$ and the entropy of each block is $o(\log |\mathcal{Z}|)$, then $\mathrm{H}_{\mathtt{usable}} \leq 0$ and the output entropy is $H_0(C) - \beta$ (if C is a constant rate code, this is $\Theta(\ell)$).

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

5 Trading Errors for Entropy

Construction 4.1 is a computational fuzzy conductor with $\mathrm{H}_{\texttt{usable}} \leq 0$. Unfortunately, it requires many blocks of W to have super-logarithmic min-entropy. In this section, we reduce the required entropy of blocks by obfuscating several blocks simultaneously, at the price of decreasing the effective error tolerance. The main idea is to sample a random subset of blocks $W_{j_1}, ..., W_{j_\eta}$ and obfuscate the concatenation of these blocks. Denote this concatenated value by V_1. This process is repeated to produce $V_1, ..., V_\ell$ and the construction proceeds by either obfuscating V_i or a random point as before. For security each value V_i needs to be unguessable. This will hold as long as enough blocks contribute some entropy:

Definition 5.1. *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:*

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

Definition 5.1 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]). 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 5.2 (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, $C \subset \{0,1\}^\ell$ be an error-correcting code and $\mathcal{O}$ be an obfuscator for the family of point functions. Define* Gen, Rep *as:*

```
Gen

  1. Input: $w = w_1, ..., w_\gamma$

  2. Select $c \xleftarrow{\$} C$.

  3. For $i = 1, ..., \ell$:

       (i) Select $\lambda_i \xleftarrow{\$} \{0,1\}^{r_{sam}}$.
      (ii) Set $j_{i,1}, ..., j_{i,\eta} \leftarrow \mathsf{Sample}_{\eta,\gamma}(\lambda_i)$
     (iii) If $c_i = 0$:
               Set $v_i = w_{j_{i,1}}, ..., w_{j_{i,\eta}}$.
               Set $\rho_i = \mathcal{O}(I_{v_i})$.
               Set $p_i = \rho_i, \lambda_i$.
      (iv) If $c_i = 1$: Select $r_i \xleftarrow{\$} \mathcal{Z}^\eta$.
               Let $p_i = \mathcal{O}(I_{r_i}), \lambda_i$.

  4. Output $(c, p)$, where $p = p_1 \dots p_\ell$.
```

```
Rep

  1. Input: $(w', p)$

  2. For $i = 1, ..., \ell$:

       (i) Parse $p_i$ as $\rho_i, \lambda_i$.
      (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) If $\rho_i(v'_i) = 1$ set $c'_i = 0$.
       (v) Else set $c'_i = 1$.

  3. Set $c = \mathsf{Decode}(c')$.

  4. Output $c$.
```

The main change in Construction 5.2 is that the obfuscated values are concatenated symbols of W. This paradigm is similar to *sample-then-extract* from the locally computable extractors literature [Lu02, Vad03]. For this reason we call Construction 5.2 *sample-then-obfuscate*. A crucial difference is that the use of a computational primitive (obfuscation) 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 construction uses a naïve sampler that takes truly random samples, but the public randomness may be substantially decreased by using more sophisticated samplers. Goldreich provides an introduction to samplers in [Gol11].

Theorem 5.3. *Let $\mathcal{Z}$ be an alphabet. Let n be a security parameter. Let $\mathcal{W}$ be the family of $(\alpha = \Omega(1), \beta \le \gamma(1-\Theta(1)))$-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 ℓ be such that $\ell = O(\mathtt{poly}(n))$ and $\ell = \omega(\log n)$. Let C be a $(\mathsf{Neigh}_{t'}, \delta_{code})$ where $t' = \Theta(\ell)$. Let $\mathcal{O}$ be an ℓ-composable VGB obfuscator for point functions with auxiliary inputs. Then for every $s_{sec} = \mathtt{poly}(n)$ there exists some $\epsilon_{sec} = \mathtt{ngl}(n)$ such that Construction 5.2 is a $(\mathcal{Z}^\gamma, \mathcal{W}, \tilde{m}, t)$-computational fuzzy conductor that is $(\epsilon_{sec}, s_{sec})$-hard with error δ for*

$$\begin{aligned} t &\le \frac{-\log(1-t'/(2\ell))}{2}\frac{\gamma-\eta}{\eta} = O(\gamma/\eta) = n/\omega(\log n) \\ \tilde{m} &= H_0(C) \\ \delta &= O(\ell/|\mathcal{Z}| + 2^{-\ell} + \delta_{code}). \end{aligned}$$

The next two sections are dedicated to proving this theorem. For security we argue that each of the v_i values is unguessable. For correctness we show that the induced error rate in v and v' is a small constant (with overwhelming probability), so that c' will be corrected to c with overwhelming probability.

5.1 Security of Construction 5.2

In order to show security Construction 5.2, we show that 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. We can then argue that $V_1, ..., V_\ell$ forms a block-unguessable distribution. Then Construction 5.2 is just Construction 4.1 applied to $V_1, .., V_\ell$, and security follows by Lemma 4.4. We begin by showing that each V_i is statistically close to a high entropy distribution (proof in Appendix D). Let Λ represent the random variable of all the coins used by Sample and $\lambda = \lambda_1 \ldots \lambda_\ell$ be some particular outcome.

Lemma 5.4. *Let all variables be as in Theorem 5.3. 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,*

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

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

Corollary 5.5. *Let ϵ_{sam}, α' be as in Lemma 5.4, and all the other variables be as in Theorem 5.3. Then $\Pr_{\lambda \leftarrow \Lambda}[\forall i, \mathrm{H}_\infty(V_i|\Lambda = \lambda) \ge \alpha'] \le 1 - \ell\epsilon_{sam}$.*

In Claim A.3 we show that any distribution where each individual block has super-logarithmic min-entropy forms a unguessable block distribution. This allows us to conclude:

Corollary 5.6. *Let ϵ_{sam}, α' be as in Lemma 5.4, and all the other variables be as in Theorem 5.3. 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 distribution.*

Thus, unless the choice of λ is very unlucky, Construction 5.2 is Construction 4.1 applied to an unguessable block distribution $V_1, ..., V_\ell$. That is,

Corollary 5.7. *Let all the variables be as in Theorem 5.3. For every* $s_{sec} = \mathtt{poly}(n)$ *there exists* $\epsilon_{sec} = \mathtt{ngl}(n)$ *such that* $H^{\mathtt{HILL}}_{\epsilon_{sec}, s_{sec}}(C|P) \geq H_0(C)$.

5.2 Correctness of Construction 5.2

The argument that 1 → 0 flips from c to c' are unlikely carries over from Construction 4.1. Recall that the code C corrects up to t' flips from 0 to 1. We now show that C is up to the task with overwhelming probability, i.e., that $\Pr_{(v,v') \leftarrow (V,V')}[v' \notin \mathsf{Neigh}_{t'}(v)] < \mathtt{ngl}(n)$. The proof is in Appendix E. The correctness argument in Theorem 5.3 is obtained by subtracting the probability of a single 1 → 0 bit flip ($\ell/|\mathcal{Z}|$) and the error of the code (δ_{code}).

Lemma 5.8. *Let all the variables be as in Theorem 5.3. Then* $\Pr[v' \in \mathsf{Neigh}_{t'}(v)] \geq 1 - O(2^{-\ell})$, *where the probability is over the coins of* Gen.

5.3 Discussion of Construction 5.2

We now show Construction 5.2 can work for partial block sources when $\mathrm{H}_{\mathtt{usable}} \leq 0$. The required entropy of partial block source is $\alpha(\gamma - \beta) = \Theta(\gamma)$. We are able to correct $O(\gamma/\eta)$ errors. This yields:

$$\mathrm{H}_{\mathtt{usable}} = \mathrm{H}_\infty(W) - \log |B_t| < \Theta(\gamma) - t \log |\mathcal{Z}| = \Theta(\gamma) - \Theta(\gamma/\eta) \log |\mathcal{Z}|$$

That is, Construction 5.2 achieves $\mathrm{H}_{\mathtt{usable}} \leq 0$ when the starting alphabet is super polynomial (noting that for super polynomial size $\mathcal{Z}$ we can set η to be super logarithmic and $o(\log |\mathcal{Z}|)$). We note that for polynomial-size alphabets, Construction 5.2 will still work as long as we use a code that corrects Hamming errors in both directions (with a polynomial size alphabet the probability of 1 → 0 bits flips is noticeable); however, for a polynomial-size alphabet, $\mathrm{H}_{\mathtt{usable}} > 0$.

Acknowledgements

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

This work is supported in part by National Science Foundation grants 0831281, 1012910, 1012798, and 1218461. 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 the United States Air Force under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

References

[BBC+14] Boaz Barak, Nir Bitansky, Ran Canetti, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Obfuscation for evasive functions. In Yehuda Lindell, editor, *TCC*, volume 8349 of *Lecture Notes in Computer Science*, pages 26–51. Springer, 2014.

[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.

[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.

[BGK^{+}14] Boaz Barak, Sanjam Garg, Yael Tauman Kalai, Omer Paneth, and Amit Sahai. Protecting obfuscation against algebraic attacks. In *Advances in Cryptology - Eurocrypt (to appear)*, 2014.

[BR13] Zvika Brakerski and Guy N. Rothblum. Virtual black-box obfuscation for all circuits via generic graded encoding. Cryptology ePrint Archive, Report 2013/563, 2013. `http://eprint.iacr.org/`.

[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.

[CGR09] Jonathan Connell, James E Gentile, and Nalini Ratha. SLIC: Short-length iris codes. In *Biometrics: Theory, Applications, and Systems, 2009*, 2009.

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

[CRVW02] Michael R. Capalbo, Omer Reingold, Salil P. Vadhan, and Avi Wigderson. Randomness conductors and constant-degree lossless expanders. In *Proceedings of the Thirty-fourth annual ACM Symposium on Theory of Computing*, pages 659–668, 2002.

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

[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.

[DS05] Yevgeniy Dodis and Adam Smith. Correcting errors without leaking partial information. In *STOC*, pages 654–663, 2005.

[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.

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, pages 40–49. IEEE Computer Society, 2013.

[Gil60] Edgar N Gilbert. Capacity of a burst-noise channel. *Bell Syst. Tech. J*, 39(9):1253–1265, 1960.

[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.

[GRC09] James E Gentile, Nalini Ratha, and Jonathan Connell. An efficient, two-stage iris recognition system. In *Biometrics: Theory, Applications, and Systems, 2009*, pages 1–5. IEEE, 2009.

[HAD06] Feng Hao, Ross Anderson, and John Daugman. Combining crypto with biometrics effectively. *Computers, IEEE Transactions on*, 55(9):1081–1088, 2006.

[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.

[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.

[Lu02] Chi-Jen Lu. Hyper-encryption against space-bounded adversaries from on-line strong extractors. In *Advances in Cryptology-CRYPTO 2002*, pages 257–271. Springer, 2002.

[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/`.

[RUW11] Christian Rathgeb, Andreas Uhl, and Peter Wild. On combining selective best bits of iris-codes. In *Biometrics and ID Management*, pages 227–237. Springer, 2011.

[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.

[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 Characterizing unguessable block distributions

Definition 4.2 is an inherently adaptive definition and a little unwieldy. In this section, we partially characterize sources that satisfy Definition 4.2. The majority of the difficulty in characterizing Definition 4.2 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 2.3). We first show some sources that are unguessable block distributions (Section A.1) and then show distributions with high overall entropy that are not unguessable block distributions (Section A.2).

A.1 Positive Examples

We begin with the case of independent blocks.

Claim A.1. *Let $W = W_1, ..., W_\ell$ be a source in which all blocks W_j are mutually independent. Let α be a parameter. Let $J \subset \{1, ..., \ell\}$ 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 $(q, \alpha - \log(q+1), \ell - |J|)$-unguessable block distribution. In particular, when $\alpha = \omega(\log n)$ and $q = \texttt{poly}(n)$, then W is a $(q, \omega(\log n), \ell - |J|)$-unguessable block distribution.

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 A.1 shows that Definition 4.2 captures, in particular, this class of distributions. However, Definition 4.2 captures more distributions. We now consider more complicated distributions where blocks are not independent.

Claim A.2. *Let $f : \{0,1\}^e \rightarrow \mathcal{Z}^\ell$ 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 distribution.*

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 A.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 A.2 is e, so there is a family of distributions with total entropy $\omega(\log n)$ for which Construction 4.1 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 A.3. *Let $W = W_1, ..., W_\ell$ 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 distribution.*

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 A.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

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

□

A.2 Negative Examples

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

Proof. Let x be the true value for $U_{p \times \ell}$. 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 A.5. *Let $W = W_1, ..., W_\ell$ be a source under ordering $i_1, ..., i_\ell$. The* jump *of a block i_j is* $\texttt{Jump}(i_j) = \max_{w_{i_1}, ..., w_{i_{j-1}}} H_0(W_{i_j} | W_{i_1} = w_{i_1}, ..., W_{i_{j-1}} = w_{i_{j-1}})$.

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

If an adversary can learn blocks in succession they can eventually recover the entire secret. In order for a distribution 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 A.6. *Let W be a distribution and let q be a parameter, if there exists an ordering $i_1, ..., i_\ell$ such that for all $j \leq \ell - \beta + 1$, $\mathtt{Jump}(i_j) = \log q/(\ell - \beta + 1)$, then W is not $(q, 0, \beta)$-unguessable block distribution.*

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

B Proof of Lemma 3.5

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:

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

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. □

C Proof of Lemma 4.4

Proof. Let $\mathcal{O}$ be a ℓ-composable VGB obfuscator with auxiliary input for point programs over $\mathcal{Z}$. Let W be a $(q, \alpha = \omega(\log n), \beta)$-unguessable block distribution. 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 4.2

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 2.3), there exists a unbounded time simulator S (making at most q queries) such that

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

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

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

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

Lemma C.2. *Let c^* be true value encoded in X and let c' a codeword in C'. Then,*

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

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

Claim C.3. *If W is a (q, α, β)-block unguessable distribution over $\mathcal{Z}$ then $\alpha \leq \log|\mathcal{Z}| - \log q$.*

Proof. Let W be a (q, α, β)-block unguessable distribution. Let $J \subset \{1, ..., \ell\}$ 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:

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

This completes the proof of Claim C.3. □

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 min-entropy. If this occurs with over $2^{-\alpha}$ probability this violates the block unguessability of W (Definition 4.2). By the union bound over the indices $j \in J$ the total probability of a 1 in J is at most $(\ell-\beta)2^{-\alpha+1}$. Recall that c^*, c' match on all indices outside of J. Thus, for all c^*, c' the statistical distance is at most $(\ell-\beta)2^{-\alpha+1}$. This concludes the proof of Lemma C.2. □

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

Now by the security of obfuscation we have that

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

Combining Equations 2 and 4 and Lemma C.1, we have

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

This is a contradiction and completes the proof of Lemma 4.4. □

D Proof of Lemma 5.4

Proof. Consider some fixed i. Recall that there a set J of size $\gamma - \beta = \Theta(\gamma)$ such that each w and block $j \in J$, $\mathrm{H}_\infty(W_j|W_1 = w_1,...,W_{j-1} = w_{j-1}, W_{j+1} = w_{j+1},...,W_\gamma = w_\gamma) \ge \alpha$. 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.1.

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

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

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

Taking the negative logarithm of both sides we have that

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

This completes the proof of Claim D.1. □

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 $\Pr[X \le \mathbb{E}[X]/2] \le e^{-2((\gamma-\beta)/2\gamma)^2\eta} = O(e^{-\eta})$.

Thus, setting $\alpha' = \frac{\alpha\eta(\gamma-\beta)}{2\gamma}$ and applying Claim D.1, we conclude that

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

□

E Proof of Lemma 5.8

Proof. Define $\mu = -\frac{1}{2}\log(1 - t'/(2\ell)) = \Theta(1)$ and note that $t \le \mu(\gamma - \eta)/\eta$. Since $\eta = \omega(\log n)$, we will assume $\eta \ge 2\mu$. Let the Bernoulli random variable $X_i = 1$ if and only if $v_i \ne v_i'$, and $X = \sum_{i=1}^{\ell} X_i$. We need to show that $\Pr[X > t'] = O(2^{-\ell})$.

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

Hence, $\mathbb{E}[X_i] \leq t'/(2\ell) = O(1)$, and $\mathbb{E}[X] \leq t'/2$. By the Chernoff bound, we have

$$
\Pr\left[\sum_{i=1}^{\ell} X_i \geq t'\right] \leq 2e^{-2(t'-\mathbb{E}[X])^2\ell} \leq 2e^{-2(t'/2)^2\ell} = O(e^{-\ell})\,.
$$

□