A Negative Result for Fuzzy Extractors

Luke Demarest
University of Connecticut
luke.h.demarest@gmail.com

Benjamin Fuller
University of Connecticut
benjamin.fuller@uconn.edu

Alexander Russell
University of Connecticut
acr@uconn.edu

April 20, 2022

Abstract

We show a negative result for efficient fuzzy extractors for all distributions with fuzzy min entropy even when given a (quasi) polynomial advice string from an unbounded collaborator.

1 Introduction

We show a negative result for efficient fuzzy extractors for all distributions with fuzzy min entropy even when given a (quasi) polynomial advice string from an unbounded collaborator.

We will compare to other works like [1].

2 Preliminaries

In this section we will introduce existing results to help clarify our place in the literature, provide necessary existing definitions to show what we are borrowing and what we build on, and we will provide new definitions in support of our novel results.

2.1 Existing Definitions

Definition (Entropy)**.** *Entropy*, denoted $H(X)$, for some random variable X is a measure of how stable the outcomes of the random variable are. It is calculated as

$$H(X) := \sum_{i=1}^{n} p(x_i) \log(p(x_i))$$

where there are n values that X takes and we denote them as x_i.

Definition (Min Entropy)**.** *Min Entropy*, denoted $H_\infty(X)$, is a best case measure of the stability of the random variable X. It is calculated as

$$H_\infty(X) := -\mathsf{log}\left(\max_{x_i} p(x_i)\right)$$

.

Definition (Average Conditional Min Entropy)**.** *Average Conditional Min Entropy*, denoted $\widetilde{H}_\infty(X \mid Y)$ for two random variables X and Y is an average measure of the remaining entropy of the former given the outcome of the latter. It is calclulated as

$$\widetilde{H}_\infty(X \mid Y) := -\mathsf{log}\left(\mathop{\mathbb{E}}_{y \leftarrow Y}\left[\max_x \Pr[X = x \mid Y = y]\right]\right).$$

Definition (Hartley Entropy)**.** *Hartley Entropy* also called *Hartley's Function* measures the uncertainty of a random variable in a basic way, measuring the number of outcomes the random variable has with non-zero probability. It can be computed as

$$H_0(X) = |\{x \in X \mid \Pr[X = x] > 0\}|.$$

Definition (Markov's Inequality)**.** Markov's inequality is a tail bound for random variables that gives an upper bound on the probability of a random variable deviating from its mean. Let X be a non-negative valued random variable. Then the following inequality holds for any $\alpha > 0$:

$$\Pr[X \geq \alpha \cdot \mathbb{E}[X]] \leq 1/\alpha.$$

Definition (Secure Sketch)**.** A *secure sketch* is an abstract crytographic primative that hides information of a sample, but allows for recovery of the original sample if presented a sample that is close enough to the original sample. It is made of two algorithms: Sketch and Recover. Sketch takes the original sample and outputs some public value. Recover then takes some other sample and the value from Sketch and if the two samples are close enough, outputs the original sample.

Formal Definition here.

2.2 Previous Results

It has been shown that universal fuzzy extractors are impossible in the information theoretic setting.

2.3 Average Conditional Min-Entropy Loss

Lemma 1. *Let $\vec{X} = (X_1, X_2, \ldots, X_k)$ be independent random variables. Let Y be a random variable arbitrarility correlated with $\vec{X}$. Then*

$$\widetilde{H}_\infty\left(\vec{X} \mid Y\right) \geq \sum H_\infty(X_i) - H_0(Y)$$

Proof. Since each X_i is independent then $H_\infty\left(\vec{X}\right) = \sum H_\infty(X_i)$. Now, by definition,

$$\widetilde{H}_\infty\left(\vec{X} \mid Y\right) = -\mathsf{log}\left(\mathop{\mathbb{E}}_{y \leftarrow Y}\left[\max_{\vec{x}} \Pr\left[\vec{X} = \vec{x} \mid Y = y\right]\right]\right) \tag{1}$$

$$= -log \sum_y \max_{\vec{x}} \Pr\left[\vec{X} = \vec{x} \mid Y = y\right] \cdot \Pr[Y = y] \tag{2}$$

$$= -log \sum_y \max_{\vec{x}} \Pr\left[\vec{X} = \vec{x} \vee Y = y\right] \tag{3}$$

$$\geq -log \sum_y \max_{\vec{x}, y'} \Pr\left[\vec{X} = \vec{x}^Y = y'\right] \tag{4}$$

$$= -\mathsf{log}\left(2^{H_0(Y)} \cdot 2^{H_\infty\left(\vec{X}, Y\right)}\right) \tag{5}$$

$$= H_\infty\left(\vec{X}, Y\right) - H_0(Y) \tag{6}$$

$$\geq H_\infty\left(\vec{X}\right) - H_0(Y) \tag{7}$$

$$= \sum H_\infty(X_i) - H_0(Y) \tag{8}$$

□

2.3.1 Markov Bound for Predictability

Markov bounds are tail bounds that use Markov's Inequality to bound the probability that a random variable deviates significantly from its expected value. In Markov's inequality, we necessarily lose a multiplicative factor (here called *alpha*) in order to control the probability of the event occuring. When discussing entropy, we are dealing with a log scaled value which makes losing multiplicative factors costly. Instead, we can perform a Markov bound on the predictability scale. In this case, rather than lose a multiplicative factor in entropy, we lose a multiplicative factor in predictability which translates to a small number of bits of entropy lost for the controlled outcomes.

Lemma 2. *Let $\vec{X} = (X_1, X_2, \ldots, X_k)$ be independent random variables. Let Y be a random variable arbitrarility correlated with $\vec{X}$. Let $\alpha > 0$, then for all but a $(1 - 1/\alpha)$ fraction of the X_i the entropy loss is less than $\mathsf{log}(\alpha)/k$*

Proof.

$$\widetilde{H}_\infty\left(\vec{X} \,|\, Y\right) = \Delta \tag{9}$$

$$-\mathsf{log}\left(\mathop{\mathbb{E}}_{Y}\left[\max_{\vec{x}} \Pr\left[\vec{X} = \vec{x} \,|\, Y = y\right]\right]\right) = \Delta \tag{10}$$

$$\mathop{\mathbb{E}}_{Y}\left[\max_{\vec{x}} \Pr\left[\vec{X} = \vec{x} \,|\, Y = y\right]\right] = 2^{-\Delta} \tag{11}$$

$$\Pr_{Y}\left[\max_{\vec{x}} \Pr\left[\vec{X} = \vec{x} \,|\, Y = y\right] \geq \alpha \cdot 2^{-\Delta}\right] \leq \frac{1}{\alpha} \tag{12}$$

$$\Pr_{Y}\left[\mathsf{log}\left(\max_{\vec{x}} \Pr\left[\vec{X} = \vec{x} \,|\, Y = y\right]\right) \geq \mathsf{log}(\alpha) - \Delta\right] \leq \frac{1}{\alpha} \tag{13}$$

$$\Pr_{Y}\left[-\mathsf{log}\left(\max_{\vec{x}} \Pr\left[\vec{X} = \vec{x} \,|\, Y = y\right]\right) < \Delta - \mathsf{log}(\alpha)\right] \leq \frac{1}{\alpha} \tag{14}$$

$$\Pr_{Y}\left[H_\infty\left(\vec{X} \,|\, Y\right) < \Delta - \mathsf{log}(\alpha)\right] \leq \frac{1}{\alpha} \tag{15}$$

□

2.3.2 Upper bound for size of a Fuzzy Extractor

In [1], Fuller et al. show that the size of a fuzzy extractor can be upper bound in a general case. We restate their lemma here for completeness of our main proof.

Lemma 3 (Lemma 5.2 in [1])**.** *Suppose $\mathcal{M}$ is $\{0,1\}^n$ with the Hamming Metric, $\kappa \geq 2$, $0 \leq t \leq n/2$, and $\epsilon > 0$. Suppose $(\mathsf{Gen}, \mathsf{Rep})$ is a $(\mathcal{M}, \mathcal{W}, \kappa, t, \epsilon)$-fuzzy extractor with error $\delta = 0$, for some distribution family $\mathcal{W}$ over $\mathcal{M}$. Let $\tau = t/n$. For any fixed p, there is a set $\mathsf{GoodKey}_p \subseteq \{0,1\}^\kappa$ of size at least $2^{\kappa-1}$ such that for every $\mathsf{key} \in \mathsf{GoodKey}_p$,*

$$\mathsf{log}(|\{v \in \mathcal{M} | (\mathsf{key}, p) \in \mathsf{supp}(\mathsf{Gen}(v))\}|) \leq n \cdot h_2\left(\frac{1}{2} - \tau\right) \leq n \cdot \left(1 - \frac{2}{\ln 2} \cdot \tau^2\right),$$

and, therefore, for any distribution $D_\mathcal{M}$ on $\mathcal{M}$,

$$H_0(D_\mathcal{M} | \mathsf{Gen}(D_\mathcal{M}) = (\mathsf{key}, p)) \leq n \cdot h_2\left(\frac{1}{2} - \tau\right) \leq n \cdot \left(1 - \frac{2}{\ln 2} \cdot \tau^2\right).$$

2.4 New Definitions

Fuzzy extractors with quasipolynomial advice.

3 Sampling Proceedure

The sampling proceedure that we will use and discuss in this work is two fold. There is first, a family of distributions that we will call $\mathcal{W}$. There is some

finite number of distributions in this family, we call $|\mathcal{W}| = \mathcal{R}$. We let Z be an index for the distributions in the family and we denote the Zth distribution W_Z. The distribution W_Z can then be sampled, and we denote a sample of W_Z as $w_Z \in \{0,1\}^n$. When Z is clear, we will omit the subscript.

To sample a uniform point in $\mathcal{W}$, we can uniformly select Z and then pick from W_Z unifromly. This two stage process gives us more tools to reason about the ability of an efficient adversary.

In general, we will assume that $\mathcal{W}$ is public to any adversary, algorithm, or party we may discuss. We will generally assume that the specific selection of Z is only shared with specific parties. We will also assume that the specific w sampled from W_Z is private and only known to parties who are given it explicity.

An interesting part of this work is how we share informtaion about W_Z. We will allow an inefficient adversary access to the entire description of W_Z and ask them to produce an advice string for an efficient adversary, that is that the advice string length is bound from above by a arbitrary polynomial function of our security parameter.

3.1 Distributions over $\{0,1\}^n$

1. Picking k points from $\{0,1\}^n$ uniformly results in a distribution we will denote $U_{n,k}$. Clearly, this is efficient with respect to $\max(n,k)$. The only condition here is that $k \leq 2^n$

2. Another distribution of interest is picking k points uniformly from $\{0,1\}^n$ and then removing points that are within distance t of one another, we denote this $U^-_{n,k,t}$. This is also efficient, but may result in fewer than k points being included in the final set. The trivial bounds here also include $t \leq n$.

3. Another way of acheieving a simillar result is by picking points until you have k (or a failure condition), that all have distance at least t from one another. This is not guaranteed to terminate without an error condition. The trivial bounds from above apply here as well.

4 Proof of Main Theorem

4.1 Proof Sketch

Our proof follows a fairly predictable structure. We begin by borrowing an existing result from [1] which gives an upperbound on the size of a set of viable points used by any good fuzzy extractor. You can find the theorem statement in Lemma 3. We then further restrict this setting by showing that in order for an adversary to succeed on average they have to be able to align these viable points with a distribution that they have only a single sample and a polynomial length advice sting. We then argue that for large high entropy distributions this advice string can only reduce the entropy of a large fraction of viable points by a small

amount. Then, we show that this small reduction of entropy for each point in the distribution means that on average the adversary cannot align the viable points with the distribution and there exists a distiguisher that can distinguish a uniform triple from a key triple.

4.2 Proof Setting

Consider $\mathcal{W}$ such that each $W \in \mathcal{W}$ is a set of 2^{ϕ} uniformly chosen independent random points in $\{0,1\}^n$. Let $|\mathcal{W}| = r$ and let $Z \in [r]$ be an indexing variable for the selection of W_Z from $\mathcal{W}$.

In this proof the goal is to show that a fuzzy extractor cannot hope to hide the input point from an outside party. We will call the party that is building the Fuzzy Extractor the constructor and the party attempting to distiguish two settings the distinguisher. The Fuzzy Extractor is set up with an enrolled sample from our two stage sampling proceedure before the distiguishing game begins. The game that the distinguisher plays is given one of two triples, real or random, do distinguish the realm to which the triple belongs. The real triple is the key value corresponding to the enrolled value in the fuzzy extractor, the public value produced by Gen, and the description of the distribution W_Z. The random triple is the same except the key value corresponding to the enrolled value is substituted with a uniform value in the domain of the key values.

In this setting the constructor when creating the Fuzzy Extractor is aided by another party, we call this party the advisor. The advisor gets a full description of the distribution W_Z and is allowed unbounded computation and time to produce an advice string info. The advisor and constructor are unbounded in their computation and time, but the length of this string is required to be polynomial in the security parameter λ. The advice string is then communicated to the constructor, who also gets a sample from the distribution $w \in W_Z$. The constructor then is tasked with creating a Fuzzy Extractor, specifically choosing a public value pub which induces a partition over the space $\{0,1\}^n$ and a key labeling of the partition which determines a key value for the initial sample.

4.3 Maximum size of a Fuzzy Extractor

From Lemma 3, we have the largest size of a set of viable points is 2^{ψ} where $\psi = n \cdot \left(1 - \frac{2}{\ln 2} \cdot \tau^2\right)$.

Now, it is clear from our setting that when building the Fuzzy Extractor the constructor has a single point w, a full description of the family of distributions $\mathcal{W}$, and the prepared advice string info about the specific selected distribution W_Z. We are primarily concerned with the ability of the constructor to align the points in the selected W_Z, with the partition induced by pub. Clearly, if info is allowed to describe every point in W_Z (or every point but w) then the constructor can include a point from W_Z in each viable section of the partition induced by pub. Fortuntely for us, the distribution has exponential entropy and the advice string has polynomial length so the advice string cannot describe the

entire remainder of the distribution. The question remains, how much entropy remains in the distribution after seeing the advice string?

Since each point in the distribution is independent and uniform in $\{0,1\}^n$ the beginning entropy (and min-entropy) of the distribution is $|W_Z| \cdot n$. Now we consider the advice string; since this string is allowed to arbitrarily depend on the distribution we can upperbound the min-entropy of the distribution conditioned on the advice string using a standard min-entropy argument found in Lemma 1.

$$\widetilde{H}_\infty(W_Z \mid \mathsf{info}) = |W_Z| \cdot 2^n - \mathsf{log}(|info|)$$

References

[1] Benjamin Fuller, Leonid Reyzin, and Adam Smith. When are fuzzy extractors possible? *IEEE Transactions on Information Theory*, 66(8):5282–5298, 2020.