FiveEyes: Cryptographic Biometric Authentication from the Iris

Luke Demarest[*], Sohaib Ahmad[†], Sixia Chen[‡], Benjamin Fuller[§], Alexander Russell[¶]

January 22, 2024

Abstract

Despite decades of effort, a stubborn chasm exists between the theory and practice of device-level biometric authentication. Deployed authentication algorithms rely on data that leaks private information about the biometric; thus systems rely on externalized security measures such as trusted execution environments. In particular, the authentication algorithms themselves provide no cryptographic security guarantees.

This is particularly frustrating given the long line of research that has developed theoretical tools—known as fuzzy extractors—that enable secure, privacy preserving biometric authentication with *public* enrollment data. Unfortunately, the best known constructions involving these rigorous tools can only provide substantial true accept rates with an estimated security of 32 bits for the iris (Simhadri et al., ISC 2019) and 45 bits for the face (Zhang, Cui, and Yu, ePrint 2021/1559).

This work introduces FiveEyes, an iris key derivation system that integrates an improved feature extractor with a fuzzy extractor that leverages a new mechanism, which we formally analyze, for selecting verification subsets based on statistics of the iris. (These statistics are computed from a class disjoint dataset from our test set.) We present various parameter regimes in order to highlight different true accept rates:

1. 65 bits of security (equivalent to 87 bits with a password) at 12% true accept rate, and
2. 50 bits of security (equivalent to 72 bits with a password) at 45% true accept rate.

We remark that powerful techniques are known that amplify true accept rates (Davida et al., IEEE S&P 1998); in particular, for the first time these results indicate practical viability of biometric authentication with strong cryptographic security.

1 Introduction

Users authenticate to devices using biometrics. There is a qualitative disconnect between the desirable security guarantees offered—in principle—by theoretical approaches and those deployed in current practice.

Currently deployed biometric authentication algorithms require enrollment data that exposes private information about the biometric. As biometrics are typically immutable, information leakage is a non-recoverable event; there are practical attacks in the event of an exposure [GRGB+12, FJR15, AF20, AMF22, TKAK23]. The common approach to mitigate this leakage threat is to place the authentication algorithm in a trusted execution environment, which are difficult to design [PAB+18, KHF+20, LSG+18].

The cryptography community has identified and studied the formal notion of a *fuzzy extractor* [DORS08, DKRS06, FMR13, CFP+16, ACEK17, ABC+18, WLH18, WL18, DFR21, ACF+22, BBR88, ŠTO05, HAD06],[1] which offers security guarantees *even with public enrollment information*; in particular, the biometric itself is protected from exposure or leakage if the enrollment data used to authenticate the biometric is revealed. While this appears

[*]Gonzaga University onlylukejohnson@gmail.com.
[†]University of Connecticut sohaib.ahmad@uconn.edu.
[‡]Adelphi University chensixia09@gmail.com.
[§]University of Connecticut benjamin.fuller@uconn.edu.
[¶]University of Connecticut. acr@uconn.edu.

[1]We do not review literature on interactive protocols [BDK+05, DKRS06, BG11, EHKM11, DKK+12, BCP13, BDCG13, DCH+16, DHP+18].

to solve the problem, these techniques have never provided sufficient *concrete security* for practice: thus, while they offer a comprehensive notion of security, the concrete parameters have never been sufficient for deployment.

We develop an integrated feature extractor for the iris and a fuzzy extractor that provides—for the first time—security guarantees consistent with practical demands. In particular, we estimate (selected by extensive parameter analysis, see Section 9),

1. 65 bits of security, 87 bits with a password,[2] at 12% true accept rate (TAR), and
2. 50 bits of security, 72 bits with a password, at 45% TAR.

These results place cryptographic guarantees in the range for practice. Substantial engineering is necessary to boost such true accept rates to those necessary for local device authentication, see Section 2.3.

Typical discussions of cryptographic guarantees provided by biometric authentication algorithm focus on the resulting number of "bits of security," intuitively reflecting the maximum number of bits of security in a secret key unlocked by a correct biometric input. One attack security measures is a straightforward brute-force enumeration of relevant biometrics. Thus, security provides some level of privacy of the biometric (roughly the unpredictability of the biometric is at least the security of the key). We construct fuzzy extractors that provide stronger properties: they leak no information about the biometric unless a successful attack is launched on the underlying key. This is called a private fuzzy extractor [DS05]. For this reason, the number of bits of security is a single metric that simultaneously reflects both the security and privacy properties of the construction.

Organization The rest of this work is organized as follows: Section 2 provides an overview of the system and technical contributions, Section 3 introduces more related work, Section 4 introduces mathematical preliminaries, Section 5 describes the datasets, train/test splits, and metrics of interest, Section 6 introduces our new feature extractor, Section 7 formally describes the fuzzy extractor, Section 8 describes the major technical change to our fuzzy extractor, Section 9 evaluates, and Section 10 concludes.

2 System Overview

`FiveEyes` is a combination of a **feature extractor** built from a convolutional neural network (CNN) and a **fuzzy extractor** built from the sample-then-lock fuzzy extractor [CFP+16]. The overall system is shown in Figure 1. We briefly review sample-then-lock to give context for discussing our contributions.

For $n = 1024$, let $W \in \{0,1\}^n$ be the probability distribution of the iris after applying a feature extractor. The *sample-then-lock* construction in enrollment samples uniform subsets $\mathcal{I}_1, ..., \mathcal{I}_\beta$ of $[0, n-1]$. One uses w restricted to those bits as the input value to a digital locker [CD08]. A digital locker is a symmetric encryption that is secure with correlated keys that only have entropy [CKVW10].[3] The system sets β different subsets as input to the digital locker with the same key as output.

The intuition for the construction is simple, 1) take as large subsets as possible so that each is hard to guess and 2) make β large enough so that two readings of the same biometric are likely to match exactly when restricted to the bits in some $\mathcal{I}_j$.

On the iris using the ND-0405 dataset [BF16] (which we also use), Simhadri, Steel, and Fuller [SSF19] showed for (small modifications to) an open-source feature extractor [ODGS16], sample-then-lock achieves 32 bits of security with a 60% TAR using $\beta = 10^6$.

To set notation, a fuzzy extractor is a triple of algorithms $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Rep})$. The Setup algorithm tailors the fuzzy extractor to the biometric of interest. It gives advice to Gen and Rep denoted as $\mathtt{stats}_W$. There are two goals:

1. **Correctness** For repeated readings from the same biometric, w, w', it should be the case that for $(\mathsf{key}, p) \leftarrow \mathsf{Gen}(w, \mathtt{stats}_W)$,
$$\Pr[\mathsf{key} = \mathsf{Rep}(w', p, \mathtt{stats}_W)] \geq 1 - \delta = \text{TAR}.$$

[2] Recent estimates of password entropy are 22 bits [KSK+11, Bon12, WZW+16].

[3] The formal definition is based on virtual black-box obfuscation [BGI+01].

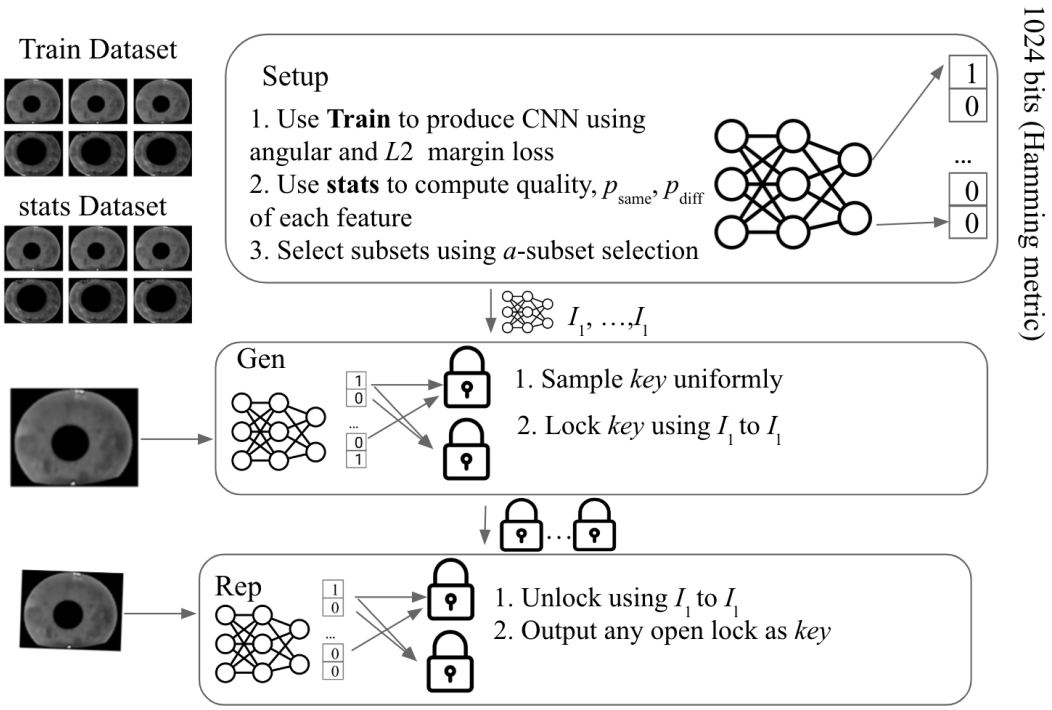


Figure 1: Overall System Architecture

2. **Security** The value key should be pseudorandom to an adversary that knows the public value, p.

The majority of our technical contributions are in Setup where the feature extractor is trained, our system selects global subsets to be used by all users. A CNN is trained to produce 1024 bit vectors, where readings of the same biometric are close according to the Hamming metric. We introduce loss functions for the goal of cryptographic key derivation that are discussed in the next subsection.

Second, we modify the sampling algorithm of sample-then-lock to choose subsets non-uniformly. Consider a single feature index i and let

$$p_{\mathtt{same},i} := \Pr[w_i \neq w'_i | w, w' \text{ readings same biometric}]$$
$$p_{\mathtt{diff},i} := \Pr[w_i \neq w'_i | w, w' \text{ readings different biometrics}]$$

These two values represent the probability of disagreement between two readings of the same biometric and readings of different biometrics respectively. During Setup we also compute these vectors and use these vectors to select subsets trading off between the entropy of a subset and how likely it is to match exactly on an image from the $\mathtt{stats}_W$ dataset. As we discuss in Section 5, we consider three class-disjoint datasets in this work, one for training the feature extractor, one for computing $\mathtt{stats}_W$, and one for evaluating correctness and security of the fuzzy extractor. An important part of our evaluation is showing that subsets selected in this way also display good entropy and TAR on biometrics not in the $\mathtt{stats}_W$ dataset.

The output of Setup is a trained CNN and subsets used for the sample-then-lock fuzzy extractor. Generate and reproduce are then straightforward using this improved CNN and subsets. We discuss the two technical improvements next.

2.1 Feature Extractor

Usually feature extractors transform iris images into feature vectors in $\{0,1\}^n$. Their goal is to maximize the tradeoff between TAR and the false accept rate (FAR). Our feature extractor uses the architecture and training regime of

ThirdEye [AF19] with new loss functions. Let the mean fractional Hamming distance between two readings of the same iris be μ and the feature extractor yields features with an estimated entropy of e. The goal of our feature extractor is to maximize the entropy that will be remaining after applying the fuzzy extractor. This is a complicated quantity to compute; it involves the particular fuzzy extractor. In our setting, one needs to analyze the entropy of individual subsets which is time consuming. Instead, we adopt the remaining entropy if one were to apply an information-theoretic fuzzy extractor. Specifically, a fuzzy extractor that writes down the syndrome of the input point using a perfect code [DORS08, FRS20]. This number represents a lower bound on the amount of security one can expect from an ideal fuzzy extractor. To correct μ fraction of errors this syndrome must be at least $n * h_2(\mu)$ bits where h_2 is the binary entropy (of a binary random variable with probability μ of being 1) and n is the length of the feature extractor. Thus, we adopt as the figure of merit

$$\mathtt{FE}_{\mathtt{Qual}} := \mathrm{H}_\infty(W) - n * h_2(\mu).$$

where H_∞ is the estimated min-entropy. For all analyzed feature extractors this quantity is negative, but $\mathtt{FE}_{\mathtt{Qual}}$ is a single quantity to meaningfully compare feature extractors. Our feature extractor represents a large improvement over prior work in this measure, see Table 1.

Our loss functions consider the error rates for like and unlike comparison but also the variance of errors in unlike comparison which is directly used in $\mathsf{EntTest}$ which is described in Section 5.2. Our loss function is a combination of L2 hard triplets (from ThirdEye [AF19]) and inner product to approximate angular distance (from SphereFace [LWY+17]).

2.2 Fuzzy Extractor

Sample-then-lock was designed for uniform subsets $\mathcal{I}_i$. Such uniform subsets treat all features as equally good. If one were only sampling one subset one would pick the features with the lowest values of $p_{\mathtt{same},i}$. However, such an approach doesn't do well when taking many subsets as it gets stuck with the subset most likely to not have errors. For meaningful security, subsets are of size ≥ 50 and so each subset has a small probability of being correct of $\leq 10^{-5}$. On the other hand, the uniform approach does very well with a large number of subsets as they are all equally likely. We introduce a new approach called ζ-sampling that moderates between these extremes. For a parameter $\zeta \in \mathbb{N}$ instead of picking bits uniformly a bit i is picked with probability proportional to $(1 - p_{\mathtt{same},i})^\zeta$, that is,

$$\text{Prob select dimension } i = \frac{(1 - p_{\mathtt{same},i})^\zeta}{\sum_i (1 - p_{\mathtt{same},i})^\zeta}\,.$$

The idea of this approach is that ζ allows one to choose how diverse to make subsets. $\zeta = 0$ represents uniform sampling while $\zeta = \infty$ only picks the bit(s) with the lowest error. We show—both empirically and analytically—that this approach outperforms uniform sampling. The sampling scheme we implement in Setup is slightly more complicated, adopting the weighting $(1 - p_{\mathtt{same},i})^{\zeta / h_2(p_{\mathtt{diff},i})}$ where h_2 is binary entropy. This weighting is used to incorporate the security accrued by each bit.

Stronger security measure In addition, we use a stronger security metric than prior work. Our security estimate is **the minimum of the entropies of subsets**. We choose subsets globally and compute the minimum of entropy across all subsets. These subsets can be used for any user.

Provably accurate entropy estimation [VV10, VV11] requires an exponentially large number of samples in the actual entropy of the distribution. However, there are established techniques for estimating the min-entropy of biometrics [Dau04a] (detailed in Section 5.2). Let $\mathsf{EntTest}$ be an entropy test for biometric values for a dataset DSet. That is, $\mathsf{e} = \mathsf{EntTest}(\mathsf{DSet})$. One advantage of sample-then-lock is that it allows one to use any such test. Simhadri et al.'s 32 bit estimate was based on the following experiment:

1. Compute some β subsets $\mathcal{I}_1, ..., \mathcal{I}_\beta$.
2. Compute $\mathsf{e}_i = \mathsf{EntTest}(\mathsf{DSet}_{\mathcal{I}_i})$. Where $\mathsf{DSet}_{\mathcal{I}_i}$ is the restriction of the dataset to the features described in $\mathcal{I}_i$.
3. Output $\tilde{\mathsf{e}} := -\log \sum_i 2^{-\mathsf{e}_i}$ as the *average min-entropy* of a subset.

Simhadri et al. only computed the *average min-entropy* across 10 subsets, we compute the minimum of min-entropy across all used subsets. For one set of subsets (used to form our high entropy parameters) the minimum of min-entropy is 52 and average min-entropy is 66 (see Table 5). We show that Simhadri et al.'s metric is incorrect based on an incorrect proof by Canetti et al. [CFP+21, Theorem 1]. We provide a corrected proof is this work, showing that given an ideal digital locker the correct figure of merit is the minimum of the e_i. This is used throughout our analysis.

2.3 Summary of Evaluation

We use $\beta = 200,000$ subsets for the mid-security and $\beta = 400,000$ subsets for the high-security. Simhadri et al. [SSF19] used $\beta = 10^6$. We perform extensive analysis on our chosen subsets and show the chosen subsets can be global for all users of the fuzzy extractor. This also removes most of the randomness and running time from Gen as one only needs to pick a random key and sample digital lockers. In prior analysis [SSF19], sampling random subsets represented the majority of the time of Gen. Our feature extractor, resulting statistics, chosen subsets, and code are open-sourced. (The ND-0405 dataset is licensed and is not included in our repository.)

We modify the Gen and Rep of Simhadri et al. [SSF19] to work with our sampling. Simhadri et al. [SSF19] reported a Gen time of 220s and a Rep time of 22s with a parallel implementation on a server machine with 4 Xeon E5-2620 v4 processors. Our modification of their implementation for the `Mid` security parameters (200K lockers in place of 1M), Gen takes 44s (with a variance in 0.45s) and Rep takes 8.6s (with a variance of 23s) on a single core of an M1 Mac.[4] Rep has a higher variance as it stops as soon as one locker "opens." This is roughly 10K lockers tested per second per core. Building this system in a lower-level language will likely yield a 1 or 2 order of magnitude improvement.[5]

Boosting TAR The biometrics community has techniques for boosting the accuracy of recognition in practice. For example, a common practice is to take three readings of an iris and for each bit i report the value that occurred in the majority of readings [DFM98, ZD08, ICF+15]. One can do averaging at enrollment and/or authentication time. The majority of our evaluation does not perform such averaging to be consistent with prior work. However, such techniques are powerful. For the first row of mid and high Security parameters in Table 6, averaging at authentication time boosts TAR from 51% to 81% and from 14% to 41% respectively. Such a change has no impact on security. These techniques motivate our focus on meaningful security with nonzero TAR, it seems harder to "boost" security than TAR.

3 Further Related Work

Throughout, we focus on computational security due to additional negative results on providing information-theoretic security [FRS16, FP19, FRS20, Ful23]. Fuzzy min-entropy is the necessary for security of a fuzzy extractor [FRS16, FRS20]. Fuzzy min-entropy requires that for all points w^*, the total probability of all $w \in W$ that would reproduce the key on w^* is negligible. The only theoretical constructions with security for all distributions with fuzzy min-entropy [FRS16, FRS20] are based on a new subset product assumption [GZ19] and based on general-purpose obfuscation techniques [BCKP14, BCKP17, PST13, GGH+13b, GGH13a, CHL+15, MSZ16, GPSZ17, MZ17].[6] The subset product assumption is directly the security of the proposed construction.

We focus on constructions that 1) state sufficient statistical properties for the security of the fuzzy extractor, 2) provide statistical evidence for those properties and 3) provide a prototype implementation.

A strong assumption is that bits of W are i.i.d. Good constructions are known in the i.i.d. setting [Mau93, MW96, MTV09, YD10, HMSS12]. However, all statistical analysis of biometrics shows that bits of W are not i.i.d. [Dau04a]. As a result one can attempt to produce independent features. Hine et al. [HKMC23] show a variant

[4] This data is collected from the first 40 classes with using the first template for Gen and running up to ten Rep for each biometric in the testing set.

[5] Bernstein estimates hashing a 64 byte message using SHA512 requires ≈ 800 cycles on a modern AMD processor `https://bench.cr.yp.to/results-hash.html`. If HMAC only consisted of two calls to SHA512 this would correspond to a speed of 10^6 lockers tested per second.

[6] We elide constructions that require exponential time [HR05, FRS16, WCD+17].

of principal/independent component analysis designed to produce independent features while controlling how much noise is in the new features. However, such algorithms only consider the covariance between two features. All our testing shows the it is larger subsets of features that display correlation, on the order of 10s of bits. However, such an algorithm may produce "better" features that are suitable for input to our algorithms.

A weaker condition than i.i.i was introduced by Canetti et al. [CFP+21]: an average subset of bits of W has min-entropy.[7] The sample-then-lock construction [CFP+21] works for such biometric sources.

4 Cryptographic Preliminaries

We use capital letters for random variables. For a set of indices J, X_J is the restriction of X to the indices in J. For integers a, b, $x_{a..b}$ denotes the restriction of vector x to the bits between a and b. U_n denotes the uniformly distributed random variable on $\{0,1\}^n$. Logarithms are base 2. A function $\nu(\lambda)$ is negligible if in the limit it shrinks faster than every inverse polynomial function $p(\lambda)$. The binary entropy function is denoted h_2 and is computed as $h_2(p) = -p\log(p) - (1-p)\log(1-p)$ The *min-entropy* of X is $\mathrm{H}_\infty(X) = -\log(\max_x \Pr[X = x])$. We use the notion of average min-entropy to measure the conditional entropy of a random variable.

Definition 1. *The* average *min-entropy of X given Y is*

$$\tilde{\mathrm{H}}_\infty(X|Y) = -\log\left(\mathop{\mathbb{E}}_{y\in Y}\max_x \Pr[X = x|Y = y]\right).$$

We write the *computational distance* between X and Y as $\Delta^c(X,Y) = \max_{\mathrm{PPT}\ D} |\mathbb{E}[D(X)] - \mathbb{E}[D(Y)]|$. For two vectors $x, y \in \{0,1\}^n$ let $\mathsf{dis}(x,y)$ be the Hamming distance between x and y. That is, $\mathsf{dis}(x,y) = |\{i|x_i \neq y_i\}|$.

We use the version of fuzzy extractors that provides security against computationally bounded adversaries [FMR13]. In addition, we extend these definitions to include a setup algorithm that is used globally. Dodis et al. provide comparable definitions for information-theoretic fuzzy extractors [DORS08].

Definition 2. *Let $\mathcal{W}, \mathcal{W}'$ be a families of correlated random variables over metric space $(\{0,1\}^n, \mathsf{dis})$. For each $W, W' \in \mathcal{W}, \mathcal{W}'$ let $\mathtt{stats}_W = f(W, W')$ be a function of W, W' of size at most $\mathtt{poly}(n)$. A triple of randomized procedures "setup" (Setup), "generate," (Gen) and "reproduce" (Rep) is an $(\mathcal{M}, \mathcal{W}, \mathcal{W}', \kappa)$*-computational fuzzy extractor *with error δ if Setup, Gen and Rep satisfy the following properties:*

- Correctness: *For random variables $(W, W') \in \mathcal{W}, \mathcal{W}'$ let $\mathtt{advise}_W \leftarrow \mathsf{Setup}(\mathtt{stats}_W)$ and $(w, w') \leftarrow (W, W')$, $(\mathsf{key}, p) \leftarrow \mathsf{Gen}(w, \mathtt{advise}_W)$,*
$$\Pr[\mathsf{Rep}(w', p) = \mathsf{key}] \geq 1 - \delta.$$
- Security*: for any pair of random variables $(W, W') \in \mathcal{W}, \mathcal{W}'$, define $\mathtt{advise}_W \leftarrow \mathsf{Setup}(\mathtt{stats}_W)$ and $(R, P) \leftarrow \mathsf{Gen}(W, \mathtt{stats}_W)$ then*
$$\Delta^c((R, P, \mathtt{advise}_W), (U_\kappa, P, \mathtt{advise}_W)) \leq \mathtt{ngl}(n).$$

Remarks Since the adversary is defined after the choice of W, W' the adversary implicitly knows the value of $\mathtt{stats}_W$. The adversary receives $\mathtt{advise}_W$ to allow for randomized Setup (which we use). We do not explicitly tackle the notion of reusable [Boy04] or robust fuzzy extractors [DKRS06] in this work. Reusable fuzzy extractors allow one to enroll noisy readings of a biometric multiple times, sample-then-lock is reusable and the use of a global Setup does not change this as long as one uses a sufficiently composable digital locker.

5 Datasets and Metrics

5.1 Dataset and Feature Extractor Training

Throughout, we use the ND-0405 iris dataset [BF16] which is a superset of the NIST iris evaluation challenge [PBF+08]. This dataset consists of 356 individuals with images of both eyes representing 712 biometrics. Left and right eyes

[7] See Demarest et al. [DFR21, Figure 1] for an overview of known constructions and corresponding statistical requirements.

are considered independent biometrics [Dau04b]. ND-0405 is captured at a near-infrared wavelength. The dataset consists of 64,980 images. We use the same training regime as in ThirdEye [AF19]. However, we split their testing set into two sets, one used for producing $\texttt{stats}_W$ and one for testing.

Training For training, we used the first 25 images of the left irises from all individuals.

$\texttt{stats}_W$ 70 of the 356 right eyes are reserved to compute $\texttt{stats}_W$. This represents 20% of both classes and images that were not used for training.

Testing The remaining 286 right eyes are used for computing test data.

Training, calculation of $\texttt{stats}_W$, and testing are all class disjoint.
For histograms shown in Figure 2, 10 randomly chosen images were taken from the union of $\texttt{stats}_W$ and testing (all images were used if an iris has fewer than 10 images). No such restriction is done in the remainder of testing. Images are segmented (iris portion separated from background) before input to the feature extractors. Segmentation is performed using [AF18] which is trained using human-labeled ground truth [Pro09]. Images have a resolution of 640×480 while segmented images have a resolution of 256×256.

5.2 Metrics

Entropy Test Throughout this work, we use the standard method of Daugman [Dau04a] for estimating the entropy of biometric feature extractors. This method was used for estimating min-entropy in sample-then-lock in Simhadri et al. [SSF19, Section 6]. This method is as follows:
$\mathsf{EntTest}(\mathsf{DSet})$:

1. Compute a histogram of all distances (fractional Hamming between the binary vectors) between readings of different biometrics (the red histogram in Figure 2).
2. Find the mean μ and standard deviation σ of this histogram.
3. Compute the degrees of freedom $\mathtt{dF} = \mu(1-\mu)/\sigma$.
4. Entropy is $\mathsf{e} = -(\mu * \log(\mu) - (1-\mu) * \log(1-\mu)) * \mathtt{dF}$.

This leads us to our first assumption which along with Assumption 2 suffices for the security of the scheme.

Assumption 1. *For a dataset* DSet *the test* $\mathsf{EntTest}$ *accurately measures the min-entropy of the distribution of biometrics from which* DSet *is drawn.*

As stated in the Introduction, one can execute the $\mathsf{EntTest}$ described above on a subset of features of the biometric (representing sampling). One can also execute $\mathsf{EntTest}$ on a subset of biometrics, in some of our tests we sample a subset of biometrics to improve efficiency. $\mathsf{EntTest}$ requires quadratic time in the size of DSet.

Computing TAR All assessments of TAR take as input a collection of subsets $\mathcal{I}_1, ..., \mathcal{I}_\beta$. For input DSet, let DSet_i represent all readings of a single biometric. We compute the following:

1. Set $\texttt{TAR}_{\texttt{num}} = 0, \texttt{TAR}_{\texttt{denom}} = 0$.
2. Randomly sample 200 classes of DSet. For each class:
   (a) Pick the first biometric (lexicographically according to file names) as w^*. Compute $w^*_{\mathcal{I}_1}, ..., w^*_{\mathcal{I}_\beta}$.
   (b) Let $w_1, ..., w_\gamma$ be the remainder of readings for the biometric w^*.
   (c) Set $\texttt{TAR}_{\texttt{denom}} = \texttt{TAR}_{\texttt{denom}} + \gamma$.
   (d) For $j = 1$ to γ: if there exists some i such that $w_{j,\mathcal{I}_i} = w^*_{\mathcal{I}_i}$, then $\texttt{TAR}_{\texttt{num}} = \texttt{TAR}_{\texttt{num}} + 1$.
3. Output $\texttt{TAR} = \texttt{TAR}_{\texttt{num}}/\texttt{TAR}_{\texttt{denom}}$.

Notes The number of biometrics per class can vary from at little as 4 to as many as 191. This means that TAR is weighted by the number of samples for a biometric. The median class (among the 286) has 73 images. Since we sample among the 200 classes we are slightly weighting towards classes with more readings. The above computation also ignores the possibility of the hash function in the digital locker (HMAC-SHA256) colliding. Removing the cryptographic component allows for substantially faster computation across parameters. The timing for the cryptographic implementation is presented in the Introduction. There was no noticeable deviation in TAR when using the cryptographic implementation.

6 The Feature Extractor

We now describe the feature extractor used in this work building on a recent feature extractor called ThirdEye [AF19]. Our changes are designed to provide better features for the sample-then-lock fuzzy extractor.

6.1 Recent Biometric CNNs

ThirdEye [AF19] is an open-source feature extractor for irises built from the ImageNet CNN. ThirdEye has two rounds of training and starts from a pre-trained model. The high-level approach begins with the ImageNet pre-trained model for a ResNet-50 architecture.

In the first stage of training, the final output layer uses a softmax to classify the input iris according to its class in the dataset. We call this stage **Cross-Entropy** as it is designed to minimize the entropy of the confusion matrix, outputting a maximally accurate model for the training set. ThirdEye then replaces the classification component of the network with a new set of layers with random initial values that serve to change the output to a feature vector with 1024 bits.

The second stage of training trains the last 20 layers of the model (including the ending weights of the classification network). For this second stage, the network is presented with triples of inputs. The network output features on these triplets are used to compute the loss function.[8] That is, a triple is created for each class, where one sample is declared as an anchor, denoted x_a and another from the same class is chosen because it has high distance from the anchor sample, denoted x_p (for positive). Another sample (from another class) is chosen from the batch from a different class that minimizes the distance between x_a and this sample, this sample is called x_n (for negative). The Triplet loss function is calculated as:

$$\texttt{TL} := \sum_i \texttt{TL}_i = \sum_i \left(m + L_2(x_{a,i}, x_{p,i}) - L_2(x_{a,i}, x_{n,i})\right) \tag{1}$$

In the above, m is a hyper-parameter indicating the desired gap between the distances of the same class and different classes, known as margin. A triplet is considered hard when $m + L_2(x_{a,i}, x_{p,i}) \geq L_2(x_{a,i}, x_{n,i})$ making the overall loss positive. This system takes the soft-margin of the above loss, described by Hermans et al. [HBL17]. After this transform an explicit definition of m is not required. The triplet loss function is calculated and back-propagated to the last 20 layers of the network.

Instead of optimizing the L_2 distance, Liu et al. [LWY+17] consider the cosine between samples of the same and different classes. Their loss function is minimized when all templates from different classes have an angle of 90° resulting in a cosine of 0 and all templates from the same class have an angle between them of 0° resulting in a cosine of 1. In each iteration of training, features are normalized so cosine can be computed using an inner product. A softmax is computed over the **angular margin** between templates of the same and different classes [LWY+17, Equation 7].

Awareness of discrete transform We note that ThirdEye and many prior feature extractors convert each real-valued feature to a bit by recording a 1 if the feature ≥ 0 and recording a 0 if the feature is negative. Typically, and is the case with ThirdEye, this results in the mean of the unlike (interclass) fractional Hamming distance being centered around 0.5. This can be seen in Figure 2b. An recent example of designing a feature extractor that is

[8] To make this stage of training more efficient, the training dataset is batched and the hardest triples are computed. These are triples with the highest current loss.

"crypto-aware" is that of Tarjela et al. [TVN20]: $L(x_i) = -1/D \sum_{i=1}^{N} \|x_i\|^2$ where x_i is the feature vector activated by the Tanh activation which has an output space of $[-1, 1]$. This forces the network to push each individual feature in the feature vector to either -1 or 1 providing better binary feature vectors. We tested adding this loss in; it universally resulted in worse performance and we do not report on results.

We trained the ThirdEye architecture using just the **Cross-Entropy** loss and fine tuned it using L_2 **triplets** and the **angular margin**. Histograms are shown in Figure 2a, 2b, and 2c. For histograms, all left eye images were used for training while 10 randomly chosen images were taken for testing. Our summary statistics of like μ, estimated entropy and $\texttt{FE}_{\texttt{Qual}}$ are shown in Table 1. Both L_2 triplets and angular margin are effective at (in comparison to just training with cross-entropy) reducing the overlap between the Like and Unlike Histograms and reducing the variance of the Like Histogram. Together this means one can set an acceptance distance t with a better TAR/FAR tradeoff. This is beneficial for a fuzzy extractor as one has to set a correction distance t and this distance must be smaller than the lowest observed Unlike comparison to yield any security.

In addition, angular margin substantially increases the estimated entropy of the features. This is visible in decreased variance in the Unlike distribution in Figure 2c. However, this increase comes at a cost, the like μ error rate increases from .19 for cross-entropy to .24.

6.2 Maximizing $\texttt{FE}_{\texttt{Qual}}$

Our design uses the ThirdEye architecture. Our design combines the ideas of L_2 margin maximization from triplet loss with angle minimization from the angular margin. `FiveEyes` retains the cross-entropy loss for the first stage of training. In the second stage of training, when 20 randomly initialized layers are added, we compute a modified triplet loss that includes the inner product between the anchor and negative examples:

$$TL_{\textbf{Final}} = \sum_i \begin{pmatrix} \overset{m}{} \\ +c_1 * L_2(x_{a,i}, x_{p,i}) \\ -L_2(x_{a,i}, x_{n,i}) \\ +c_2|\mu_{IP}| \end{pmatrix}.$$

In the above m represents the margin as before and is set to .8. c_0 represents additional weighting on positive examples and is set to 1.1. c_2 represents the weighting between the L_2 loss and the inner product and is set to 2.

Here μ_{IP} represents the average inner product between all pairs of points of different irises in the testing set. Unlike the angular margin we do not normalize feature vectors. The idea behind the approach is that angular margin forced the mean of comparisons between templates of different classes to be centered at .5 since vectors were normalized and minimized the cosine. With the new loss definition, $TL_{\textbf{Final}}$ can be reduced by either reducing the gap between distances or decreasing the mean inner-product. The inner-product can be reduced by either: 1) increasing the angle between the vectors, or 2) decreasing their norm. Decreasing the μ_{IP} by just reducing norm is likely to decrease the L_2 gap so these two objectives are competing. Using the hyperparameters to vary between the triplet and inner-product losses we can move the mean of unlike comparisons away from .5 but decrease the variance, yielding a larger overall entropy estimate. This is shown in Table 1 where `FiveEyes` has the largest estimated entropy and largest $\texttt{FE}_{\texttt{Qual}}$. The histogram in Figure 2d has a similar unlike variance to angular margin (Figure 2c) but with a smaller like μ.

Looking ahead to Section 9, the `FiveEyes` feature extractor with uniform sampling with $\beta = 250k$ subsets of size 60 yields a TAR of 42% at roughly 43 bits of security. This is only the minimum of 10 min-entropies, so should not be considered a final security estimate, see Section 9.2. The feature extractor alone yields a roughly 10 bit security increase over Simhadri et al. [SSF19] using a quarter of the samples. We defer discussion of additional results until after introducing ζ-sampling and what $\texttt{stats}_W$ are computed.

Input Augmentations The pipeline is further optimized by augmenting at train time. The train time augmentations include random use of image sharpening, rotations of $30°$,$-30°$ and a flip along the horizontal axis.

Feature Extractor	Like μ	Est. Entropy	$\texttt{FE}_{\texttt{Qual}}$
Cross entropy	.19	56	-662
L_2 hard triplets	.21	66	-693
Angular Margin	.24	132	-682
`FiveEyes`	.20	185	-627

Table 1: Illustration of Improvement between Error Rate and Estimated Entropy. As a reminder the value of $\texttt{FE}_{\texttt{Qual}} := \text{Ent} - n * h_2(\mu)$ is a lower bound on fuzzy min-entropy (for the mean error rate). This indicates how close the source is to extractable using standard coding-theoretic fuzzy extractors without relying on additional structure [FRS20].

6.3 Resulting features and $\texttt{stats}_W$

The feature extractor generates the feature vectors of length 1024. Our $\texttt{stats}_W$ consists of 2048 real values from $[0, 1]$. From here on we use w to refer to the output of the feature extractor. For each feature i we compute:

$$p_{\texttt{same},i} := \Pr[w_i \neq w'_i | w, w' \text{ readings same biometric}]$$
$$p_{\texttt{diff},i} := \Pr[w_i \neq w'_i | w, w' \text{ readings different biometrics}]$$

These probabilities are computed across the entire $\texttt{stats}_W$ set. There were also 4 positions in the 1024 length vector where the feature is a constant 0 for the entire set of $\texttt{stats}_W$. These positions are excluded from our subset selection algorithms and the uniform subset selection that we use as a comparison point representing prior work. The mean of the blue distribution in Figure 2d is $\mathbb{E}_i\, p_{\texttt{same},i}$ while the mean of the red distribution is $\mathbb{E}_i\, p_{\texttt{diff},i}$. We show the histogram of $p_{\texttt{same},i}$ and $p_{\texttt{diff},i}$ in Figure 3a. Figure 3b shows a histogram of $p_{\texttt{same},i} - p_{\texttt{diff},i}$. Based on these two histograms we make a few observations.

Features provide very different quality in terms of the gap $p_{\texttt{same},i} - p_{\texttt{diff},i}$. Many prior works have selected features of Gabor transform based feature extractors [HBF08, RUW11, ZLN12]. It appears such techniques are still necessary for modern deep-learning architectures. Simhadri et al. [SSF19, Table 3] showed that the correctness of sample-then-lock begins to degrade when one restricts to fewer than 1024 bits. Thus, we do not exclude any features aside from the 4 constant features.

Computation of $\texttt{stats}_W$ on data that is not used for testing is crucial. We computed the same histograms for the training data. For that data almost all of the features had a gap greater than 0.4. The network was able to achieve a $p_{\text{same},i} \leq .1$ for almost all features and $p_{\texttt{diff},i} \approx .5$ for almost all features. These error rates did not indicate any variation in quality of features and also did not predict error rates seen on the test set. The histograms are shown in Figure 4.

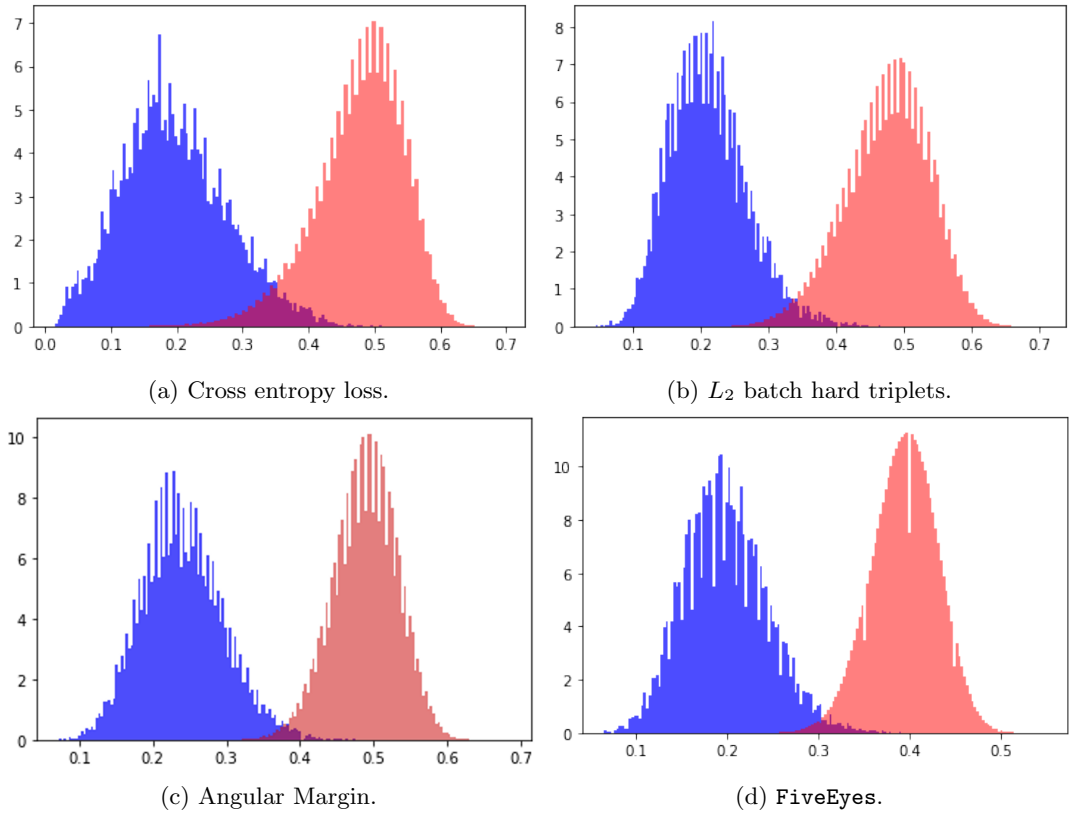
7 The Fuzzy Extractor

The high-level idea of sample-then-lock [CFP+16] is to encrypt the same key multiple times using different subsets of w. We generate subsets globally in the Setup algorithm and our method of sampling subsets is our main technical contribution on fuzzy extractors. Sample-then-lock uses digital lockers [CD08]. Digital lockers are computationally secure symmetric encryption schemes that retain security when the key comes from a distribution with some (unspecified) amount of entropy as long as that entropy is super logarithmic in the security parameter that bounds the running time of the adversary [CKVW10]. Notationally, it is an obfuscation of the function $I_{\mathsf{val},\mathsf{key}}(\mathsf{val}') = \mathsf{key}$ if and only if $\mathsf{val}' = \mathsf{val}$. Notationally, we say that $\mathsf{unlock}_{\mathsf{val},\mathsf{key}} \leftarrow \mathsf{lock}(\mathsf{val}, \mathsf{key})$. It should be the case that $\mathsf{unlock}_{\mathsf{val},\mathsf{key}}$ is functionally equivalent to $I_{\mathsf{val},\mathsf{key}}$.

Definition 3. *The algorithm* lock *with security parameter* λ *is an* β*-composable digital locker with error* γ *if the following hold:*

Correctness *For any triple* $\mathsf{key}, \mathsf{val}, \mathsf{val}' \neq \mathsf{val}$*,*

$$\Pr[\mathsf{unlock}(\mathsf{val}) = \mathsf{key} | \mathsf{unlock} \leftarrow \mathsf{lock}(\mathsf{val}, \mathsf{key})] \geq 1 - \gamma,$$
$$\Pr[\mathsf{unlock}(\mathsf{val}') = \perp | \mathsf{unlock} \leftarrow \mathsf{lock}(\mathsf{val}, \mathsf{key})] \geq 1 - \gamma.$$



(a) Cross entropy loss. (b) L_2 batch hard triplets.

(c) Angular Margin. (d) `FiveEyes`.

Figure 2: Distance comparisons for loss functions used in developing `FiveEyes`. Comparisons between readings of the same biometric are in blue. Comparisons between readings of different biometrics are in red. The x-axis differs. This figure combines data from $\texttt{stats}_W$ and testing

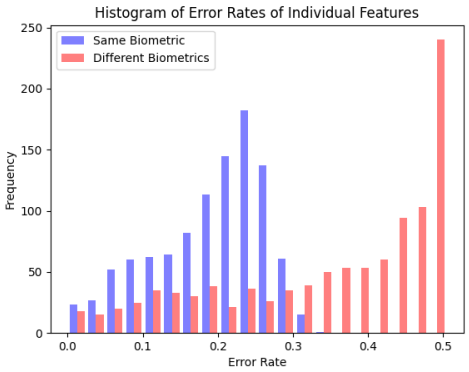
Security *For each PPT A, positive polynomial p, there exists a (possibly inefficient) simulator S and a polynomial $q(\lambda)$ such that for any sufficiently large s, any polynomially-long sequence of values $(\mathsf{val}_i, \mathsf{key}_i)$ for $i = 1, \ldots, \beta$, and any auxiliary input $z \in \{0,1\}^*$,*

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

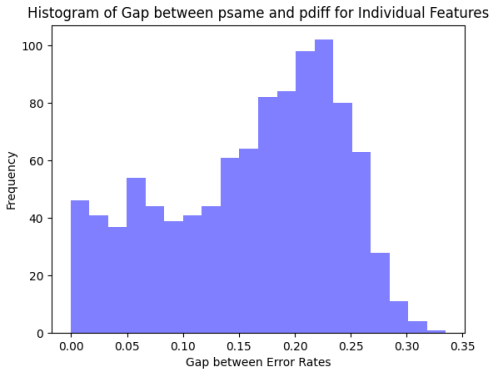
The above definition is virtual grey-box obfuscation (because the simulator is allowed to run in unbounded time). It implies distributional indistinguishability which says that all distributions with $\mathrm{H}_\infty(\mathsf{val}) \geq \omega(\log \lambda)$ are indistinguishable. In fact the definitions are equvalent if there are a constant number of digital lockers or the same val is used [Can97, BC10, Var10, FF20, ACF+22]

Unfortunately, the security definition of digital lockers is "inherently" asymptotic. A different simulator is allowed for each distance bound $p(s)$ making it difficult to argue what quality key is provided with respect to a particular adversary.

Let `HMAC` be HMAC-SHA256. Our construction assumes that `HMAC` can be used to construct digital lockers. The "locking" algorithm outputs the pair $\mathsf{nonce}, \texttt{HMAC}(\mathsf{nonce}, w) \oplus (0^{128}||\mathsf{Key})$, where nonce is a nonce, $||$ denotes concatenation, 0^{128} is the all zeros string of length 128. Unlocking proceeds by recomputing the hash and checking for a prefix of 0^{128}. If this prefix is found then the suffix Key' is output. Digital lockers can be constructed from variants of the Diffie-Hellman assumption [CD08, Zha19] and Learning with Errors [WZ17, GKW17]. The HMAC construction used in this work construction was proposed in [BR93] and shown to be secure in the random oracle



(a) Histogram of $p_{\mathtt{same},i}$ and $p_{\mathtt{diff},i}$ across i.



(b) Value of $p_{\mathtt{same},i} - p_{\mathtt{diff},i}$ across i.

Figure 3: Different features have different error rates of $p_{\mathtt{same},i}$ and $p_{\mathtt{diff},i}$ and different gaps between these values.

model by Lynn, Prabhakaran, and Sahai [LPS04, Section 4]. It is plausible that in the standard model (without random oracles) hash functions provide the necessary security [CD08, Section 3.2], [Dak09, Section 8.2.3].

7.1 Sample-then-lock Overview

We let β denote the number of subsets and assume that $\mathcal{I}_1, ..., \mathcal{I}_\beta$ is provided as input to Gen and Rep as $\mathtt{advise}_W$. Pseudocode for Gen and Rep is in Figure 5.

The parameters k (size of each subset) and β (number of subsets) represent a tradeoff between correctness and security. Canetti et al. [CFP+16, Section 4] note that rather than using independent subsets they could be selected using a sampler [Gol11]. Simhadri et al. [SSF19] noted that each subset on its own needs to be random. For a particular output of Setup define the minimum of the subset min-entropies:

$$\nu := \left(\min_{1 \le i \le \beta} \{ \mathrm{H}_\infty(W_{\mathcal{I}_i} | \mathsf{Setup}(\mathtt{stats}_W) = \mathcal{I}_1, ..., \mathcal{I}_\beta \} \right)$$

We assume that the security level provided is the minimum of the min-entropies used as input. We state this formally below:

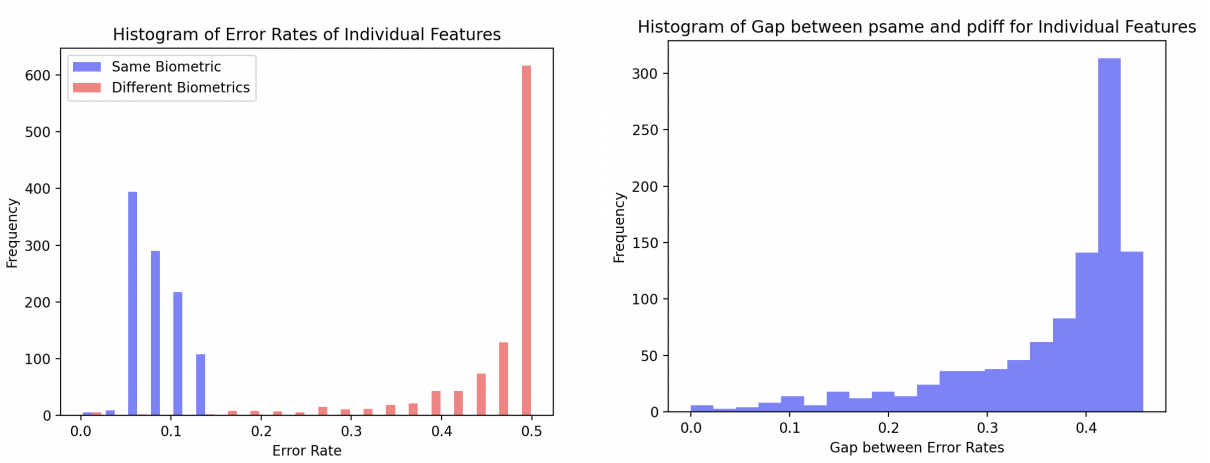
Assumption 2. *Let $\mathsf{Val}_1, ..., \mathsf{Val}_\beta, Z$ be sampled from (correlated) distributions and let U_κ, U'_κ be uniformly chosen. Let $\nu := (\min_{1 \le i \le \beta} \{ \mathrm{H}_\infty(\mathsf{Val}_i | Z) \})$. Then for all PPT A the following holds:*

$$\left| \begin{array}{r} \Pr\left[A\left(Z, \{\mathsf{lock}\left(\mathsf{Val}_i, U_\kappa\right)\}_{i=1}^{\beta}, U_\kappa \right) = 1 \right] \\ - \Pr\left[A\left(Z, \{\mathsf{lock}\left(\mathsf{Val}_i, U_\kappa\right)\}_{i=1}^{\beta}, U'_\kappa \right) = 1 \right] \end{array} \right| \le 2^{-\Theta(\nu)}. \tag{2}$$

7.1.1 Bug and fix of proof of [CFP+21, Theorem 1]

As mentioned above, Definition 3 is an inherently asymptotic definition. This is due to a different simulator being used for each desired inverse polynomial quality. For our discussion throughout this paper, we ignore the difference between an adversary with the real obfuscation and the simulator with an oracle. That is, why the above is an assumption rather than a Lemma.

Canetti et al. [CFP+21, Theorem 1] bound adversary success when given an oracle to the digital locker functionality. Specifically, they show that when Val_i are all chosen from the same distribution specified by Z_i respectively, it suffices for $\tilde{\mathrm{H}}_\infty(\mathsf{Val}_i | Z) = \omega(\log n)$. While their theorem statement is correct, their proof has a bug and does



(a) Histogram of $p_{\texttt{same},i}$ and $p_{\texttt{diff},i}$ across i.

(b) Value of $p_{\texttt{same},i} - p_{\texttt{diff},i}$ across i.

Figure 4: The danger of computing $\texttt{stats}_W$ from the training set. Different features have different error rates of $p_{\texttt{same},i}$ and $p_{\texttt{diff},i}$ and different gaps between these values when $\texttt{stats}_W$ is computed from training set. The feature extractor optimizes all output features nearly equally on the training set. This does not indicate the $p_{\texttt{same},i}, p_{\texttt{diff},i}$ on future data.

$\mathsf{Gen}(w, \texttt{stats}_W = \mathcal{I}_1, ..., \mathcal{I}_\beta)$:

1. Sample random 128 bit Key.
2. For $i = 1, ..., \beta$:
   (i) Choose 512 bit hash key h_i.
   (ii) Set $c_i = \texttt{HMAC}(h_i, w_{\mathcal{I}_i})$.
   (iii) Set $p_i = (0^{128} || \mathsf{Key}) \oplus c_i$.
3. Output (Key, p_i, h_i).

$\mathsf{Rep}(w', p_1, ..., p_\beta, h_1, ..., h_\beta, \texttt{stats}_W = \mathcal{I}_1, ..., \mathcal{I}_\beta)$:

1. For $i = 1, ..., \beta$:
   (i) Set $c_i = \texttt{HMAC}(h_i, w'_{\mathcal{I}_i})$.
   (ii) If $(c_i \oplus p_i)_{1..128} = 0^{128}$ then output $(c_i \oplus p_i)_{129..256}$.
2. Output $\perp$.

Figure 5: Adaption of sample-then-lock to use global subsets from $\texttt{advise}_W = \mathcal{I}_1, ..., \mathcal{I}_\beta$.

not account for variation is the min-entropy of $\mathsf{Val}_i|Z$. Their proof assumes that each of these distributions has the same entropy as the average min-entropy. In particular, [CFP+21, Lemma 2] is incorrect as stated. However, Theorem 1 is still correct as one can bound the entropy drop by a fraction of ζ with overwhelming probability (see Lemma 2 in the Appendix). However, it does impact the actual hardness of guessing a value $\mathsf{Val}_i|Z_i$. This is why we measure our security by the minimum of entropies in contrast to Simhadri et al. [SSF19] who consider the average min-entropy of $\mathsf{Val}_i|Z_i$. We produce a corrected proof of the main lemma here for completeness.

Lemma 1. *Let $\mathsf{Val}_1, ..., \mathsf{Val}_\beta, Z$ be correlated random variables and let U_κ, U'_κ be uniformly random values. For some outcome z let $\nu := (\min_{1 \le i \le \beta}\{\mathrm{H}_\infty(\mathsf{Val}_i|Z = z)\})$. Then for any S given at most q queries it is true that*

$$\left| \begin{array}{r} \Pr\left[S^{\{I_{\mathsf{val}_i, U_\kappa}(\cdot)\}_{i=1}^{\beta}}\left(z, \{|\mathsf{val}_i|\}_{i=1}^{\beta}, \kappa, U_\kappa\right) = 1\right] \\ -\Pr\left[S^{\{I_{\mathsf{val}_i, U_\kappa}(\cdot)\}_{i=1}^{\beta}}\left(z, \{|\mathsf{val}_i|\}_{i=1}^{\beta}, \kappa, U'_\kappa\right) = 1\right] \end{array} \right| \le \frac{q(q+1)}{2^\nu}. \tag{3}$$

In addition, let $\nu_{avg} = \left(\min_{1\le i\le\beta}\{\tilde{\mathrm{H}}_\infty(\mathsf{Val}_i|Z)\}\right)$ *then*

$$\left|\begin{array}{r}\Pr\left[S^{\{I_{\mathsf{val}_i,U_\kappa}(\cdot)\}_{i=1}^{\beta}}\left(Z,\{|\mathsf{val}_i|\}_{i=1}^{\beta},\kappa,U_\kappa\right)=1\right]\\ -\Pr\left[S^{\{I_{\mathsf{val}_i,U_\kappa}(\cdot)\}_{i=1}^{\beta}}\left(Z,\{|\mathsf{val}_i|\}_{i=1}^{\beta},\kappa,U'_\kappa\right)=1\right]\end{array}\right| \le \frac{q(q+1)+1}{2^{\nu_{avg}/2}}. \tag{4}$$

Proof of Lemma 1. We restate a Lemma on the amount average min-entropy decreases across choices of b [DORS08, Lemma 2.2b]:

Lemma 2. *Let* A, B *be random variables. For any* $\delta > 0$*,*

$$\Pr_b[\mathrm{H}_\infty(A|B=b) \ge \tilde{\mathrm{H}}_\infty(A|B) - \log(1/\delta)] \ge 1-\delta.$$

In particular,

$$\Pr_b\left[\mathrm{H}_\infty(A|B=b) \ge \frac{1}{2}\tilde{\mathrm{H}}_\infty(A|B)\right] \ge 1 - 2^{-\frac{1}{2}\tilde{\mathrm{H}}_\infty(A|B)}.$$

Equation 4 follows from Equation 3 by Application of Lemma 2 with $\delta = 2^{-\zeta/2}$. We focus on Equation 3. The proof simply corrects a bug in [CFP+21, Lemma 2]. A proof is reproduced below for completeness.

Fix any $u, u' \in \{0,1\}^\kappa$ (the lemma will follow by averaging over all u). The only information about whether the values u, u' can obtained by S through the query responses. First, modify S slightly to quit immediately if it gets a response not equal to $\perp$. Such S is equally successful at distinguishing between u, u'. There are $q+1$ possible values for the view of S on a given input (q of those views consist of some number of $\perp$ responses followed by the first non-$\perp$ response, and one view has all q responses equal to $\perp$). By [DORS08, Lemma 2.2b], $\tilde{\mathrm{H}}_\infty(\mathsf{Val}_i|View(S), Z=z) \ge \tilde{\mathrm{H}}_\infty(\mathsf{Val}_j|Z=z) - \log(q+1) \ge \nu - \log(q+1)$. Therefore, at each query, the probability that S gets a non-$\perp$ answer (equivalently, guesses some Val_i) is at most $(q+1)2^{-\nu}$ across q queries of S, the probability is at most $q(q+1)/2^\nu$. □

8 ζ-Subset Selection

We now turn to our second technical contribution of ζ-sampling. Our goal is to select subsets for a sample-then-lock better than the uniform subset selection. In this section, we propose an algorithm that uses auxiliary information (the confidence information from the feature extractor) to select better subsets.

The heart of ζ-sampling is to use $p_{\mathtt{same},i}$ to select subsets that are least likely to introduce an error (between two readings of the same iris). To ensure that bits are not selected that are stable across all readings, we weight selection by the binary entropy of $p_{\mathtt{diff},i}$ to increase the entropy of the included bits. Both of versions of the algorithms use a sampling characteristic parameter ζ and are shown in Figure 6.

An increase in ζ causes a sharper curve on the probability that a bit is selected based on its $p_{\mathtt{same},i}$. The idea is that low values of ζ pick close to uniformly from the indices (zero being a uniform selection) and at high values better indices are selected with much higher probability. For both algorithms we also ensure that no subset has duplicate indices, but we do <u>not</u> enforce that no two subsets are the same. We: 1) analyze the number of steps required for ζ-sampling to reach a target correctness, 2) show that ζ-sampling has a positive partial derivative with respect to $\zeta = 0$ (which is uniform selection), and 3) give a mechanism for estimating the optimal ζ for a given $p_{\mathtt{same}}$. Our analysis focuses on the setting when $1_{\mathtt{entropy}} = 0$ but we give intuition for the objective of ζ-sampling when $1_{\mathtt{entropy}} = 1$.

$\zeta - \texttt{Sample}(p_{\texttt{same}}, p_{\texttt{diff}}, 1_{\texttt{entropy}})$:

1. For $i = 1$ to β:
   (a) For $j = 1$ to n:
   If $1_{\texttt{entropy}} == 0$: Set $w_j = (1 - p_{\texttt{same},j})^{\zeta}$.
   Else, set $w_j = (1 - p_{\texttt{same},j})^{\frac{\zeta}{h_2(p_{\texttt{diff},j})}}$.
   (b) Let $\mathcal{D}$ denote the probability distribution on $\{1, \ldots, n\}$ proportional $w_i / \sum_{\ell=1}^{n} w_\ell$.
   (c) Independently draw k items, $q_1, \ldots, q_k$, from $\mathcal{D}$. While $q_1, ..., q_k$ are not distinct, repeat.
   (d) Output $\vec{q_i} = q_1, ..., q_k$.
2. Set $\vec{q}_1, ..., \vec{q}_\beta$.

Figure 6: ζ-Sampling.

8.1 The Abstract Problem Description

To provide a theoretical justification and analysis of the proposed family of subset selection algorithms above, we formulate an idealized version of the problem that posits a family of independent "features," each of which can be correctly predicted with known probability $p_i := 1 - p_{\texttt{same},i}$. We then analyze the success probability of the algorithm that selects features with probability proportional to p_i^{ζ}, and succeeds when the features so selected are distinct and, furthermore, can be simultaneously predicted. Throughout our formal analysis we assume that all features are independent, which is usually not true in practice. Our actual implementation of the ζ-norm algorithms in Sec. 5.2 additionally weights selection by $p_{\texttt{diff},i}$. This heuristic algorithm is the one we use in experiments in Section 9.

The abstract problem is described by a sequence $\mathbf{p} = (p_1, \ldots, p_n)$, with each p_i in the range $[1/2, 1]$, and an integer k. In the context of $\mathbf{p}$ and k, we are interested in designing algorithms $\mathbf{A}$:

1. Let $X_1, \ldots, X_m$ to be a family of independent random variables, each taking values in the set $\{0, 1\}$, with the property that $\Pr[X_i = 1] = p_i$.
2. The algorithm $\mathbf{A}$ (with knowledge of $\mathbf{p}$ and k but without knowledge of the X_i), selects a subset of $\{1, \ldots, n\}$ of size exactly k. If $X_i = 1$ for each $i \in Q$, the game ends. Otherwise, this step is repeated.

$\mathbf{A}$'s goal is to adopt a strategy that ends the game as quickly as possible (that is, after the minimum number of queries) and measure the success of a strategy using tail bounds of the form

$$\Pr[\mathbf{A} \text{ requires more than } T \text{ steps to win}] \leq \epsilon_T \,. \tag{5}$$

We note that a deterministic strategy for $\mathbf{A}$ is completely described by a sequence $Q_1, Q_2, \ldots$ of queries. In our case, we will be studying randomized strategies for this game, which place a probability distribution on such sequences of queries; in this case, the probability space over which this probability is taken is given by both the X_i and the selection of the random strategy. We say that a strategy is (T, ϵ_T)-bounded if is meets the criteria (5) above.

8.1.1 The ζ-norm sampling algorithms

We propose and analyze an algorithm that we call *ζ-norm sampling* and write $\mathbf{A}^{\zeta}$. In the context of $\mathbf{p} = (p_1, \ldots, p_n)$ and k, each query follows the same randomized mechanism:

- Let $\mathcal{D}$ denote the probability distribution on $\{1, \ldots, n\}$ that is proportional to the ζ-norm of $\mathbf{p}$, which is to say that $\mathcal{D}(i) = p_i^{\zeta} / (\sum_i p_i^{\zeta})$.
- Independently draw k items, $q_1, \ldots, q_k$, from the distribution $\mathcal{D}$.

- If k distinct items were not drawn, abandon the query and restart. Otherwise, issue the query $Q = \{q_1, \ldots, q_k\}$.

Theorem 1. $\mathbf{A}^\zeta$ *is* $(8/\mathbb{E}(M)^k, 9k^2\Gamma)$*-bounded, meaning*

$$\Pr[A^\zeta \textit{ takes more than } 8/\mathbb{E}(M)^k \textit{ steps to win}] \leq 9k^2\Gamma$$

where

$$\Gamma = \sum_i p_i^{2\zeta+1} / (\sum_i p_i^{\zeta+1})^2 .$$

The general idea of the proof is to measure the probabilities of sampling indices that match and then to measure the probability that those matching indices are not unique. Using tail bounds we then can bound the probability that we end up with a selection that wins the simplified game.

Proof. We proceed here by treating ζ as a free parameter and discuss the choice of ζ afterwards. As the ζ-norm sampling algorithm A^ζ is randomized, the behavior of the algorithm depends on both the particular values taken by the random variables X_i and the randomly sampled points. Our analysis treats these two sources of randomness separately.

We call an index whose corresponding X_i is equal to 1 "good" and other indices "bad." We then focus on two quantities of interest, determined by the random variables X_i. For a fixed set of values $x_1, \ldots, x_n$ taken by the random variables X_i, consider a single selection q of A^ζ (according to $\mathcal{D}$); then we define

$$M(x_1, \ldots, x_n) = \Pr[q \text{ is good} \mid \forall i, X_i = x_i] = \frac{\sum_i x_i p_i^\zeta}{\sum_i p_i^\zeta} .$$

This reflects the probability that an individual item chosen by A^ζ is good. Along these same lines, consider a pair of queries q, q' generated by A^ζ (with each drawn independently from $\mathcal{D}$) and define

$$\begin{aligned} C(x_1, \ldots, x_n) &= \Pr[q, q' \text{ are good and } q = q' \mid \forall i, X_i = x_i] \\ &= \frac{\sum_i x_i p_i^{2\zeta}}{(\sum_i p_i^\zeta)^2} . \end{aligned}$$

This reflects the probability that a good item drawn by two particular samples is the same. Continuing to work with a particular setting of the variables X_i (to x_i), we can calculate the probability that a particular query generated by A^ζ is not abandoned and, furthermore, wins the game, which is to say that the query consists of k distinct, good items:

$$\begin{aligned} S(x_1, \ldots, x_n) &= \Pr[k \text{ distinct, good items are selected by } A^\zeta] \\ &= \Pr[\text{all selected items are good}] \\ &\quad - \Pr[\text{selected items are good, repeat}] \\ &\geq M(x_1, \ldots, x_n)^k - \binom{k}{2} \cdot C(x_1, \ldots, x_n) \cdot M(x_1, \ldots, x_n)^{k-2} \\ &\geq M(x_1, \ldots, x_n)^k - k^2 \cdot C(x_1, \ldots, x_n) \cdot M(x_1, \ldots, x_n)^{k-2} . \end{aligned} \tag{6}$$

Finally, since draws of A^ζ are independent, observe that for this particular assignment of the X_i the running time of A^ζ is no more than $1/S(x_1, \ldots, x_n)$, where S is the quantitfy in 6. With this observation, the remainder of the argument will focus on the values taken by M and C under selection of the X_i. In particular, we may treat $M(X_1, \ldots, X_n)$ and $C(X_1, \ldots, X_n)$ as random variables (determined entirely by the X_i) which, for brevity, we simply write as M and C. These determine a bound on $S = S(X_1, \ldots, X_n)$, as above, which is treated similarly.

Our strategy shall be to evaluate the expected values of M and C and establish tail bounds on these random variables that show that they are unlikely to deviate from their expectations in ways that degrade the inequality 6.

We conclude that with high probability in the random variables X_i, the resulting quantity 6 provides a satisfactory bound on the running time of the algorithm A^ζ.

We will first apply Chebyshev's inequality to control the difference between M and its expected value $\mathbb{E}(M)$. Throughout, we let $\mathbb{E}(Z)$ and $\mathrm{Var}(Z)$ denote the expectation and variance of the random variable Z.

We immediately compute: $\mathbb{E}(X_i) = p_i$, $\mathrm{Var}(X_i) = p_i(1-p_i)$, and

$$\mathbb{E}(M) = \frac{\sum_i p_i^\zeta \, \mathbb{E}(X_i)}{\sum_i p_i^\zeta} = \frac{\sum_i p_i^{\zeta+1}}{\sum_i p_i^\zeta}.$$

As the X_i are independent, we can compute $\mathrm{Var}(M)$ as follows.

$$\begin{aligned}\mathrm{Var}(M) &= \frac{\mathrm{Var}(\sum_i X_i p_i^\zeta)}{(\sum_i p_i^\zeta)^2} = \frac{\sum_i p_i^{2\zeta}\mathrm{Var}(X_i)}{(\sum_i p_i^\zeta)^2} \\ &= \frac{\sum_i p_i^{2\zeta+1}(1-p_i)}{(\sum_i p_i^\zeta)^2} \le \frac{\sum_i p_i^{2\zeta+1}}{(\sum_i p_i^\zeta)^2}.\end{aligned}$$

According to Chebyshev's inequality, we have

$$\begin{aligned}\Pr\left[M \le \left(1-\frac{1}{k}\right)\mathbb{E}(M)\right] \le \frac{\mathrm{Var}(M)}{(\frac{1}{k}\,\mathbb{E}(M))^2} \le \\ \frac{k^2 \frac{\sum_i p_i^{2\zeta+1}}{(\sum_i p_i^\zeta)^2}}{\left(\frac{\sum_i p_i^{\zeta+1}}{\sum_i p_i^\zeta}\right)^2} = k^2 \frac{\sum_i p_i^{2\zeta+1}}{(\sum_i p_i^{\zeta+1})^2}.\end{aligned} \tag{7}$$

We now turn our attention to C. We compute

$$\mathbb{E}(C) = \frac{\sum_i p_i^{2\zeta}\,\mathbb{E}(X_i)}{(\sum_i p_i^\zeta)^2} = \frac{\sum_i p_i^{2\zeta+1}}{(\sum_i p_i^\zeta)^2},$$

and, therefore, by Markov's inequality

$$\begin{aligned}\Pr\left[C \ge \frac{\mathbb{E}(M)^2}{8k^2}\right] \le \frac{\mathbb{E}(C)}{\frac{\mathbb{E}(M)^2}{8k^2}} \le \\ \frac{8k^2 \frac{\sum_i p_i^{2\zeta+1}}{(\sum_i p_i^\zeta)^2}}{\left(\frac{\sum_i p_i^{\zeta+1}}{\sum_i p_i^\zeta}\right)^2} = 8k^2 \frac{\sum_i p_i^{2\zeta+1}}{(\sum_i p_i^{\zeta+1})^2}.\end{aligned} \tag{8}$$

Noting the similarity between the right-hand sides of 7 and 8, we define

$$\Gamma = \sum_i p_i^{2\zeta+1} / (\sum_i p_i^{\zeta+1})^2 \,.$$

The two inequalities (7) and (8) can then be written as

- $\Pr[M \le (1-\frac{1}{k})\,\mathbb{E}(M)] \le k^2\Gamma$.
- $\Pr[C \ge \frac{\mathbb{E}(M)^2}{8k^2}] \le 8k^2\Gamma$.

Combining these, we note

$$\begin{aligned}&\Pr\left[M > (1-\frac{1}{k})\,\mathbb{E}(M) \cap C < \frac{\mathbb{E}(M)^2}{8k^2}\right] \\ &= 1 - \Pr\left[M \le (1-\frac{1}{k})\,\mathbb{E}(M) \cup C \ge \frac{\mathbb{E}(M)^2}{8k^2}\right] \\ &\ge 1 - (k^2\Gamma + 8k^2\Gamma) = 1 - 9k^2\Gamma.\end{aligned} \tag{9}$$

Under the assumption that $M > (1-1/k)\,\mathbb{E}(M)$ and $C < \mathbb{E}(M)^2/8k^2$, we can bound the probability in (6) as follows:

$$\begin{aligned}
&\Pr[k \text{ distinct, good items are selected}]] \geq= M^{k-2}(M^2 - k^2 C) \\
&> \left(1-\frac{1}{k}\right)^{k-2} \mathbb{E}(M)^{k-2} \left(\left(1-\frac{1}{k}\right)^{2} \mathbb{E}(M)^2 - \frac{\mathbb{E}(M)^2}{8}\right) \\
&= \left(1-\frac{1}{k}\right)^{k} \mathbb{E}(M)^{k} - \left(1-\frac{1}{k}\right)^{k-2} \frac{\mathbb{E}(M)^k}{8} \\
&\geq \frac{\left(1-\frac{1}{k}\right)^{k}}{2} \mathbb{E}(M)^k \geq \frac{\mathbb{E}(M)^k}{8}.
\end{aligned} \tag{10}$$

In the above expression, since $k \geq 2$, the third inequality holds because $(1-1/k)^2\,\mathbb{E}(M)^2 \geq \mathbb{E}(M)^2/4 > \mathbb{E}(M)^2/8$, and the second to last inequality holds because $1/(1-1/k)^2 \cdot 1/8 \leq 1/2$. The last one holds because $(1-1/k)^k \in [1/4, 1/e)$.

Thus, when $M > (1-1/k)\,\mathbb{E}(M)$ and $C < \mathbb{E}(M)^2/8k^2$, the number of expected queries by A^ζ is no more than $8/\,\mathbb{E}(M)^k$. In conclusion, the probability that the expected number of queries that A^ζ takes is more than $8/\,\mathbb{E}(M)^k$ is no more than $9k^2\Gamma$. □

8.1.2 Analyzing Entropy weighted ζ-sampling

The algorithm in Figure 6 with $1_{\texttt{entropy}} = 0$ will naturally tend to sample subsets $Q = \{q_1, \ldots, q_k\}$ of variables for which the expectations p_{q_i} are large. However, in our setting we wish to ensure that the resulting sampled subsets also have entropy. As an extreme example, if one had not excluded the constant features, they would always be included in the sample. We consider a heuristic sampling algorithm that maximizes prediction appropriately scaled by (Shannon) entropy.

In more detail, for a sequence of variables $X_{q_1}, \ldots, X_{q_k}$, the logarithm of the probability of correctly predicting this (independent) sequence is

$$\log \prod_i p_{q_i} = \sum_i \log p_{q_i};$$

the Shannon entropy of this collection is $\sum_i H(X_{q_i})$.

Recalling that our goal is to achieve a target entropy total $\texttt{e}$ while maximizing the probability of prediction, this calls for selecting bits that maximize the ratio

$$\frac{\log p_{q_i}}{h_2(p_{\texttt{diff},q_i})}.$$

This is equivalent to maximizing

$$2^{\frac{\log p_{q_i}}{h_2(p_{\texttt{diff},q_i})}} = p_{q_i}^{1/h_2(p_{\texttt{diff},q_i})} = (1-p_{\texttt{same},q_i})^{1/h_2(p_{\texttt{diff},q_i})},$$

which can be contrasted with the algorithms of the previous section. In particular, we study the family of algorithms, parameterized by $\zeta > 0$, that sample as above by assigning weight $w_i = p_i^{\zeta/h_2(p_{\texttt{diff},i})}$ to each index $i \in [1,n]$ and so sample with the probability: $w_i/\sum_i w_i$.

This is the weighting that we use in all experiments.

To summarize, adjusting ζ in this family of algorithms determines the relative weight given to bits with larger values of $p_i^{1/h_2(p_{\texttt{diff},i})}$. As our ultimate measure of success is given by the probability that at least one selected subset has no errors, optimizing choice of ζ must balance two competing phenomena: while increasing ζ will tend to select individual subsets that are less likely to induce errors, this also concentrates the distribution of selected bits, increases the likely overlap between pairs of sets selected in this way, and so increases the correlation of failure among the chosen sets.

8.2 Simple estimates of the optimal ζ

We note that

$$\frac{\partial \Gamma}{\partial \zeta} = \frac{2}{(\sum_i p_i^{\zeta+3})^3} \sum_{i<j} p_i^{\zeta+1} p_j^{\zeta+1} (p_i^\zeta - p_j^\zeta)(\ln p_i - \ln p_j) \geq 0$$

and

$$\frac{\partial \mathbb{E}(M)}{\partial \zeta} = \frac{1}{(\sum_i p_i^{\zeta})^2} \sum_{i<j} p_i^{\zeta} p_j^{\zeta} (p_i^\zeta - p_j^\zeta)(\ln p_i - \ln p_j) \geq 0\,.$$

Thus Γ and $\mathbb{E}(M)$ both grow monotonically with ζ (and $8/\mathbb{E}^k(M)$ and $1 - 9k^2\Gamma$ both decrease); this provides a direct trade-off between the guaranteed running time and the probability of the guarantee.

It's useful to identify some particular values of ζ that provide specific guarantees of interest. For example, consider the value $\zeta_{1/2}$ defined to be the maximum ζ for which $9k^2\Gamma \leq 1/2$, which is to say that the running time guarantee should apply with probability at least $1/2$ in the choice of the X_i. In light of the monotonicity comments above, this choice of ζ optimizes the running time bound $8/\mathbb{E}(M)^k$ under this constraint.

One difficulty with articulating such bounds is that the quantity Γ is somewhat difficult to directly interpret and optimize. To provide a collection of bounds that are easier to interpret, we note that

$$\Gamma = \frac{\sum_i p_i^{2\zeta+1}}{(\sum_i p_i^{\zeta+1})^2} \leq \frac{\max_i p_i^\zeta \cdot \sum_i p_i^{\zeta+1}}{(\sum_i p_i^{\zeta+1})^2} = \frac{\max_i p_i^\zeta}{(\sum_i p_i^{\zeta+1})}\,.$$

This leads to a simpler definition of an attractive choice of ζ: Specifically, define $\zeta^*_{1/2}$ to be the maximum value of ζ for which

$$\Gamma \leq \frac{\max_i p_i^\zeta}{(\sum_i p_i^{\zeta+1})} \leq \frac{1}{18k^2}\,.$$

Then the probability that the expected number of queries that ζ-Algorithm takes is no more than $8/\mathbb{E}^k(M)$ is no less than $1/2$.

9 Evaluation

The primary goal of our experimental setup is to evaluate the TAR versus entropy tradeoff of a full system using irises processed by the `FiveEyes` feature extractor (Section 6), and placed into a *sample-then-lock* fuzzy extractor with subsets sampled using ζ-sampling (Section 7). We now use DSet to denote the test dataset described in Section 5.1. We explore three main questions.

1. The impact of the new loss functions on the entropy vs. TAR tradeoff. This compares the five-eye feature extractor with the feature extractor using angular margin.
2. The impact of ζ-sampling and subset size k on the entropy vs. TAR tradeoff. For efficiency reasons these tests consider a small number of subsets so the minimum of entropies is inaccurate.
3. For the most promising parameters, we report on a detailed investigation into the minimum of entropies across all subsets that would be used in practice. We publish the analyzed subsets as part of this work. These subsets (along with the trained CNN) are the output of Setup.

9.1 Parameter Finding

This subsection provides an overview of our three main tests: 1) comparing the angular margin and `FiveEyes` feature extractors (the other two feature extractors exhibited insufficient entropy of 56 and 66 for security before sampling), 2) comparing TAR and entropy across ζ and subset sizes k, and 3) for the most promising combinations of ζ, k a detailed analysis of the full set of subsets.

	k	60	65	70	75	80	85	90	95
Angular	TAR	.18	.11	.06	.04	.02	.01	.01	.01
	Ent.	39	40	42	45	47	49	50	52
`FiveEyes`	TAR	.32	.22	.15	.10	.06	.04	.03	.02
	Entropy	40	45	44	48	50	55	56	60

Table 2: Comparison of the angular margin and `FiveEyes` feature extractors. $\beta = 10^5$ subsets with uniform sampling.

9.1.1 Comparing Angular Margin and `FiveEyes`

Our first test is to confirm the hypothesis that `FiveEyes` produces a superior TAR vs. entropy tradeoff than angular margin. For this test, we take a small number of subsets $\beta = 10^5$ for $k \in \{60, 65, 70, 75, 80, 85, 90, 95\}$. We then compute TAR for this family of 10^5 subsets. Then 10 subsets are picked to assess entropy. The minimum of the entropies is reported. Results are shown in Table 2. Across parameters `FiveEyes` outperforms angular margin with respect to both TAR and entropy. Based on these findings the rest of this section restricts to considering `FiveEyes`.

9.1.2 Estimating TAR vs. entropy Tradeoff across ζ, k

We now restrict to finding parameters of interest for the `FiveEyes` feature extractor. For this analysis we set the number of subsets to $\beta = 250$K. We perform the TAR vs. entropy test for the following settings

$$k \in \{65, 70, 75, 80, 85, 90, 95, 100\}$$
$$\zeta \in \{0, 5, 10, 15, 20, 25, 30, 35, 40\}.$$

For $k = 60$ we used $\zeta \in \{0, 0.5, 0.9, 5, 10, 15, 20, 30\}$ to understand high TAR behavior. For each selected parameter, 10 subsets are picked for entropy analysis. We output the minimum of the 10 values entropies. Results are shown in Figure 7a.

An increase in k yields a decrease in TAR and an increase in TAR for a fixed ζ. Surprisingly, the relationship appears to be almost linear between TAR and entropy. However, for different ζ the slope and intercept of this line appears to be different. Figure 7b shows the same scatter plot restricted to $\zeta = 15$ or uniform sampling. Uniform sampling appears to have a lower "intercept" but a more modest slope as one increases k.

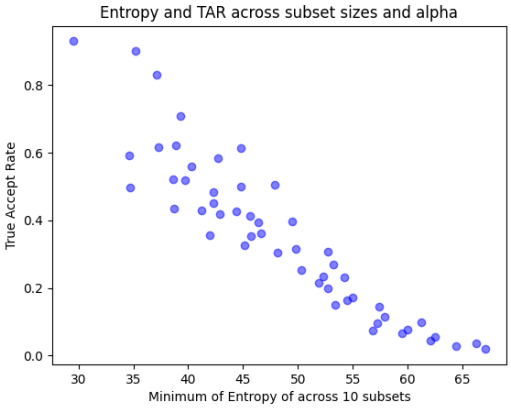
For a fixed subset size k, the relationship between TAR and entropy as one adjusts ζ is not monotone. There is often an optimum ζ that increases both TAR and entropy (positive slope) before one has to sacrifice TAR to achieve higher entropy (negative slope).

For a fixed entropy level, the best setting of ζ with respect to TAR is always nonzero. For different entropy levels, we report on the parameters that yield the best TAR in Table 3. For a fixed TAR level the best setting of ζ with respect to entropy is always nonzero (Table 4). For a fixed entropy level, ζ sampling nearly doubles the TAR. When uniform sampling can achieve a TAR level, ζ-sampling usually yields 5 bits more entropy.

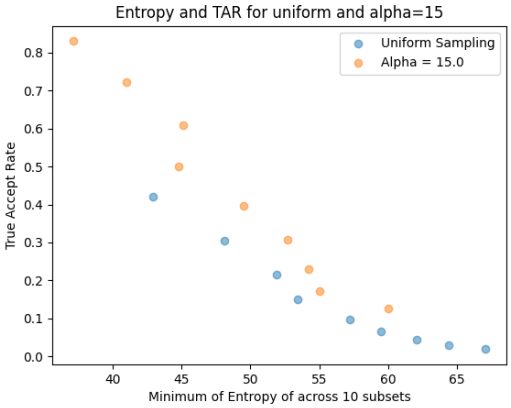
At larger values of k, one naturally must use a smaller ζ as one has to include more distinct indices. This is necessary to get a subset of size k and for the various subsets to have small enough overlap for them to contribute some probability of unlocking.

We considered the following baseline parameter regimes and perform deep dives on the mid and high-security settings:

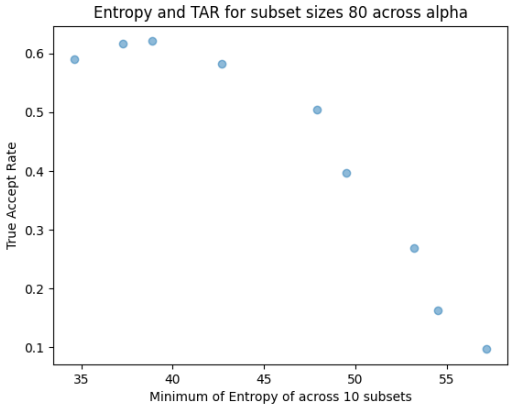
- **Low-security** $\zeta = 20, k = 60$ yielding a TAR of 90% and a minimum entropy of 35.
- **Mid-security** $\zeta = 20, k = 80$ with a TAR of 51% and a minimum entropy of 48.
- **High-security** $\zeta = 10, k = 95$ with a TAR of 10% and a minimum entropy of 61.



(a) All parameters across k, ζ.



(b) Restriction to $\zeta = 15$ across subset sizes. For fixed ζ as k increases TAR quickly decays with an increase in entropy.



(c) Variation of ζ for fixed subset size. We typically observe an initial increase in both TAR and entropy followed by a decrease in TAR to further increase entropy.

Figure 7: Scatter plot between entropy (across 10 runs) and TAR for different values of ζ and k.

9.2 Detailed analysis of TAR vs. entropy

Our estimated security level is the minimum of all chosen subsets. This means that our security estimate (and the adversary's job) is impacted by outliers. However, by selecting subsets at Setup time, one can exclude subsets with a low entropy assessment. To illustrate this for the high security regimes of $\zeta = 10, k = 95$ we sampled $500,000$ subsets. A histogram of the min-entropy of this subsets is shown in Figure 8 and statistics are shown in Table 5. Note that the minimum of entropies, our security figure of merit, is 12 bits lower than the average min-entropy across subsets which was incorrectly used as a figure of merit in Simhadri et al. [SSF19]. We also note that when one computes the average min-entropy or minimum of min-entropies using only 10 subsets one overestimates these values. This underscores the importance of performing the entropy test on all subsets.

Recovering Average-Case Behavior As described above, there are a small number of subsets with low min-entropy where an attacker can focus their attention. By sampling $\mu > 250$K subsets we may be able to exclude

	Best ζ-sampling			Best uniform	
Ent Level	TAR	k	ζ	TAR	k
≥ 30	.90	60	20	.42	60
≥ 35	.90	60	20	.42	60
≥ 40	.62	75	20	.42	60
≥ 45	.51	80	20	.30	65
≥ 50	.31	85	15	.22	70
≥ 55	.17	95	15	.10	80
≥ 60	.10	95	10	.04	90
≥ 65	.04	100	5	.02	100

Table 3: Best TAR for each entropy levels.

	Best ζ-sampling			Best uniform	
TAR Level	Ent	k	ζ	Ent	k
$\geq .1$	61	95	10	53	75
$\geq .2$	54	90	15	52	70
$\geq .3$	53	85	15	48	65
$\geq .4$	48	80	20	43	60
$\geq .5$	48	80	20	$\perp$	$\perp$
$\geq .6$	45	75	20	$\perp$	$\perp$
$\geq .7$	39	60	10	$\perp$	$\perp$
$\geq .8$	37	60	15	$\perp$	$\perp$
$\geq .9$	35	60	20	$\perp$	$\perp$

Table 4: Best entropy for each TAR level.

subsets that have low min-entropy. A natural concern about excluding low min-entropy subsets is that they are responsible for a disproportionate amount of TAR. That is, that the entropy of subset is inversely proportional to its contribution to TAR. We study this question next and show that one can cut off the tail of the entropy curve without eliminating "most" of the TAR.

For the mid-security regime, denoted as `Mid` we sample and store $250K$ subsets. For the high-security regime, denoted as `High`, we sample and store $500K$ subsets. For both parameter sets, we compute the individual entropy of each sampled subset using `EntTest`. Our hypothesis is that if low-entropy subsets are "rare" one can exclude them from the output of `Setup`. To understand the impact of excluding low min-entropy subsets, Table 6 shows a TAR test performed with different portions of the `Mid` and `High` subsets. We consider three sampling regimes, random subsets, the highest entropy subsets, and the lowest entropy subsets. Rows in bold in Table 6 are our final recommended parameters. These are the $200K$ subsets of the `Mid` security setting yielding 50 bits of security at a 45% TAR. The $400K$ highest entropy subsets for the `High` security regime yielding 65 bits of security at a 12% TAR. These subsets are included in our configuration [rep].

Discussion There is a correlation between the TAR contributed by a subset and whether it is high entropy or low entropy ("*all*" refers to sampling all sets in the table). However, this effect is smaller than the effect of the number of subsets, showing that for ζ sampling TAR is sublinear function in the number of subsets.

10 Discussion and Conclusion

This work presents `FiveEyes`, an iris key derivation system that yields 50 bits of security at a 45% TAR or 65 bits of security at a 12% TAR. If one incorporates a password with an estimated entropy of 22 bits [KSK+11, Bon12, WZW+16], this would yield mid and high-security estimates of 72 and 87 bits respectively. The sample-then-lock construction naturally supports prepending of a password. Our scheme drastically improves on the security and

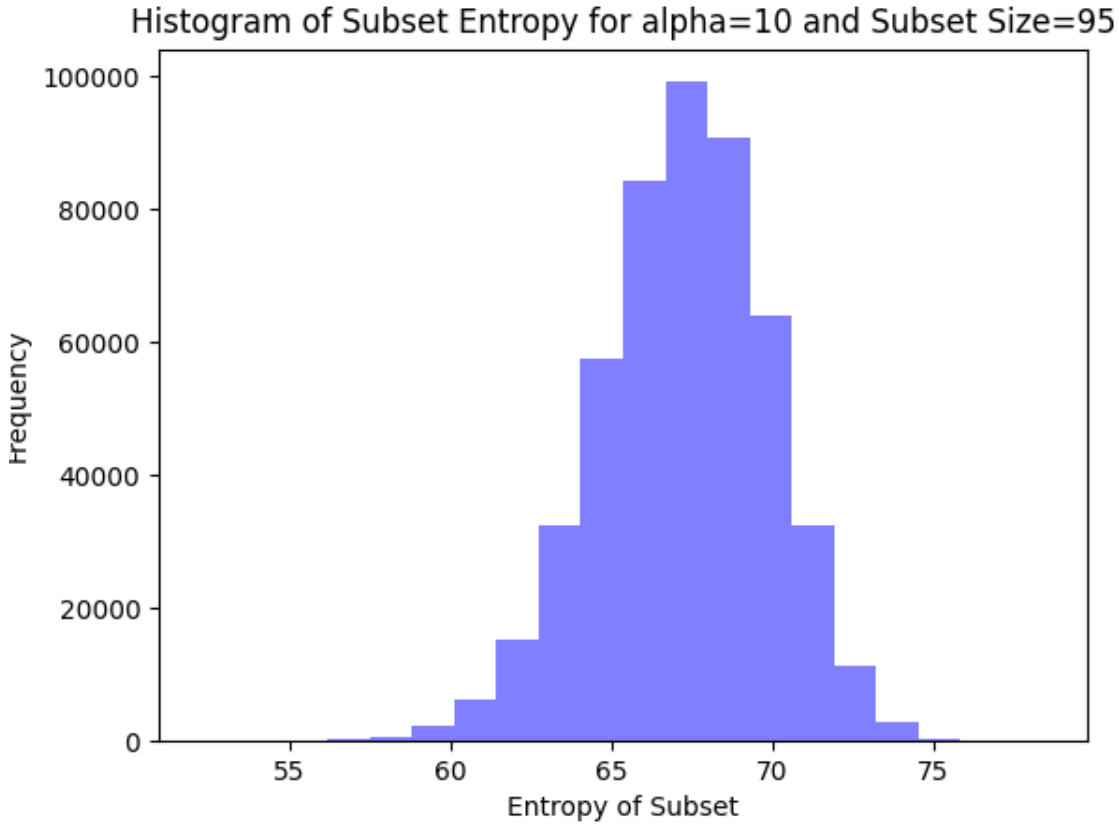


Figure 8: Histogram of entropy for subsets for high security settings with $\zeta = 10$ and $\beta = 500,000$.

efficiency of prior work for iris key derivation. In particular, for our mid-security regime we observe a five-fold reduction in storage cost and a 18 bit improvement in security using a more stringent security measure. Our two technical contributions of a new feature extractor and new subset selection algorithm work together. Each improves security over prior work by roughly 10 bits.

More work is needed to increase both security and TAR. Iris averaging techniques are discussed in the Introduction. Another option is the use of *local confidence* [HRvD+16, DFR21], where one estimates the $p_{\texttt{same},i}$ for the current iris based on the current reading. This approach cannot be used with sample-then-lock as one has globally determined what subsets to use. Prior work using local confidence [HRvD+16, DFR21] allows one to test every subset, this is likely substantially simplify the adversary's job as shown by the spread in entropy among subsets.

Acknowledgements

The authors are grateful to the reviewers for their important comments in improving this work. The work of B.F. is supported by NSF grants #2141033 and #2232813. L.D. was supported by the Harriott Fellowship while at the University of Connecticut. S.A. was supported by a fellowship from Synchrony Inc. and the State of Connecticut while at the University of Connecticut. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19020700008. This material is based upon work supported by the Defense Advanced Research Projects Agency, DARPA, under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views

Entropy Statistic	Value
Minimum	52
20%	65
25%	66
Median	67
75%	69
Maximum	78
Average	67
$\tilde{\mathrm{H}}_\infty$	64
$\tilde{\mathrm{H}}_\infty$ first 10 subsets	66

Table 5: Various entropy measurements across 500K subsets for high entropy regimes. Simhadri et al. consider average min-entropy of first 10 subsets while we consider the minimum of entropies for included subsets, excluding those with low entropy. For this subset selection, the gap between these measures is 14.

Params	Subsets	$\min \mathrm{H}_\infty$	TAR	TAR/#Subsets
`Mid`	250k all	37	.51	$2.0 * 10^{-6}$
$k = 80$	200k lowest	37	.49	$2.5 * 10^{-6}$
	150k lowest	37	.48	$3.2 * 10^{-6}$
$\zeta = 20$	**200k highest**	50	.45	$2.3 * 10^{-6}$
	150k highest	52	.40	$2.7 * 10^{-6}$
`High`	500k all	52	.14	$2.7 * 10^{-7}$
$k = 95$	400k lowest	52	.13	$3.2 * 10^{-7}$
	250k lowest	52	.11	$4.4 * 10^{-7}$
$\zeta = 10$	**400k highest**	65	.12	$2.9 * 10^{-7}$
	250k highest	67	.09	$3.4 * 10^{-7}$

Table 6: Comparison of entropy versus TAR for subsets for mid and high security parameters. TAR is computed across chosen subsets. For subsets of size 80, 200k subsets with the largest entropy represent our medium security parameters. For subsets of size 95, 400k subsets with the largest entropy represent our high-security parameters.

of DARPA. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

[ABC+18] Quentin Alamélou, Paul-Edmond Berthier, Chloé Cachet, Stéphane Cauchie, Beñjamin Fuller, Philippe Gaborit, and Sailesh Simhadri. Pseudoentropic isometries: A new framework for fuzzy extractor reusability. In *AsiaCCS*, 2018.

[ACEK17] Daniel Apon, Chongwon Cho, Karim Eldefrawy, and Jonathan Katz. Efficient, reusable fuzzy extractors from LWE. In *International Conference on Cyber Security Cryptography and Machine Learning*, pages 1–18. Springer, 2017.

[ACF+22] Daniel Apon, Chloe Cachet, Benjamin Fuller, Peter Hall, and Feng-Hao Liu. Nonmalleable digital lockers androbust fuzzy extractors intheplain model. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022*, pages 353–383, Cham, 2022. Springer Nature Switzerland.

[AF18] Sohaib Ahmad and Benjamin Fuller. Unconstrained iris segmentation using convolutional neural networks. In *Asian Conference on Computer Vision*, pages 450–466. Springer, 2018.

[AF19] Sohaib Ahmad and Benjamin Fuller. Thirdeye: Triplet based iris recognition without normalization. In *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–9. IEEE, 2019.

[AF20] Sohaib Ahmad and Benjamin Fuller. Resist: Reconstruction of irises from templates. In *2020 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10. IEEE, 2020.

[AMF22] Sohaib Ahmad, Kaleel Mahmood, and Benjamin Fuller. Inverting biometric models with fewer samples: Incorporating the output of multiple models. In *2022 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–11. IEEE, 2022.

[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.

[BCKP17] Nir Bitansky, Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On virtual grey box obfuscation for general circuits. *Algorithmica*, 79(4):1014–1051, 2017.

[BCP13] Julien Bringer, Hervé Chabanne, and Alain Patey. SHADE: Secure hamming distance computation from oblivious transfer. In *International Conference on Financial Cryptography and Data Security*, pages 164–176. Springer, 2013.

[BDCG13] Carlo Blundo, Emiliano De Cristofaro, and Paolo Gasti. EsPRESSo: efficient privacy-preserving evaluation of sample set similarity. In *Data Privacy Management and Autonomous Spontaneous Security*, pages 89–103. Springer, 2013.

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

[BF16] Kevin W Bowyer and Patrick J Flynn. The nd-iris-0405 iris image dataset. *arXiv preprint arXiv:1606.04853*, 2016.

[BG11] Marina Blanton and Paolo Gasti. Secure and efficient protocols for iris and fingerprint identification. In *European Symposium on Research in Computer Security*, pages 190–209. Springer, 2011.

[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.

[Bon12] Joseph Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *2012 IEEE Symposium on Security and Privacy*, pages 538–552. IEEE, 2012.

[Boy04] Xavier Boyen. Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM conference on Computer and Communications Security*, pages 82–91, 2004.

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

[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.

[CFP+16] Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam Smith. Reusable fuzzy extractors for low-entropy distributions. In *Advances in Cryptology – EUROCRYPT*, pages 117–146. Springer, 2016.

[CFP+21] Ran Canetti, Benjamin Fuller, Omer Paneth, Leonid Reyzin, and Adam Smith. Reusable fuzzy extractors for low-entropy distributions. *Journal of Cryptology*, 34(1):1–33, 2021.

[CHL+15] Jung Hee Cheon, Kyoohyung Han, Changmin Lee, Hansol Ryu, and Damien Stehlé. Cryptanalysis of the multilinear map over the integers. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 3–12. Springer, 2015.

[CKVW10] Ran Canetti, Yael Tauman Kalai, Mayank Varia, and Daniel Wichs. On symmetric encryption and point obfuscation. In *Theory of Cryptography Conference*, pages 52–71. Springer, 2010.

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

[Dau04a] John Daugman. How iris recognition works. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):21 – 30, January 2004.

[Dau04b] John Daugman. Iris recognition border-crossing system in the uae. *International Airport Review*, 8(2), 2004.

[DCH+16] Siddhant Deshmukh, Henry Carter, Grant Hernandez, Patrick Traynor, and Kevin Butler. Efficient and secure template blinding for biometric authentication. In *Communications and Network Security (CNS), 2016 IEEE Conference on*, pages 480–488. IEEE, 2016.

[DFM98] George I Davida, Yair Frankel, and Brian J Matt. On enabling secure applications through off-line biometric identification. In *Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No. 98CB36186)*, pages 148–157. IEEE, 1998.

[DFR21] Luke Demarest, Benjamin Fuller, and Alexander Russell. Code offset in the exponent. In *2nd Conference on Information-Theoretic Cryptography (ITC 2021)*, 2021.

[DHP+18] Pierre-Alain Dupont, Julia Hesse, David Pointcheval, Leonid Reyzin, and Sophia Yakoubov. Fuzzy password-authenticated key exchange. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 393–424. Springer, 2018.

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

[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.

[DS05] Yevgeniy Dodis and Adam Smith. Correcting errors without leaking partial information. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 654–663, 2005.

[EHKM11] David Evans, Yan Huang, Jonathan Katz, and Lior Malka. Efficient privacy-preserving biometric identification. In *Proceedings of the 17th conference Network and Distributed System Security Symposium, NDSS*, 2011.

[FF20] Peter Fenteany and Benjamin Fuller. Same point composable and nonmalleable obfuscated point functions. In *Applied Cryptography and Network Security: 18th International Conference, ACNS 2020, Rome, Italy, October 19–22, 2020, Proceedings, Part II 18*, pages 124–144. Springer, 2020.

[FJR15] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.

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

[FP19] Benjamin Fuller and Lowen Peng. Continuous-source fuzzy extractors: source uncertainty and insecurity. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 2952–2956. IEEE, 2019.

[FRS16] Benjamin Fuller, Leonid Reyzin, and Adam Smith. When are fuzzy extractors possible? In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 277–306. Springer, 2016.

[FRS20] Benjamin Fuller, Leonid Reyzin, and Adam Smith. When are fuzzy extractors possible? *IEEE Transactions on Information Theory*, 66(8):5282–5298, 2020.

[Ful23] Benjamin Fuller. Impossibility of efficient information-theoretic fuzzy extraction. Cryptology ePrint Archive, Paper 2023/172, 2023. `https://eprint.iacr.org/2023/172`.

[GGH13a] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *Advances in Cryptology–EUROCRYPT 2013*, pages 1–17. Springer, 2013.

[GGH+13b] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. *Proc. of FOCS*, 2013.

[GKW17] Rishab Goyal, Venkata Koppula, and Brent Waters. Lockable obfuscation. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 612–621. IEEE, 2017.

[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.

[GPSZ17] Sanjam Garg, Omkant Pandey, Akshayaram Srinivasan, and Mark Zhandry. Breaking the subexponential barrier in obfustopia. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 156–181. Springer, 2017.

[GRGB+12] Javier Galbally, Arun Ross, Marta Gomez-Barrero, Julian Fierrez, and Javier Ortega-Garcia. From the iriscode to the iris: A new vulnerability of iris recognition systems. *Black Hat Briefings USA*, 1:8, 2012.

[GZ19] Steven D Galbraith and Lukas Zobernig. Obfuscated fuzzy hamming distance and conjunctions from subset product problems. In *Theory of Cryptography: 17th International Conference, TCC 2019, Nuremberg, Germany, December 1–5, 2019, Proceedings, Part I*, pages 81–110. Springer, 2019.

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

[HBF08] Karen P Hollingsworth, Kevin W Bowyer, and Patrick J Flynn. The best bits in an iris code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):964–973, 2008.

[HBL17] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[HKMC23] Gabriel Emile Hine, Ridvan Salih Kuzu, Emanuele Maiorana, and Patrizio Campisi. Unlinkable zero-leakage biometric cryptosystem: Theoretical evaluation and experimental validation. *IEEE Transactions on Information Forensics and Security*, 2023.

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

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

[HRvD+16] Charles Herder, Ling Ren, Marten van Dijk, Meng-Day Yu, and Srinivas Devadas. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. *IEEE Transactions on Dependable and Secure Computing*, 2016.

[ICF+15] Gene Itkis, Venkat Chandar, Benjamin W Fuller, Joseph P Campbell, and Robert K Cunningham. Iris biometric security challenges and possible solutions: For your eyes only? using the iris as a key. *IEEE Signal Processing Magazine*, 32(5):42–53, 2015.

[KHF+20] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, et al. Spectre attacks: Exploiting speculative execution. *Communications of the ACM*, 63(7):93–101, 2020.

[KSK+11] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Serge Egelman. Of passwords and people: measuring the effect of password-composition policies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2595–2604. ACM, 2011.

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

[LSG+18] Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg. Meltdown. *arXiv preprint arXiv:1801.01207*, 2018.

[LWY+17] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

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

[MSZ16] Eric Miles, Amit Sahai, and Mark Zhandry. Annihilation attacks for multilinear maps: Cryptanalysis of indistinguishability obfuscation over ggh13. In *Annual Cryptology Conference*, pages 629–658. Springer, 2016.

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

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

[MZ17] Fermi Ma and Mark Zhandry. The mmap strikes back: obfuscation and new multilinear maps immune to clt13 zeroizing attacks. Technical report, Cryptology ePrint Archive, Report 2017/946, 2017.

[ODGS16] Nadia Othman, Bernadette Dorizzi, and Sonia Garcia-Salicetti. Osiris: An open source iris recognition software. *Pattern Recognition Letters*, 82:124–131, 2016.

[PAB+18] Andrew Prout, William Arcand, David Bestor, Bill Bergeron, Chansup Byun, Vijay Gadepally, Michael Houle, Matthew Hubbell, Michael Jones, Anna Klein, et al. Measuring the impact of spectre and meltdown. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–5. IEEE, 2018.

[PBF+08] P Jonathon Phillips, Kevin W Bowyer, Patrick J Flynn, Xiaomei Liu, and W Todd Scruggs. The iris challenge evaluation 2005. In *2008 IEEE Second International Conference on Biometrics: Theory, Applications and Systems*, pages 1–8. IEEE, 2008.

[Pro09] Hugo Proenca. Iris recognition: On the segmentation of degraded images acquired in the visible wavelength. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1502–1516, 2009.

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

[rep] https://anonymous.4open.science/r/CompFE-51FB/.

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

[SSF19] Sailesh Simhadri, James Steel, and Benjamin Fuller. Cryptographic authentication from the iris. In *International Conference on Information Security*, pages 465–485. Springer, 2019.

[ŠTO05] Boris Škorić, Pim Tuyls, and Wil Ophey. Robust key extraction from physical uncloneable functions. In *Applied Cryptography and Network Security: Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005. Proceedings 3*, pages 407–422. Springer, 2005.

[TKAK23] Gioacchino Tangari, Shreesh Keskar, Hassan Jameel Asghar, and Dali Kaafar. On the adversarial inversion of deep biometric representations. *arXiv preprint arXiv:2304.05561*, 2023.

[TVN20] Veeru Talreja, Matthew C Valenti, and Nasser M Nasrabadi. Deep hashing for secure multimodal biometrics. *IEEE Transactions on Information Forensics and Security*, 16:1306–1321, 2020.

[Var10] Mayank Harshad Varia. *Studies in program obfuscation*. PhD thesis, Massachusetts Institute of Technology, 2010.

[VV10] Gregory Valiant and Paul Valiant. A CLT and tight lower bounds for estimating entropy. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 17, page 9, 2010.

[VV11] Gregory Valiant and Paul Valiant. Estimating the unseen: an n/log (n)-sample estimator for entropy and support size, shown optimal via new CLTs. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 685–694. ACM, 2011.

[WCD+17] Joanne Woodage, Rahul Chatterjee, Yevgeniy Dodis, Ari Juels, and Thomas Ristenpart. A new distribution-sensitive secure sketch and popularity-proportional hashing. In *Annual International Cryptology Conference*, pages 682–710. Springer, 2017.

[WL18] Yunhua Wen and Shengli Liu. Robustly reusable fuzzy extractor from standard assumptions. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 459–489. Springer, 2018.

[WLH18] Yunhua Wen, Shengli Liu, and Shuai Han. Reusable fuzzy extractor from the decisional Diffie–Hellman assumption. *Designs, Codes and Cryptography*, Jan 2018.

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under lwe. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 600–611. IEEE, 2017.

[WZW+16] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. Targeted online password guessing: An underestimated threat. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1242–1254. ACM, 2016.

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

[ZD08] Sheikh Ziauddin and Matthew N Dailey. Iris recognition performance enhancement using weighted majority voting. In *2008 15th IEEE International Conference on Image Processing*, pages 277–280. IEEE, 2008.

[Zha19] Mark Zhandry. The magic of elfs. *Journal of Cryptology*, 32:825–866, 2019.

[ZLN12] Lin Zhang, Hongyu Li, and Junyu Niu. Fragile bits in palmprint recognition. *IEEE Signal processing letters*, 19(10):663–666, 2012.