IrisLock: Iris Biometric Key Derivation with 42 bits of security

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

Caleb Manicke[¶], Alexander Russell[‖], Amey Shukla[**]

November 5, 2024

Abstract

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

This is frustrating given the research that has developed theoretical tools—known as fuzzy extractors—that enable secure, privacy-preserving biometric authentication with *public* enrollment data (Dodis et al., SIAM JoC 2008). Unfortunately, fuzzy extractor systems either:

1. Make strong independence assumptions, such as:
   (a) Bits of biometrics are i.i.d. (or that all correlation is pairwise between features (Hine et al., TIFS 2023)), or
   (b) For an error-correcting code, the nearest codeword and the coset of biometric readings are independent (Zhang, Cui, and Yu, ePrint 2021/1559).

   These assumptions either have not been statistically checked or statistical analysis indicates they are false.
2. Or use incorrect cryptographic analysis. Simhadri et al. (ISC, 2019) assume the security of sample-then-lock (Canetti et al., Journal of Cryptology 2021) is captured by the average min-entropy of subsets. Zhang et al. (ICPR, 2022) show an attack on this incorrect analysis.

This work introduces IrisLock, an iris key derivation system powered by technical advances in both 1) feature extraction from the iris and 2) the fuzzy extractor used to secure authentication keys. The fuzzy extractor builds on sample-then-lock (Canetti et al., Journal of Cryptology 2021). We correct a proof in Canetti et al. and show the minimum of min-entropy of subsets is the relevant security measure. Our primary parameters are 42 bits of security at 45% true accept rate (TAR). Our quantitive level of security is as good as the above systems, Simhadri et al's incorrect analysis yields an estimate of 32 bits, while Zhang et al.'s system on the face estimates 45 bits (with the independence condition). One can easily incorporate a password, boosting security to 64 bits.

Irises used to evaluate TAR and security are class disjoint from those used for training and collecting statistics (the open dataset regime). The only statistical assumption made is necessary: the accuracy of min-entropy estimation.

[*] University of Connecticut. sohaib50k@gmail.com.

[†] Adelphi University. chensixia09@gmail.com.

[‡] Gonzaga University. onlylukejohnson@gmail.com.

[§] University of Connecticut. benjamin.fuller@uconn.edu.

[¶] University of Connecticut. caleb.manicke@uconn.edu.

[‖] University of Connecticut. acr@uconn.edu.

[**] University of Connecticut. amey.shukla@uconn.edu.

1 Introduction

Biometric authentication is widely adopted in practice. There is a longstanding, qualitative, and quantitive disconnect between the desirable security guarantees offered—in principle—by theoretical approaches and deployed solutions.

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 established practical attacks in the event of an exposure [GRGB$^+$12, FJR15, AF20, AMF22, TKAK23, WGCJ22, LNWS23, ASW$^+$24]. The common approach to mitigate this leakage threat is to place the authentication algorithm in a trusted execution environment, which have proven to be difficult to design correctly [PAB$^+$18, KHF$^+$20, LSG$^+$18] (and, of course, place an additional hardware burden on the device). Ideally, cryptography could mitigate some or all of this burden.

The cryptography community has identified and studied the formal notion of a *fuzzy extractor* [BBR88, DRS04, DORS08, ŠTO05, HAD06, DKRS06, FMR13, CFP$^+$16, ACEK17, ABC$^+$18, WLH18, WL18, DFR21, ACF$^+$22],[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.

State of Prior Work Since their *introduction*, fuzzy extractors have sufficed for concrete security if: 1) bits of biometric W are i.i.d. [Mau93, MW96, MTV09, YD10, HMSS12, LC23], 2) good error-correcting codes exist, and 3) the entropy "rate" of the biometric is greater than the error "rate." Assume, for example, that $|W| = n = 1024$ and one wishes to correct μ fraction of errors. Let $\texttt{ent}$ be the min-entropy of W and $\texttt{err} := n * h_2(\mu)$ where h_2 is binary entropy.[2] To correct a μ fraction of errors, one must write down $\texttt{err}$ bits about the biometric. The code-offset or syndrome constructions [DORS08] match this bound if a perfect code exists for the particular n, μ. The quantity

$$\texttt{FE}_{\texttt{Qual}} := \texttt{ent} - \texttt{err}$$

then measures how many bits of security [CFP$^+$16, Proposition 1] these constructions provide via a conditional entropy argument.

For the iris, using a state of the art feature extractor [AF19], $\mu \geq .19$ and bits of different irises agree with probability .5. If one assumes W is i.i.d. then $\texttt{ent} = 1024$ and $\texttt{FE}_{\texttt{Qual}}(W) = 1024 - 718 = 306$. However, when one uses optimistic, heuristic statistical tests to estimate the entropy of W, the entropy is < 250 and $\texttt{FE}_{\texttt{Qual}}$ is negative. Indeed, **all existing statistical analysis of biometrics shows that bits of W are not i.i.d. [Dau04a].** To overcome this correlation, a natural goal is to design feature extractors that produce independent features. Hine et al. [HKMC23] take an important step, designing a variant of principal/independent component analysis to create independent features while controlling how much noise is in the new features. Such principal component analysis-based algorithms can remove pairwise correlation between features. Unfortunately, the correlation between features is higher dimensional, showing up on larger sets of features [SSF19, Figure 2].

As of this writing, prior work that provides nonzero security for a biometric can be classified into two categories:

1. **Independence** Assumes some independence of biometric features [GKTF16, ZCY21, HKMC23]. As mentioned above, Hine et al. [HKMC23] attempt to handle pairwise correlation. For a family of error-correcting codes, Zhang et al. [ZCY21] assume that the nearest codeword and coset with respect to the code of a face feature vector are independent.

2. **Incorrect Analysis** Misapplies cryptographic techniques, overestimating security level. As we discuss below, Simhadri et al. [SSF19] build on a construction [CFP$^+$21, Theorem 1] with an incorrect proof. They estimate security level as the average min-entropy of subsets of features. 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 entropy of subsets

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

[2] Binary entropy of a binary random variable with probability μ of being 1.

of features used in the system. Zhu et al. [ZSC$^+$22] present an attack on Simhadri et al.'s system that targets the lowest entropy subset (and hill climbs based on learned information).[3]

The Goal: Given these issues, we focus on providing concrete security assuming only *accuracy of min-entropy assessment* [Dau04a] (described in Section 4.2). Entropy assessment is inherently heuristic [VV11] but necessary. We focus on the iris in this work.

System Overview This work introduces `IrisLock`, an integrated iris feature extractor and fuzzy extractor. We estimate, 42 bits of security, 64 bits with a password,[4] at 45% true accept rate (TAR). We call this parameter regime `Forty`. We have a second parameter regime of 33 bits of security at 87% TAR called `Thirty` (blue cells in Table 4). Our system builds on the sample-then-lock fuzzy extractor [CFP$^+$16] taking subsets of size 65 and 60 respectively. Our system uses a feature extractor that intentionally produces *heterogeneous* features in contrast to most biometric feature extractors. We show this heterogeneity produces stronger security and allows for a sharper tradeoff between TAR and security.

`IrisLock` provides better concrete security than was claimed in previous iris work [SSF19]. As mentioned above, Simhadri et al.'s [SSF19] analysis is incorrect; despite using a stronger (and correct) metric security we improve on their reported security by 10 bits of security (at comparable TAR). We also provide comparable security (and TAR) to Zhang et al. [ZCY21] work on the facial biometric.[5] Zhang et al.'s work assumes the biometric features are independent from the used error correcting code without any evaluation.

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 reflected by such security measures is 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). `IrisLock` provides stronger properties: no information is leaked about the enrolled 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 Section 2 provides an overview, Section 3 introduces mathematical preliminaries, Section 4 describes the datasets, Section 5 introduces our feature extractor, Section 6 describes the fuzzy extractor, Section 7 describes the major technical change to our fuzzy extractor, Section 8 evaluates, and Section 9 concludes.

2 System Overview

`IrisLock` is a combination of a **heterogeneous feature extractor** built from a convolutional neural network (CNN) and a **fuzzy extractor** built from the sample-then-lock fuzzy extractor [CFP$^+$16]. The overall system is shown in Figure 1. We briefly review the sample-then-lock fuzzy extractor [CFP$^+$21], which is the core of our fuzzy extractor, 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 when one creates ciphertexts with correlated keys that only have entropy [CKVW10]. (The formal definition is based on virtual black-box obfuscation [BGI$^+$01], see Def. 3.) The system sets β different subsets as input to the digital locker with the same key as output. As we note in Lemma 1, subsets can be sampled from any distribution that is not dependent on the enrollment value.

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

[3] Zhu et al. also propose a modification of sample-then-lock that selects subsets differently for each iris, yielding much larger subsets. However, they do not consider an adversary that analyzes the selected subsets to learn about the individual's iris.

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

[5] We average multiple readings to boost TAR, this is common in the biometrics community, see Section 8.3.

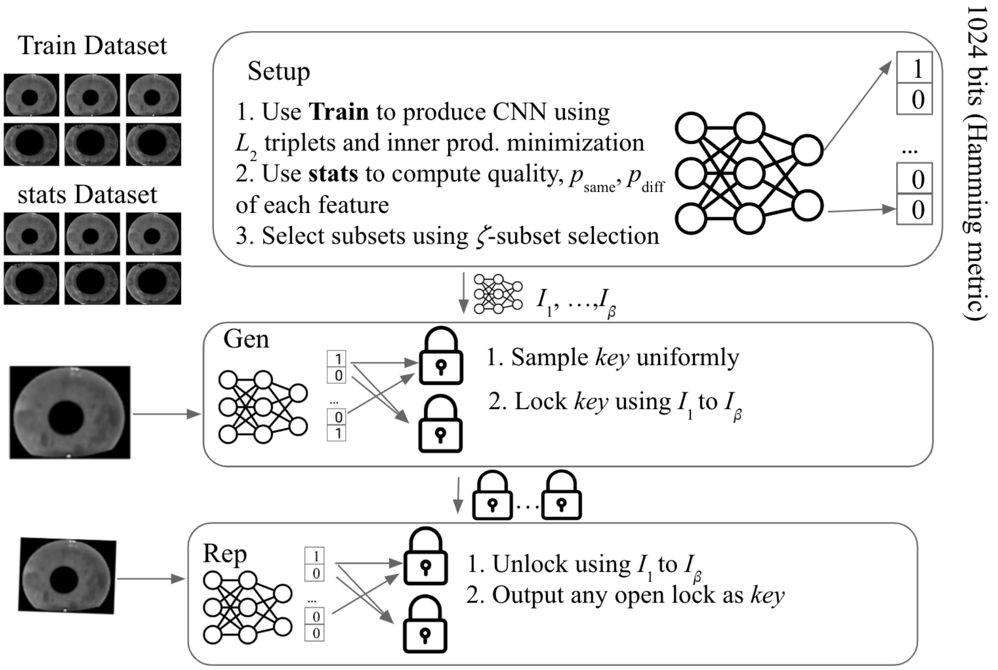

Train Dataset
stats Dataset
Setup
1. Use Train to produce CNN using
L_2 triplets and inner prod. minimization
2. Use stats to compute quality, p_{same}, p_{diff}
of each feature
3. Select subsets using ζ-subset selection
1
0
...
0
0
1024 bits (Hamming metric)
$I_1, \ldots, I_\beta$
Gen
1
0
...
0
1
1. Sample key uniformly
2. Lock key using I_1 to I_β
Rep
1
0
...
0
0
1. Unlock using I_1 to I_β
2. Output any open lock as key

Figure 1: Overall System Architecture

to the bits in some $\mathcal{I}_j$. **The minimum of min-entropy of subsets is the relevant security metric (see Assumption 2).** In all of our measurements, the entropy of each subset is smaller than its size, often by at least 50%, see Table 4. As mentioned in our `Thirty` parameters subsets are of size 60 while the entropy estimate is 33.

2.1 To be dependent or to be independent

As mentioned above, most feature extractors try to produce independent homogeneous features. Current methods are capable of preventing small k-wise correlations between features. Given homogeneous and independent features uniform subsets are optimal. On the iris using the ND-0405 dataset [BF16] (which we also use), Simhadri et al. [SSF19] showed for (small modifications to) an open-source feature extractor [ODGS16], sample-then-lock claims 32 bits of security with a 60% TAR using $\beta = 10^6$. Using modern feature extractor techniques (discussed in Section 5) gives comparable results, one can achieve 36 bits of security with a TAR of 20% using $\beta = 250000$. These numbers are not quite comparable, Simhadri et al. [SSF19] is the average min-entropy of 10 subsets, while our number is the minimum of min-entropy of 10 subsets. The ND-0405 dataset is a superset of the NIST Iris Evaluation Challenge [PBF$^+$08].

We *encourage* the feature extractor to produce heterogeneous features with different error rates and different amounts of entropy. We then bias subset selection to be aware of the differences in feature quality and the correlations between features. Our subsets selection is non-uniform and depends on data properties. Thus, we publish subsets which can be used globally for all users. To make this change, we consider three class-disjoint datasets in this work, one for training the feature extractor, one for computing non-uniform subsets, and one for evaluating correctness and security of the fuzzy extractor.

2.2 Our contribution

Our technical contributions are as follows:

1. **Security Analysis** Showing that global sampling of subsets for sample-then-lock is secure. We also fix the proof of sample-then-lock that led to prior work to overestimate security.

2. **ζ-sampling** Design of a non-uniform selection algorithm, called ζ-sampling that samples better bits more frequently.
   (a) Variants of ζ-sampling that use the feature entropy.
   (b) A negative result that shows that sampling pairs of features does not help. This provides evidence that the correlation between features is not pairwise.
3. **Feature Extractor** A feature extractor that produces heterogeneous features.
4. **Parameter Analysis** Extensive parameter analysis showing that our heterogeneous feature extractor combined with non-uniform sampling results in a better security/TAR tradeoff.

Security Analysis We redefine a fuzzy extractor [Ful24] as a triple of algorithms (Setup, Gen, Rep). The Setup algorithm tailors the fuzzy extractor to the biometric of interest, in our case sampling "good" subsets for use in sample-then-lock. 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 $(\mathsf{key}, p) \leftarrow \mathsf{Gen}(w, \mathtt{stats}_W)$,
$$\Pr[\mathsf{key} = \mathsf{Rep}(w', p, \mathtt{stats}_W)] \geq \text{desired TAR}.$$
This is a change from normal fuzzy extractor security where correctness is guaranteed for all w, w' that are close enough according to a distance measure.
2. **Security** The value key is pseudorandom given p.

Our security estimate is **the minimum of the min-entropy 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. There are established techniques for estimating the min-entropy of biometrics [Dau04a] (detailed in Section 4.2). 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. We compute the minimum of min-entropy across all used subsets, in some of our tests this number is 9 bits smaller than the average min-entropy of subsets (see Table 4). Part of our evaluation in **Parameter Analysis** is showing **one can filter out low entropy subsets without sacrificing too much TAR.**

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

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

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 unpredictability of a subset and how likely it is to match.

We introduce a new approach called ζ-sampling that moderates between these extremes. For a parameter $\zeta \in \mathbb{R}^+$ instead of picking bits uniformly a bit i is picked with probability proportional to $p^{\zeta}_{\mathtt{same},i}$, that is,

$$\text{Prob select dimension } i = \frac{p^{\zeta}_{\mathtt{same},i}}{\sum_i p^{\zeta}_{\mathtt{same},i}}.$$

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.

We evaluate three versions of ζ sampling where the numerator of the above is:

$$\begin{aligned}\texttt{LikeOnly} &= p_{\texttt{same},i}^{\zeta},\\ \texttt{UnlikeRatio} &= \left(\frac{p_{\texttt{same},i}}{\max\{p_{\texttt{diff},i}, 1-p_{\texttt{diff},i}\}}\right)^{\zeta},\\ \texttt{UnlikeExp} &= p_{\texttt{same},i}^{\zeta/H_\infty(p_{\texttt{diff},i})}.\end{aligned}$$

where $\mathrm{H}_\infty(p_{\texttt{diff},i}) = -\log(\max\{p_{\texttt{diff},i}, 1-p_{\texttt{diff},i}\})$. The final two weightings are designed to incorporate the entropy represented by each feature.

In addition, we then compute similar statistics for pairs of bit i, j. Sampling by pairs of features does not improve the entropy vs. TAR tradeoff (see Table 2).

Feature Extractor 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. A CNN is trained to produce 1024 bit vectors, where readings of the same biometric are close according to the Hamming metric. We use triplet loss [WS09] and angular margin from SphereFace [LWY$^+$17]. The main change from prior work is a loss term that measures the overall inner product between all pairs of features (like and unlike), encouraging the feature extractor to reduce the norm of vectors. This is used to create heterogeneous features.

Parameter Analysis and Cryptographic Efficiency We use $\beta = 200,000$ subsets for all parameters, Simhadri et al. [SSF19] used 5 times this amount. We perform extensive analysis and publish our chosen subsets. 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 [ADF24]. (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 our parameters (250K lockers in place of 10^6), 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.[6] 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.[7]

2.3 Further Related Work

Throughout, we focus on computational security due to additional negative results on providing information-theoretic security [FRS16, FP19, FRS20, Ful24]. 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: 1) on a new subset product assumption [GZ19], 2) on general-purpose obfuscation techniques [BCKP14, BCKP17, PST13, GGH$^+$13b, GGH13a, CHL$^+$15, MSZ16, GPSZ17, MZ17], and 3) information theoretic techniques requiring exponential time [HR05, FRS16, WCD$^+$17]. The subset product assumption is directly the security of the proposed construction.

[6] 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.

[7] The main work in the construction is HMAC-SHA-512 [BCK96]. 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.

3 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 $\mathtt{poly}(\lambda)$. 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*

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

For distribution ensembles $X := \{X_\lambda\}_{\lambda\in\mathbb{N}}, Y := \{Y_\lambda\}_{\lambda\in\mathbb{N}}$, 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 $x, y \in \{0,1\}^n$, let $\mathsf{dis}(x,y) = |\{i | x_i \neq y_i\}|$ be the Hamming distance between x and y.

We use the version of fuzzy extractors that provides security against computationally bounded adversaries [FMR13]. In addition, we include a setup algorithm that is used globally (called advice by Fuller [Ful24, Definition 8]). Dodis et al. provide a comparable definition for information-theoretic fuzzy extractors [DORS08].

Definition 2. *Let $\mathcal{M} = (\{0,1\}^n, \mathsf{dis}$ be a metric space. Let W, W' be a pair of (correlated) probability distributions where W, W' are over $\mathcal{M}$. Let $\mathtt{stats}_W$ be a string where $|\mathtt{stats}_W| = \mathtt{poly}(n)$. A pair of randomized procedures "setup" (*Setup*), "generate," (*Gen*) and "reproduce" (*Rep*) is an $(\mathcal{M}, W, W', \mathtt{stats}_W, \kappa := \kappa(\lambda))$-*computational fuzzy extractor *with error δ if* Setup*,* Gen *and* Rep *satisfy the following properties:*
Correctness: *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*: Let $\mathtt{advise}_W \leftarrow \mathsf{Setup}(\mathtt{stats}_W)$ and $(R, P) \leftarrow \mathsf{Gen}(W, \mathtt{stats}_W)$ then $\Delta^c((R, P, \mathtt{advise}_W), (U_\kappa, P, \mathtt{advise}_W)) \leq \mathtt{ngl}(\kappa)$.*

Remarks The adversary knows the value of $\mathtt{stats}_W$. The adversary also receives $\mathtt{advise}_W$ to allow for randomized Setup (which we use). We do not 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. Upgrades to robust fuzzy extractors are known in multiple cryptographic models [Boy04, ACF+22, CHRF24].

4 Datasets and Metrics

4.1 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 are considered independent biometrics [Dau04b]. ND-0405 is captured at a near-infrared wavelength. The dataset consists of 64980 images. We use the same training regime as in ThirdEye [AF19]. However, we split their testing set into two sets, one used for producing $\mathtt{stats}_W$ and one for testing.

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

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

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

Training, calculation of $\mathtt{stats}_W$, and testing are all class disjoint. For histograms shown in Figure 2, 10 randomly chosen images for each biometric were taken from the union of $\mathtt{stats}_W$ and testing (all images were used if an iris has fewer than 10 images). The rest of tables and figures only use testing data. Images are segmented (iris portion separated from background) before input to the feature extractors. Segmentation is performed using Ahmad and Fuller [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.

4.2 Metrics

Entropy Test Throughout this work, we use the standard method of Daugman [Dau04a] for estimating the entropy of biometric feature extractors. We adapt this method to consider min-entropy in place of Shannon entropy. The core of the method is measuring the (min-)entropy of a binomial that fits the set of distances between readings of different biometrics. 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 stdev. σ of this histogram.
3. Compute the degrees of freedom $\mathtt{dF} = \mu(1-\mu)/\sigma^2$.
4. Min-entropy is $\mathsf{e} = \min\{-\log(\mu), -\log(1-\mu)\} * \mathtt{dF}$.

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 $\mathcal{I}_1, ..., \mathcal{I}_\beta$. We consider two tests `TARfast` and `TARfull`. The `TARfast` first subsamples from every class including at least two images from each class and uses this sub datasets as input to `TARfull`. `TARfull` takes input DSet where DSet_i is all readings of a single biometric. We compute the following:

1. Set $\mathtt{TAR}_{\mathsf{num}} = 0, \mathtt{TAR}_{\mathsf{denom}} = 0$.
2. 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 $\mathtt{TAR}_{\mathsf{denom}} = \mathtt{TAR}_{\mathsf{denom}} + \gamma$.
   (d) For $j = 1$ to γ: if there exists some i such that $w_{j,\mathcal{I}_i} = w^*_{\mathcal{I}_i}$, then $\mathtt{TAR}_{\mathsf{num}} = \mathtt{TAR}_{\mathsf{num}} + 1$.
3. Output $\mathtt{TAR} = \mathtt{TAR}_{\mathsf{num}}/\mathtt{TAR}_{\mathsf{denom}}$.

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. This means our overall computation of TAR is slightly weighted towards classes with more readings. The above computation also ignores the possibility of digital locker unlocking for multiple values. This would require a collision in HMAC-SHA256. 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 observable deviation in TAR when using the cryptographic implementation.

5 The Feature Extractor

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

ThirdEye [AF19] outlines a two round training pipeline, starting with a model with pre-trained ImageNet weights. Unlike ThirdEye [AF19], which used a ResNet-50 architecture [HZRS16], we use a DenseNet-169 architecture [HLVDMW17].

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 classifying the training set. Training on accurate prediction of class labels allows the model to learn discriminative features between irises that will ideally translate into an open dataset. ThirdEye then replaces the classification component of the network with a new 1024 neuron feature layer with randomly initialized values.

The second stage of training trains the last 20 layers of the model (including the ending weights that were used for classification). For the second stage, the network output features on irises. For each batch, these features are used to produce triplets that are used to compute a distance-based loss function.[8] Each triplet is comprised of one sample that we declared as an anchor, denoted x_a, another sample from the same class, denoted x_p (for positive), and a sample from a different class, denoted x_n. Triplets are chosen so that at current weights the positive sample x_p has high distance from the anchor sample x_a and the distance between the negative sample x_n and the anchor x_a is the smallest in the batch. The Triplet loss function is calculated as:

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

In the above, m is a hyper-parameter that specifies the desired gap between the distances of the same class and different classes, known as margin. L_2 refers to the Euclidean distance. 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. Again, this triplet loss function is calculated and back-propagated to only the last 20 layers of the network.

Instead of optimizing the L_2 distance, Liu et al. [LWY+17] consider the cosine distance 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].

We trained the DenseNet169 architecture according to our ThirdEye pipeline, using just the **Cross-Entropy** loss and fine tuned it using **L_2 triplets** and **angular margin**. Histograms are shown in Figure 2a, 2b, and 2c. For all histograms, all left eye images were used for training while 10 randomly chosen images were taken from the union of $\mathtt{stats}_W$ and the Test dataset. 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/false accept rate (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. As a reminder, the estimated min-entropy is

$$\mathsf{e} = \frac{-\mu_{Unlike}(1 - \mu_{Unlike}) \log \max\{\mu_{Unlike}, 1 - \mu_{Unlike}\}}{\sigma^2_{Unlike}},$$

To maximize entropy means one seeks to set μ_{Unlike} as close to possible with σ^2_{Unlike} as small as possible. However, this increase comes at a cost, the like μ_{Like} error rate increases from .19 for cross-entropy to .24 for angular margin.

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

5.1 Our design - A Heterogeneous Feature extractor

Our design uses the ThirdEye training pipeline to produce a feature extractor with heterogeneous features. Our design combines the ideas of L_2 margin maximization from triplet loss with angle minimization from angular margin. `IrisLock` retains the Cross-Entropy loss for the first stage of training. In the second stage of training, where only the last 20 layers are trained, we compute a modified triplet loss that includes additional weighting for our distance terms and the **inner product** between the anchor and negative examples:

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

In the above m represents the margin as before and is set to .8. c_1 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. Angular margin "forced" the mean of comparisons between templates of different classes to be centered at .5 since vectors were normalized and the loss minimizes 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 μ_{IP} by just reducing the 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. As a side impact, this causes the network to create different quality features with different error rates and entropy as shown in Figure 3.

We do note that this decrease in unlike mean from $\mu_{Unlike} \approx .5$ to $\mu_{Unlike} \approx .4$ shown in Figure 2 does impact the min-entropy test. All of our evaluation considers both the Angular and Heterogeneous feature extractor and shows that the heterogeneous feature extractor has a better TAR/entropy tradeoff.

All the pipelines are optimized by augmenting at train time with random use of image sharpening, rotations of $30°$,$-30°$, and a flip along the horizontal axis.

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

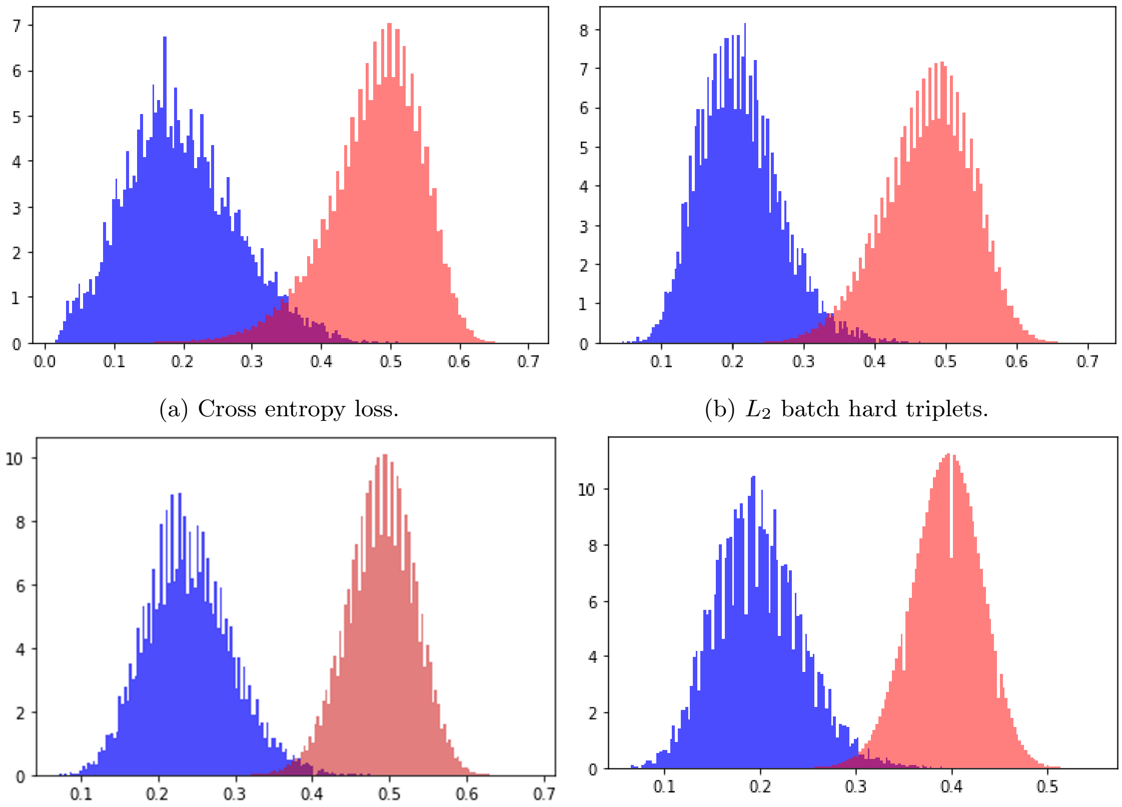
The feature extractor generates the feature vectors of length 1024. Our $\texttt{stats}_W$ consists of 2048 real values from $[0,1]$. From here on we use w to refer to the output of the feature extractor. For each feature i we compute:

$$p_{\texttt{same},i} := \Pr[w_i \neq w_i' | w, w' \text{ readings same biometric}]$$
$$p_{\texttt{diff},i} := \Pr\left[\max\left\{\begin{matrix} w_i \neq w_i' \\ w_i = w_i' \end{matrix}\right\}\middle| w, w' \text{ readings different biometrics}\right]$$

These probabilities are computed across the $\texttt{stats}_W$ dataset. For the `Heterogeneous` feature extractor, there are 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. 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 $p_{\texttt{diff},i}$ and their difference in Figure 3 for both the angular and `Heterogeneous` feature extractor.

Based on these two histograms, we make a few observations. First, for the angular feature extractor, both $p_{\texttt{diff},i}$ and $p_{\texttt{same},i}$ have a small variance (.035 and .030), the gap between features is relatively small, furthermore, the covariance is .00037. For the `Heterogeneous` feature extractor, the variance of $p_{\texttt{diff},i}$ and $p_{\texttt{same},i}$ climbs to .140 and .060 respectively with covariance of .0081 (a multiplicative increase of 20). It appears that features of the `Heterogeneous` feature extractor are different. However, it isn't clear that this is beneficial for security of a fuzzy extractor.

Computation of $\texttt{stats}_W$ on data that is not used for testing is crucial. Computing $p_{\texttt{same},i} - p_{\texttt{diff},i} \geq .4$ for almost all features on training data with $p_{\texttt{same},i} \leq .1$. These error rates did not indicate any variation in quality of features and also did not predict error rates seen on the unseen irises.



(a) Cross entropy loss.

(b) L_2 batch hard triplets.

(c) Angular Margin.

(d) Heterogeneous Feature Extractor.

Figure 2: Distance comparisons for loss functions used in developing `IrisLock`. 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 $\mathtt{stats}_W$ and testing.

Looking ahead to when we test ζ-sampling on pairs we compute the following analogues of the above:

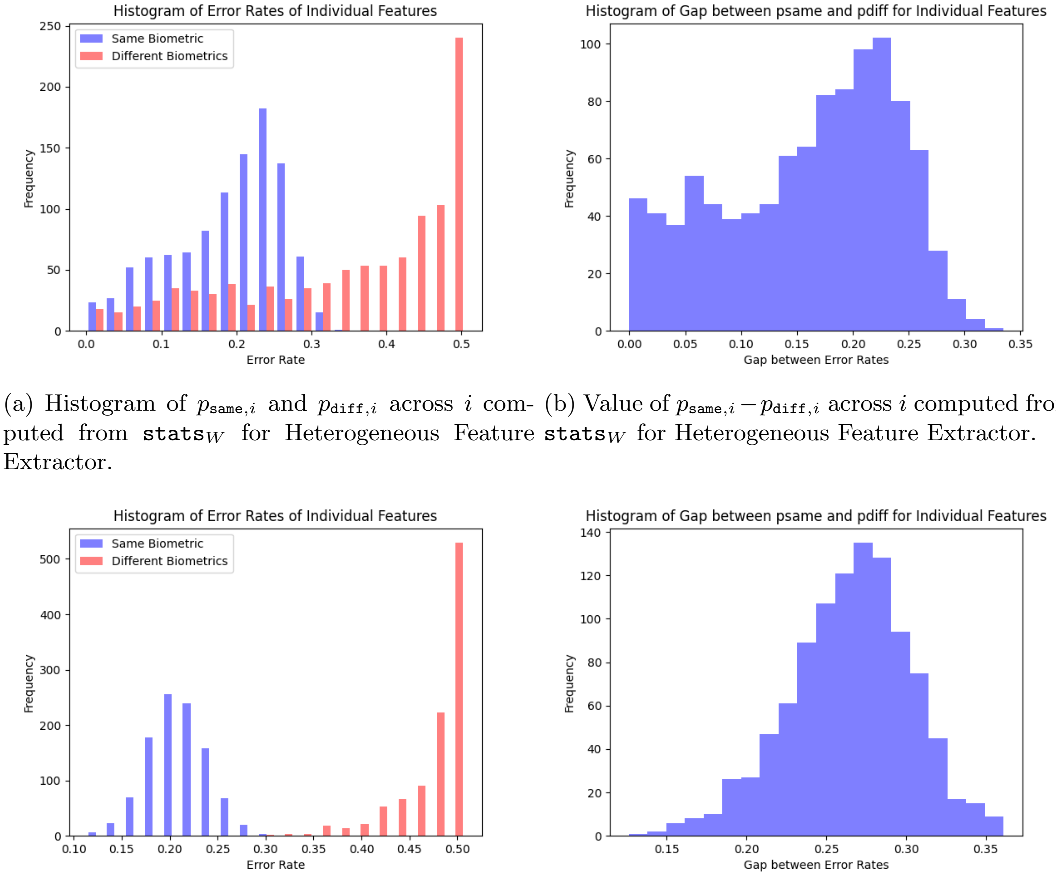
$$p_{\mathtt{same},i,j} := \Pr[w_i = w_i' \wedge w_j = w_j' | w, w' \text{ same biometric}]$$

$$p_{\mathtt{diff},i,j} := \max_c \Pr\left[\begin{matrix} w_i \oplus w_i' || \\ w_j \oplus w_j' \end{matrix} = c \middle| w, w' \text{ different biometrics} \right]$$

6 The Fuzzy Extractor

Sample-then-lock [CFP$^+$16] "encrypts" the same key multiple times using different subsets of w. We generate subsets globally in the Setup algorithm; our method of sampling subsets is our main technical contribution on fuzzy extractors.

Sample-then-lock uses digital lockers [CD08]. We first present the standard asymptotic definition of digital lockers and then discuss our assumptions of the meaning for concrete security. 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 $\mathsf{val}' = \mathsf{val}$. We say that $\mathsf{unlock}_{\mathsf{val},\mathsf{key}} \leftarrow \mathsf{lock}(\mathsf{val}, \mathsf{key})$ to describe producing the obfuscation. For correctness, it should be the case that $\mathsf{unlock}_{\mathsf{val},\mathsf{key}}$ is functionally equivalent to $I_{\mathsf{val},\mathsf{key}}$.



(a) Histogram of $p_{\texttt{same},i}$ and $p_{\texttt{diff},i}$ across i computed from $\texttt{stats}_W$ for Heterogeneous Feature Extractor.

(b) Value of $p_{\texttt{same},i} - p_{\texttt{diff},i}$ across i computed from $\texttt{stats}_W$ for Heterogeneous Feature Extractor.

(c) Histogram of $p_{\texttt{same},i}$ and $p_{\texttt{diff},i}$ across i computed from $\texttt{stats}_W$ for Angular Feature Extractor.

(d) Value of $p_{\texttt{same},i} - p_{\texttt{diff},i}$ across i computed from $\texttt{stats}_W$ for Angular Feature Extractor.

Figure 3: Different features have different error rates of $p_{\texttt{same},i}$ and $p_{\texttt{diff},i}$ and different gaps between these values. Figures 3a and 3b consider data computed from $\texttt{stats}_W$ for the heterogeneous feature extractor. Figures 3c and 3d use the angular feature extractor.

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* $\mathsf{key}, \mathsf{val}, \mathsf{val}' \neq \mathsf{val}$,

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

In the above, the probability is over the randomness of lock*.* ***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* $z \in \{0,1\}^*$,

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

The probability is over the randomness of A *and* S*.*

```
$\mathsf{Gen}(w, \mathtt{stats}_W = \mathcal{I}_1, ..., \mathcal{I}_\beta)$:
    1. Sample random 128 bit $\mathsf{Key}$.
    2. For $i = 1, ..., \beta$:
        (i) Choose 512 bit hash key $h_i$.
        (ii) Set $c_i = \mathtt{HMAC}(h_i, w_{\mathcal{I}_i})$.
        (iii) Set $p_i = (0^{128} || \mathsf{Key}) \oplus c_i$.
    3. Output $(\mathsf{Key}, p_i, h_i)$.

$\mathsf{Rep}(w', p_1, ..., p_\beta, h_1, ..., h_\beta, \mathtt{stats}_W = \mathcal{I}_1, ..., \mathcal{I}_\beta)$:
    1. For $i = 1, ..., \beta$:
        (i) Set $c_i = \mathtt{HMAC}(h_i, w'_{\mathcal{I}_i})$.
        (ii) If $(c_i \oplus p_i)_{1..128} = 0^{128}$ then
             output $(c_i \oplus p_i)_{129..256}$.
    2. Output $\perp$.
```

Figure 4: Adaption of sample-then-lock to use global subsets from $\mathtt{advise}_W = \mathcal{I}_1, ..., \mathcal{I}_\beta$.

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. The definitions are equivalent if there are a constant number of digital lockers or the same val is used [Can97, BC10, Var10, FF20, ACF$^+$22]. Digital lockers security is 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 $\mathtt{HMAC}$ be HMAC-SHA256. Our construction assumes that $\mathtt{HMAC}$ can be used to construct digital lockers. The "locking" algorithm outputs the pair $\mathsf{nonce}, \mathtt{HMAC}(\mathsf{nonce}, w) \oplus (0^{128} || \mathsf{Key})$, 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 shown to be secure in the random oracle model [BR93] by Lynn, Prabhakaran, and Sahai [LPS04, Section 4]. Standard model (without random oracles) hash functions may suffice [CD08, Section 3.2], [Dak09, Section 8.2.3].

6.1 Sample-then-lock Overview

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 4.

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:

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

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*

$$\nu := \left(\min_{1 \leq i \leq \beta} \{ \mathrm{H}_\infty (\mathsf{Val}_i | Z) \} \right).$$

Then for all A of size at most s the following holds:

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

6.1.1 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. Throughout this paper, we ignore the difference between an adversary with the real obfuscation and the simulator with an oracle. **We measure security by the quantity ν.**

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 $\tilde{\mathrm{H}}_\infty(\mathsf{Val}_i|Z_i) = \omega(\log n)$. While their theorem statement is correct, their proof has a bug and does not account for variation is the min-entropy of $\mathsf{Val}_i|Z_i$. Their proof assumes that each of these distributions has the same entropy as the average min-entropy. In particular, [CFP$^+$21, Lemma 2] is incorrect as stated. However, [CFP$^+$21, Theorem 1] is correct as one can bound the entropy drop by a fraction of ν with overwhelming probability (Lemma 2 in the upcoming proof). 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 $\mathsf{Val}_i|Z_i$. As mentioned, Zhu et al. [ZSC$^+$22] present an attack on Simhadri et al.'s system that targets the lowest entropy subset. 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*

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

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

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

Where probabilities are over randomness of S and U_κ, U'_κ and $\mathsf{val}_1, ..., \mathsf{val}_\beta, z \leftarrow \mathsf{Val}_1, ..., \mathsf{Val}_b, Z$.

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 $\delta > 0$,*

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

Equation 3 follows from Equation 2 by Application of Lemma 2 with $\delta = 2^{-\nu_{avg}/2}$. We focus on Equation 2.

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

```
$\zeta - \mathtt{Sample}(p_{\mathtt{same}}, p_{\mathtt{diff}}, \mathtt{SelMethod})$:
  1. For $i = 1$ to $\beta$:
     (a) For $j = 1$ to $n$:
         If $\mathtt{SelMethod} = \mathtt{LikeOnly}$: Set $p_j = p_{\mathtt{same},j}^{\zeta}$.
         Else if $\mathtt{SelMethod} = \mathtt{UnlikeRatio}$: Set $p_j = \left(\frac{p_{\mathtt{same},j}}{p_{\mathtt{diff},j}}\right)^{\zeta}$.
         Else if $\mathtt{SelMethod} = \mathtt{UnlikeExp}$: Set $p_j = (p_{\mathtt{same},j})^{\frac{\zeta}{-\log p_{\mathtt{diff},j}}}$.
     (b) Let $\mathcal{D}$ denote the probability distribution on $\{1, \ldots, n\}$ proportional $p_i / \sum_{j=1}^{n} p_j$.
     (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 5: ζ-Sampling.

followed by the first non-$\perp$ response, and one view has all q responses equal to $\perp$). By [DORS08, Lemma 2.2b], $\tilde{\mathrm{H}}_\infty(\mathsf{Val}_i | View(S), Z = z) \geq \tilde{\mathrm{H}}_\infty(\mathsf{Val}_j | Z = z) - \log(q+1) \geq \nu - \log(q+1)$. 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. Taking a union bound over all q queries the overall probability of a non-$\perp$ response is at most $q(q+1)/2^{\nu}$. □

7 ζ-Subset Selection

We now turn to ζ-sampling. For the `Heterogeneous` feature extractor, our goal is to select subsets for a sample-then-lock better than the uniform subset selection.

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). We consider three versions of the algorithm, one that uses only $p_{\mathtt{same},i}$, one that uses the ratio of $p_{\mathtt{same},i}/p_{\mathtt{diff},i}$ and one that uses $p_{\mathtt{same},i}^{1/\mathrm{H}_\infty(p_{\mathtt{diff},i})}$ denoted as `LikeOnly`, `UnlikeRatio`, and `UnlikeExp`. In `UnlikeRatio` the security and correctness of each bit are on the same scale, while in `UnlikeExp` variations in $p_{\mathtt{diff},i}$ are exponentially more important. All versions of the algorithms use a sampling characteristic parameter ζ and are shown in Figure 5.

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 not 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 $\mathtt{SelMethod} = \mathtt{LikeOnly}$ but we give intuition for the objective of ζ-sampling when $\mathtt{SelMethod} = \mathtt{UnlikeExp}$.

7.1 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 $p_i := 1 - p_{\mathtt{same},i}$. 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. 4.2 additionally weights selection by $p_{\mathtt{diff},i}$. This heuristic algorithm is the one we use in experiments in Section 8.

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

$$\Pr[\mathbf{A} \text{ requires more than } T \text{ steps to win}] \leq \epsilon_T \,. \tag{4}$$

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 (4) above.

7.1.1 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:

$$\begin{aligned} &S(x_1, \ldots, x_n) = \Pr[k \text{ distinct, good items are selected by } A^\zeta] \\ &\quad = \Pr[\text{all selected items are good}] \\ &\quad - \Pr[\text{selected items are good, repeat}] \\ &\quad \geq M(x_1, \ldots, x_n)^k - \binom{k}{2} \cdot C(x_1, \ldots, x_n) \cdot M(x_1, \ldots, x_n)^{k-2} \\ &\quad \geq M(x_1, \ldots, x_n)^k - k^2 \cdot C(x_1, \ldots, x_n) \cdot M(x_1, \ldots, x_n)^{k-2}\,. \end{aligned} \tag{5}$$

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 5. 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 5. We conclude that with high probability in the random variables X_i, the resulting quantity 5 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.

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

According to Chebyshev's inequality, we have

$$\begin{aligned} \Pr\left[M \leq \left(1 - \frac{1}{k}\right) \mathbb{E}(M)\right] &\leq \frac{\mathrm{Var}(M)}{(\frac{1}{k}\, \mathbb{E}(M))^2} \leq \\ &\frac{k^2 \frac{\sum_i p_i^{2\zeta+1}}{(\sum_i p_i^\zeta)^2}}{\left(\frac{\sum_i p_i^{\zeta+1}}{\sum_i p_i^\zeta}\right)^2} = k^2 \frac{\sum_i p_i^{2\zeta+1}}{(\sum_i p_i^{\zeta+1})^2}. \end{aligned} \tag{6}$$

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

$$\Pr\left[C \geq \frac{\mathbb{E}(M)^2}{8k^2}\right] \leq \frac{\mathbb{E}(C)}{\frac{\mathbb{E}(M)^2}{8k^2}} \leq \frac{8k^2\,\frac{\sum_i p_i^{2\zeta+1}}{(\sum_i p_i^{\zeta})^2}}{\left(\frac{\sum_i p_i^{\zeta+1}}{\sum_i p_i^{\zeta}}\right)^2} = 8k^2\,\frac{\sum_i p_i^{2\zeta+1}}{(\sum_i p_i^{\zeta+1})^2}. \tag{7}$$

Noting the similarity between the right-hand sides of 6 and 7, we define

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

The two inequalities (6) and (7) can then be written as

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

Combining these, we note

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

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 (5) as follows:

$$\begin{aligned} &\Pr[k \text{ distinct, good items are selected}]] \geq= M^{k-2}(M^2 - k^2C) \\ &> \left(1-\frac{1}{k}\right)^{k-2}\mathbb{E}(M)^{k-2}\left(\left(1-\frac{1}{k}\right)^2\mathbb{E}(M)^2 - \frac{\mathbb{E}(M)^2}{8}\right) \\ &= \left(1-\frac{1}{k}\right)^{k}\mathbb{E}(M)^{k} - \left(1-\frac{1}{k}\right)^{k-2}\frac{\mathbb{E}(M)^k}{8} \\ &\geq \frac{\left(1-\frac{1}{k}\right)^k}{2}\,\mathbb{E}(M)^k \geq \frac{\mathbb{E}(M)^k}{8}. \end{aligned} \tag{9}$$

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 $9k^2\Gamma$. □

7.1.2 Analyzing Entropy weighted ζ-sampling

The algorithm in Figure 5 with $\texttt{SelMethod} = \texttt{LikeOnly}$ 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 min-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 min entropy of this collection is $\sum_i \mathrm{H}_\infty(X_{q_i})$.

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

$$\frac{\log p_{q_i}}{\mathrm{H}_\infty(p_{\texttt{diff},q_i})}.$$

This is equivalent to maximizing

$$2^{\frac{\log p_{q_i}}{\mathrm{H}_\infty(p_{\texttt{diff},q_i})}} = p_{q_i}^{1/\mathrm{H}_\infty(p_{\texttt{diff},q_i})},$$

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 $w_i = p_i^{\zeta/\mathrm{H}_\infty(p_{\texttt{diff},i})}$ to each index $i \in [1, n]$.

To summarize, adjusting ζ in this family of algorithms determines the relative weight given to bits with larger values of $p_i^{1/\mathrm{H}_\infty(p_{\texttt{diff},i})}$. 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.

7.2 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

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

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.

8 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 best feature extractor (Section 5), and placed into a *sample-then-lock* fuzzy extractor with subsets sampled using the best sampling methodology (Section 6). We now use DSet to denote the test dataset described in Section 4.1. We explore two main questions.

1. The relative tradeoff between the Angular and Heterogeneous feature extractor for both uniform and ζ-sampling. We note that Angular features are strictly better than the cross-entropy and L_2 batch hard triplets. For efficiency reasons these tests consider a small number of subsets so the minimum of entropies is inaccurate. In addition, these test uses TARfast.
2. 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. In addition, these tests use TARfull. These subsets (along with the trained CNN) are the output of Setup [ADF24].

8.1 Parameter Finding

This subsection provides an overview of our three main tests: 1) comparing the angular margin and Heterogeneous feature extractors, 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.

8.1.1 Comparing Angular Margin and Heterogeneous

Our first test is to understand the relative tradeoff between the Angular and Heterogeneous feature extractors both for uniform and non-uniform sampling. We consider the following parameters: For this analysis we set the number of subsets to

$$\begin{aligned}
\beta &= 250\text{K}\\
k &\in \{60, 65, 70, 75, 80, 85, 90, 95, 100\},\\
\zeta &\in \{0, 0.5, 1, 2, 3, 5, 8, 10, 12, 15, 20\}\\
\texttt{SelMethod} &\in \{\texttt{LikeOnly}, \texttt{UnlikeRatio}, \texttt{UnlikeExp}\}.
\end{aligned}$$

For each of the above parameters 10 subsets are picked to assess entropy and then TARfast is run to estimate TAR. As a reminder, we report the minimum of assessed entropy using the EntTest from Section 4.2. Results are shown in a scatter plot in Figure 6. A few observations are apparent:

1. ζ-sampling largely does not work for the Angular feature extractor. This is shown in Table 1 which show the best results for a given entropy and TAR level. Here all methods of incorporating ζ provide nearly identical performance to uniform sampling.

2. ζ-sampling does allow the Heterogeneous feature extractor to explore more of the entropy versus TAR tradeoff space.

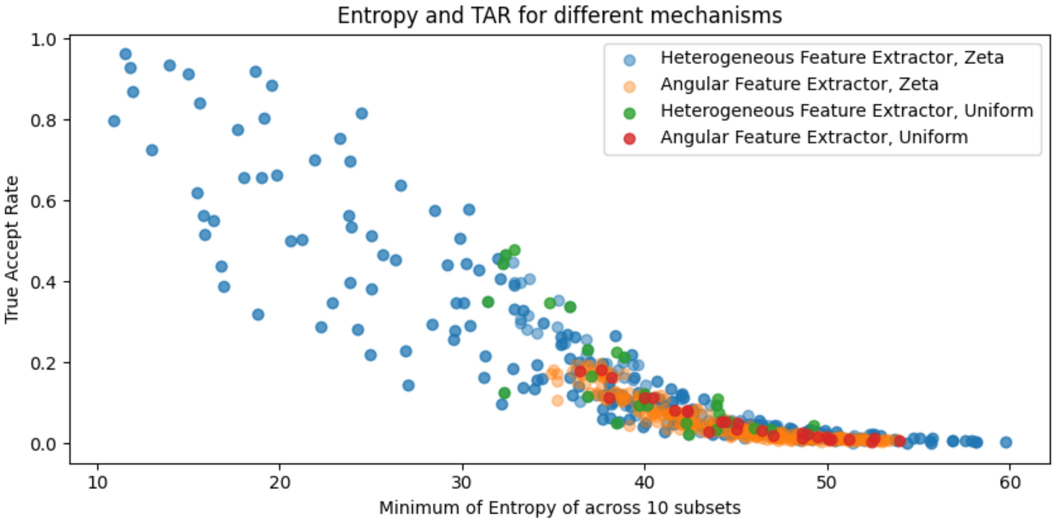


Figure 6: Scatter plot between entropy (across 10 runs) and TAR for different values of ζ and k. `LikeOnly`, `UnlikeRatio`, `UnlikeExp` are all shown together for each feature extractor.

3. It is not immediately obvious for the higher entropy and lower TAR regime if the Angular feature extractor or `Heterogeneous` feature extractor has better performance. We show the best TAR for a fixed entropy requirement for the `Heterogeneous` feature extractor in Table 2. The best entropy for a fixed TAR requirement is shown in Table 3. Cells are left blank if no tested parameters achieved the required entropy (or TAR).

4. Between these four tables a cell is colored blue if the TAR or entropy level is the best among the different `SelMethod` and the two feature extractors. While we see variation in the best `SelMethod` with `LikeOnly` performing best for high TAR targets, the `Heterogeneous` feature extractor always has the best entropy for a TAR requirement, and the best TAR for an entropy requirement. Based on these tests, we conclude that the `Heterogeneous` feature extractor in conjunction with ζ-sampling produces a better TAR versus entropy tradeoff.

In addition Tables 2 and 3 show a more complex sampling method for the `Heterogeneous` feature extractor that uses pairs of features. We note that this requires storage of $p_{\texttt{same},i,j}, p_{\texttt{diff},i,j}$ for all i, j which is 1M parameters in place of 2K parameters. Here we use ζ-sampling to pick pairs instead of features. Note that as in the case of individual features one or both features of a pair may already be included in a subset. This is handled by deduplicating and continuing until the required k is achieved. If pairwise correlation was present between features, one would expect ζ-sampling on pairs to improve performance over sampling based on individual features. However, as shown in Tables 2 and 3, at best, pairwise sampling the same performance as sampling on individual features. This provides further evidence that even if one makes `Heterogeneous` features modern neural networks are able to remove low-dimensional correlation between features.

Based on these experiments we consider four parameter regimes for deep dives:

`SelMethod`	k	ζ	min Ent	TAR
`UnlikeExp`	60	.5	33	.45
`UnlikeRatio`	65	10	42	.11
`UnlikeExp`	70	12	42	.11
`LikeOnly`	80	.5	42	.12

These parameters (and detailed results) are found in Table 4.

	LikeOnly			UnlikeRatio			UnlikeExp			Uniform	
Ent Level	k	ζ	TAR	k	ζ	TAR	k	ζ	TAR	k	TAR
≥ 20	60	15.0	0.20	60	0.5	0.20	60	2.0	0.19	60	0.16
≥ 25	60	15.0	0.20	60	0.5	0.20	60	2.0	0.19	60	0.16
≥ 30	60	15.0	0.20	60	0.5	0.20	60	2.0	0.19	60	0.16
≥ 35	60	15.0	0.20	60	0.5	0.20	60	2.0	0.19	60	0.16
≥ 40	65	0.5	0.11	70	0.5	0.08	65	2.0	0.12	65	0.11
≥ 42	70	1.0	0.08	70	2.0	0.08	70	1.0	0.09	70	0.08
≥ 44	75	1.0	0.06	75	2.0	0.05	75	0.5	0.06	75	0.05
≥ 46	85	8.0	0.03	85	0.5	0.03	80	3.0	0.03	85	0.02
≥ 48	85	0.5	0.02	85	0.5	0.03	85	0.5	0.03	90	0.02
≥ 50	95	2.0	0.02	90	1.0	0.02	90	2.0	0.02	100	0.01
≥ 52	100	5.0	0.01	90	8.0	0.01	100	5.0	0.01		
TAR Level	k	ζ	Ent	k	ζ	Ent	k	ζ	Ent	k	Ent
≥ 0.05	75	2.0	45	75	2.0	45	75	0.5	44	75	44
≥ 0.10	65	0.5	41	65	8.0	40	65	2.0	41	65	40
≥ 0.15	60	0.5	38	60	8.0	38	60	5.0	39	60	38

Table 1: Best TAR for each entropy level and best entropy for each TAR level. Angular feature extractor.

	Single Features											Pair Features								
	LikeOnly			UnlikeRatio			UnlikeExp			Uniform		LikeOnly			UnlikeRatio			UnlikeExp		
Ent	k	ζ	TAR	k	ζ	TAR	k	ζ	TAR	k	TAR	k	ζ	TAR	k	ζ	TAR	k	ζ	TAR
≥ 20	60	8	.82	60	.5	.43	60	.5	.45	60	.47	60	15	0.53	60	15	0.53	60	8	0.48
≥ 25	60	5	.64	60	.5	.43	60	.5	.45	60	.47	60	15	0.53	60	15	0.53	60	8	0.48
≥ 30	60	3	.57	60	.5	.43	60	.5	.45	60	.47	60	15	0.53	60	0.5	0.51	60	8	0.48
≥ 35	60	1	.26	65	.5	.27	60	8	.35	65	.34	70	20	0.26	70	10	0.26	65	2	0.34
≥ 40	80	.5	.12	70	2	.15	70	8	.14	95	4	80	2	0.12	75	20	0.19	85	5	0.10
≥ 42	80	.5	.12	65	10	.11	75	10	.09	95	.04	85	3	0.10	80	2	0.11	85	20	0.07
≥ 44	95	.5	.03	70	12	.06	80	5	.07	100	.03	85	0.5	0.07	90	20	0.06	90	0.5	0.06
≥ 46	100	1	.03	70	12	.06	90	1	.05			95	8.0	0.05	95	0.5	0.04	90	15	0.06
≥ 48	100	.5	.03	80	8	.04	85	15	.03						100	2.0	0.01	100	10	0.03
≥ 50				95	1	.03	90	10	.02									100	10	0.03
≥ 52				85	12	.02	90	10	.02											
≥ 54				85	12	.02														
≥ 56				100	20	.01														

Table 2: Best TAR for each entropy level. Heterogeneous feature extractor.

8.2 Detailed analysis of TAR vs. entropy

Our 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. Statistics are shown in Table 4. Note that the minimum of entropies, our security figure of merit, is as much as 9 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 there is an additionally inaccuracy in this value (compared to the full average min-entropy).

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. We may be able to exclude 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 both parameter sets, we compute the individual entropy of each sampled subset using EntTest. To understand the impact of excluding low min-entropy subsets, Table 4 shows the entropy of the 20% subset and a TAR test

	Single Features											Feature Pairs								
	`LikeOnly`			`UnlikeRatio`			`UnlikeExp`			Uniform		`LikeOnly`			`UnlikeRatio`			`UnlikeExp`		
TAR	k	ζ	Ent	k	ζ	Ent	k	ζ	Ent	k	Ent	k	ζ	Ent	k	ζ	Ent	k	ζ	Ent
$\geq .05$	80	.5	42	70	12	46	80	5	46	85	40	90	10.0	44	90	0.5	45	90	15.0	46
$\geq .10$	80	.5	42	70	8	43	70	12	42	70	37	85	3.0	43	80	2.0	42	80	0.5	40
$\geq .15$	75	2	36	72	2	41	65	12	40	70	37	75	0.5	39	75	20	40	75	1.0	39
$\geq .20$	75	2	36	65	2	39	70	1	39	70	37	70	20.0	38	70	10	37	75	15.0	37
$\geq .30$	65	.5	33	60	2	33	60	8	35	65	36	65	2	35	65	3	36	65	0.5	36
$\geq .40$	65	2	32	60	.5	31	60	1	34	60	32	60	0.5	33	60	3	33	60	0.5	33
$\geq .50$	60	3	30									60	15.0	31	60	0.5	33			
$\geq .60$	60	5	27																	
$\geq .70$	60	8	24																	
$\geq .80$	60	8	24																	
$\geq .90$	60	12	19																	

Table 3: Best entropy for each TAR level.

			Entropy Comparison						250K		High		High Subsets, TAR at # grouped					
Selection	k	ζ	Min	Med	Avg	Max	$\tilde{\mathrm{H}}_\infty$	1st 10	Ent	TAR	Ent	TAR	3	5	7	9	11	21
`UnlikeExp`	60	.5	25	35	35	42	34	35	25	.45	33	.41	.66	.76	.80	.83	.84	.87
`UnlikeRatio`	65	10	35	43	43	48	42	42	35	.13	42	.12	.21	.31	.34	.39	.38	.45
`UnlikeExp`	70	12	34	43	43	49	42	40	34	.13	41	.12	.23	.33	.37	.42	.42	.49
`LikeOnly`	80	.5	32	43	43	41	41	41	32	.13	41	.12	.23	.31	.38	.43	.43	.48

Table 4: Various entropy measurements across 250K subsets for all parameters chosen for deep dives. Simhadri et al. consider average min-entropy of first 10 subsets, denoted in the table as 1st 10. We consider the minimum of entropies for included subsets, excluding those with low entropy. For these parameters, the gap between these measures is as high as 9. We then restrict to the 200K highest entropy subsets, showing the Entropy gain and the TAR. Lastly, readings are grouped to recover TAR.

restricted to only the 200K highest entropy subsets. These are shown under the High column header. These subsets are included in our configuration [ADF24].

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, for ζ-sampling with the tested parameters TAR is a sublinear function in the number of subsets.

8.3 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]. In Table 4, we average the readings using $1, 3, 5, 7, 9, 11$, and 21 readings. This boosts TAR by as much as a factor of 4. The blue cell colors are the parameters chosen in the Introduction for `Thirty` and `Forty` parameters.

9 Discussion and Conclusion

This work presents `IrisLock`, an iris key derivation system that yields 42 bits of security at a 45% TAR or 33 bits of security at a 87% TAR. If one incorporates a password with an estimated entropy of 22 bits [KSK+11, Bon12, WZW+16], this would yield security estimates of 64 and 55 bits respectively. The sample-then-lock construction naturally supports prepending of a password. Our scheme drastically improves on the security and efficiency of prior work for iris key derivation. Our two technical contributions of a new feature extractor and new subset selection algorithm work together.

More work is needed to increase both security and TAR. A possibility is the use of *local confidence* [HRvD+16, DFR21], where one estimates the $p_{\mathtt{same},i}$ 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. The spread in entropy among subsets shows this will likely improve the adversary's task substantively.

Acknowledgements

The authors are grateful to the reviewers for their important comments in improving this work. In addition, the authors thank Alie Slade and Deep Inder Mohan for their helpful comments. 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 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 and robust fuzzy extractors in the plain model. In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022*, pages 353–383, Cham, 2022. Springer Nature Switzerland.

[ADF24] Sohaib Ahmad, Luke Demarest, and Benjamin Fuller. Computational fuzzy extractors, 2024. `https://github.com/benjaminfuller/Compfe`.

[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.

[ASW$^+$24] Sani M Abdullahi, Shuifa Sun, Beng Wang, Ning Wei, and Hongxia Wang. Biometric template attacks and recent protection mechanisms: A survey. *Information Fusion*, 103:102144, 2024.

[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.

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Message authentication using hash functions: The hmac construction. *RSA Laboratories' CryptoBytes*, 2(1):12–15, 1996.

[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.

[CHRF24] Chloe Cachet, Ariel Hamlin, Maryam Rezapour, and Benjamin Fuller. Upgrading fuzzy extractors. In *International Conference on Applied Cryptography and Network Security*, pages 156–182. Springer, 2024.

[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.

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *EUROCRYPT*, pages 523–540. Springer, 2004.

[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.

[Ful24] Benjamin Fuller. Impossibility of efficient information-theoretic fuzzy extraction. *Designs, Codes and Cryptography*, pages 1–27, 2024. `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.

[GKTF16] Zimu Guo, Nima Karimian, Mark M Tehranipoor, and Domenic Forte. Hardware security meets biometrics for the age of iot. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1318–1321. IEEE, 2016.

[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.

[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.

[HLVDMW17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[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.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 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.

[LC23] Kuo-Chun Lin and Yen-Ming Chen. A high-security-level iris cryptosystem based on fuzzy commitment and soft reliability extraction. *IEEE Transactions on Dependable and Secure Computing*, 2023.

[LNWS23] Song-Hong Lee, Cing-Ping Nien, Shun-Chi Wu, and A Lee Swindlehurst. Reconstruction attacks in template-based ecg biometric recognition systems. *IEEE Internet of Things Journal*, 2023.

[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/.

[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.

[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.

[WGCJ22] Kanishka P Wijewardena, Steven A Grosz, Kai Cao, and Anil K Jain. Fingerprint template invertibility: Minutiae vs. deep templates. *IEEE Transactions on Information Forensics and Security*, 18:744–757, 2022.

[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.

[WS09] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

[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.

[ZCY21] Kaiyi Zhang, Hongrui Cui, and Yu Yu. Facial template protection via lattice-based fuzzy extractors. Cryptology ePrint Archive, Paper 2021/1559, 2021. `https://eprint.iacr.org/2021/1559`.

[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.

[ZSC$^+$22] Feng Zhu, Peisong Shen, Kaini Chen, Yucheng Ma, and Chi Chen. A secure and practical sample-then-lock scheme for iris recognition. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 833–839. IEEE, 2022.