Fuzzy Extractors are Practical: Cryptographic Strength Key Derivation from the Iris

Amey Shukla
University of Connecticut
Storrs, Connecticut, United States
amey.shukla@uconn.edu

Luke Demarest
Gonzaga University
Spokane, Washington, United States
onlylukejohnson@gmail.com

Benjamin Fuller
University of Connecticut
Storrs, Connecticut, United States
benjamin.fuller@uconn.edu

Sohaib Ahmad
University of Connecticut
Storrs, Connecticut, United States
sohaib.ahmad@uconn.edu

Caleb Manicke
University of Connecticut
Storrs, Connecticut, United States
caleb.manicke@uconn.edu

Alexander Russell
University of Connecticut
Storrs, Connecticut, United States
acr@uconn.edu

Sixia Chen
Adelphi University
Garden City, New York, United States
schen@adelphi.edu

ABSTRACT

Despite decades of effort, a persistent chasm has existed between the theory and practice of device-level biometric authentication. Theoretical constructions can, in principle, provide biometric authentication with cryptographically secure public enrollment data. However, concrete implementations of these techniques have failed to provide security with real-world parameters. The result is that 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.

We close this chasm. We introduce a key derivation system with 105 bits of entropy and a 92% true accept rate (TAR) for the iris. Our system advances 1) the feature extraction from the iris and 2) the fuzzy extractor used to derive keys. The fuzzy extractor builds on sample-then-lock (Canetti et al., Journal of Cryptology 2021). We

(1) Introduce a new sampling method with a better trade-off between TAR and entropy when features have different quality,
(2) Correct Canetti et al.'s main security proof, showing the minimum of min-entropy over subsets is the relevant security measure, and
(3) Tighten Canetti et al.'s concrete analysis, nearly doubling security under reasonable assumptions.

Our final feature extractor incorporates ideas from the new sampling method to produce features optimized for the sample-then-lock construction.

The only statistical assumption needed to show security of our system is necessary: the accuracy of min-entropy estimation.

At 105 bits, our quantitative level of security is well above prior work. Simhadri et al. (ISC, 2019) report 32 bits on the iris, but they have a bug in their analysis that reduces their strength. Zhang et al.'s (ePrint 2021/1559) system achieves 45 bits on the face but assumes independence between biometrics and the used error-correcting code, an assumption that cannot be easily verified.

KEYWORDS

Biometric privacy; fuzzy extractors; key derivation

ACM Reference Format:
Amey Shukla, Luke Demarest, Benjamin Fuller, Sohaib Ahmad, Caleb Manicke, Alexander Russell, and Sixia Chen. 2025. Fuzzy Extractors are Practical: Cryptographic Strength Key Derivation from the Iris. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS '25), October 13–17, 2025, Taipei, Taiwan.* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3719027.3765098

1 INTRODUCTION

Biometric authentication is widely deployed. There is a longstanding 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 and there are devastating practical attacks in the event of an exposure [1–8]. 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 [9–11] (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* [12–26],[1] which offers security

[1]We do not review literature on interactive protocols [17, 27–34].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CCS '25, October 13–17, 2025, Taipei, Taiwan
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1525-9/2025/10…$15.00
https://doi.org/10.1145/3719027.3765098

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. However, these asymptotic techniques have never been successfully instantiated with the concrete parameters arising in real-world biometrics.

1.0.1 State of Prior Work. Since their introduction, fuzzy extractors have sufficed for concrete security if the "entropy rate" of the biometric is greater than the "error rate." This is easier to reason about if one assumes that bits of biometric W are i.i.d. [35–40]. Assume, for example, that $|W| = n = 1024$ and one wishes to correct μ fraction of errors to achieve a target true accept rate (TAR) ≈ 1. 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. Roughly, the construction needs to identify which of $\mu * n$ error patterns occurred, there are $2^{\texttt{err}}$ such patterns. Standard analysis assumes min-entropy reduces by the length of this string, which is at least $\texttt{err}$. The syndrome construction [14, 41] writes down exactly this many bits if a perfect code exists for the particular n, μ. The quantity $\texttt{FE}_{\texttt{Qual}} := \max\{0, \texttt{ent} - \texttt{err}\}$ lower bounds how many bits of security [19, Proposition 1] these constructions provide.

For the iris, prior feature extractors [42] achieve $\mu \geq .10$ and bits of different irises agree with probability .5. If one assumes W is i.i.d. then $\texttt{ent} = 1024$ and

$$\texttt{FE}_{\texttt{Qual}}(W) = 1024 - 480 = 544.$$

However, when one uses optimistic, heuristic statistical tests to estimate the entropy of W, the entropy is < 250 and $\texttt{FE}_{\texttt{Qual}}(W) = 0$.

All existing statistical analyses of biometrics conclude that bits of W are not i.i.d. [43]. A natural goal is to design feature extractors that produce independent features. Hine et al. [44] 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 [45, Figure 2].

1.1 Prior Work

Prior work that provides nonzero security for a biometric can be classified into two categories:

(1) **Assumption of Independence.** Typical prior work has assumed some independence of biometric features [44, 46, 47]. As mentioned above, Hine et al. [44] addresses pairwise correlation. For a family of error-correcting codes, Zhang et al. [47] assume that the nearest codeword and coset with respect to the code of a face feature vector are independent. Such assumptions appear difficult to statistically justify.
(2) **Incorrect Analysis.** As we discuss below, Simhadri et al. [45] build on a construction [48, Theorem 1] with an incorrect proof. Based on this incorrect proof, they estimate security level as the average min-entropy of subsets of features. Zhu et al. [49] present an attack on Simhadri et al.'s system that targets the lowest entropy subset (and hill climbs based on learned information).[3]
We correct the proof of [48, Theorem 1] in this work, showing that given an ideal digital locker the correct figure of merit is the minimum of the entropy of subsets of features used in the system.

1.1.1 Claimed Security. Simhadri et al.'s [45] claim a 32 bit key and a 60% TAR. Zhang et al's [47] work on the facial biometric and claim a 45 bit key with a 30% TAR.

1.2 The ClearEyes System

This work introduces ClearEyes, an integrated iris feature extractor and fuzzy extractor, translating iris readings into a stable key. In addition, we consider an amplified version of ClearEyes, which we call ClearEyes-M, that uses multiple iris readings for enrollment and authentication; this is a standard technique for augmenting TAR. ClearEyes enrolls and authenticates with 1 reading and has a TAR of 64%. ClearEyes-M uses 3 readings for enrollment and 2 for authentication and has a TAR of 92%. The two systems and Simhadri et al. are compared in Table 1.

Both systems provides concrete security of 105 *bits of security for the iris biometric.*

The only statistical assumption is the accuracy of min-entropy assessment [43] (described in Section 5.2.1). Polynomial time entropy assessment is inherently heuristic [50] but necessary. Our cryptographic assumption is the existence of composable digital lockers [51] which can be efficiently achieved using hash functions [52] and is a weaker assumption than modeling the hash function as a random oracle.

Despite using a stronger metric security than Simhadri et al., ClearEyes improves on their reported security by 73 bits of security while improving TAR from 60% to 64%. ClearEyes-M further improves TAR to 92%. We improve on Zhang et al.'s work by 60 bits of security and roughly the same improvement in TAR.

Guarantees Beyond a Standard Fuzzy Extractor. 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). Our system 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 [53]. 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. Furthermore, sample-then-lock is a reusable fuzzy extractor [54] allowing an individual to derive multiple keys from the same biometric. This is the reason we can enroll the same value multiple times for better TAR.

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

[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.

		System		
		Simhadri et al. [45]	ClearEyes	ClearEyes-M
System	Num stored lockers, β	1,000,000	200,000	600,000
	Subset size in each locker, k	43	115	115
	Readings at Enroll	1	1	3
	Readings at Auth	1	1	2
	Storage	96MB	7MB	19MB
Security	Entropy heuristic	Daugman Binomial fit [43]		
	Security metric	Avg. Min-Entropy over 10 Subsets	Min of Min-Entropy over all subsets	
	Security estimate	26	105	105
	$-\log$(est. FAR)	6	87	84
Enroll time	Total hashes	1,000,000	200,000	600,000
	Create subsets	412s	7s	22s
	Enroll biometric	204s	0.2s	0.5s
Auth time	Total hashes	1,000,000	200,000	1,200,000
	Time	17.3/0.8s	0.16s	0.98s
Correctness	Feature ext. mean error rate	0.32	0.08	0.08
	log(Est. TAR per locker)	-23.9	-12.6	-12.6
	TAR	0.60	0.64	0.92

Table 1: Summary of comparison with prior work. Estimated TAR assumes that all repeated readings have the same noise as the mean observed μ. The relationship between estimated TAR and observed TAR is highly dependent on the distribution of errors between readings of the same iris. Actual TAR and security estimates are taken from Table 5. For Auth, Simhadri et al.'s timing is reported both with their original and our faster implementation which tests a subset in $0.82\mu s$. All reported times are for all lockers. For our implementation authentication is faster on the dataset by a factor of 2 due to finding a match sooner.

1.2.1 Organization. Section 2 provides an overview, Section 3 introduces mathematical preliminaries, Section 4 describes the fuzzy extractor, Section 5 describes the datasets, Section 6 introduces our feature extractor, Section 7 evaluates, and Section 8 concludes.

2 SYSTEM OVERVIEW

ClearEyes is a combination of a feature extractor built from a convolutional neural network (CNN) and a fuzzy extractor built from the sample-then-lock fuzzy extractor [19, 48]. We briefly review the sample-then-lock fuzzy extractor [48], to give context for discussing our technical approach.

2.0.1 Sample-Then-Lock. For $n = 1024$,[4] let $W \in \{0,1\}^n$ be the random variable of creating a new biometric, collecting a reading, and processing the reading using a feature extractor. We use w to mean an outcome of W. We use $(w, w') \leftarrow (W, W')$ to indicate collecting two noisy readings from a single biometric representing enrollment and authentication. W and W' are correlated random variables. The *sample-then-lock* construction in enrollment:

(1) Uniform subsets $\mathcal{I}_1, \ldots, \mathcal{I}_\beta$ of $[0, n-1]$ are sampled, where each subset is of size k. In our final system, $k = 115$.
(2) A uniform value key is sampled.
(3) For a biometric $w \leftarrow W$, the string w restricted to each of the subsets $\mathcal{I}_i$ is used as the input value to a digital locker [51] with output value key.

A digital locker is a symmetric encryption that is secure when one creates ciphertexts with correlated keys that are computationally unpredictable [55]. (The formal definition is based on virtual black-box obfuscation [56], see Def. 3.)

[4]Our final transform reduces length to 512.

The system sets β different subsets as input to the digital locker with the same key as output in every locker. As we note in Lemma 1, subsets can be sampled from any distribution chosen independently from the enrollment value. The intuition for the construction is

(1) Make subsets as large as possible so that each subset is hard to guess, and
(2) Make β large enough so that for w, w' readings of the same biometric, there is some subset $\mathcal{I}_j$ where they match exactly: $w_{\mathcal{I}_j} = w'_{\mathcal{I}_j}$

The honest party and adversary face fundamentally different challenges when attempting to open a locker. Intuitively, a legitimate user can successfully open a single locker with probability $(1-\mu)^k$, where μ is the error rate of the feature extractor (chosen by examining the the error levels between W and W'). A satisfactory feature extractor will, in particular, provide sufficiently small error so that this task is feasible with a limited number of attempts (lockers). In contrast, an ideal feature extractor will generate uniformly random bits when applied to biometrics drawn uniformly from the full population; this corresponds to an effective error of $\mu_{Adv} = 1/2$. In fact, our security analysis shows that the adversary's ability to exploit feature correlation—which will be present in any practical feature extractor—does not significantly reduce the difficulty of this task. The bigger the gap between μ and μ_{Adv}, the smaller β can be while retaining both TAR and meaningful security. The interplay between the two parameters k and β depends on the feature extractor and the distribution used to sample each $\mathcal{I}_j$.

The minimum of min-entropy of subsets is the relevant security metric (see Assumption 1). The security analysis assumes the adversary completely recovers the biometric when it is able to open a digital locker.

As mentioned above, a standard design goal for feature extraction is approximately independent features. Current methods are capable of preventing small k-wise correlations between features. However, use of the uniform distribution for subset selection assumes that all features are of the same quality from the perspective of error rate and unpredictability.

When features have different quality, we introduce a method to bias subset selection to exploit the differences in feature quality and the correlations between features. Our subsets selection is non-uniform and depends on the observed distribution of W. We publish subsets which can be used globally for all users. (A user can also freely reproduce our methods with our open-source code.) 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.

2.1 Our contribution

Our technical contributions are the following:

(1) **Security Analysis.** We establish that global sampling of subsets for sample-then-lock is secure. We also fix the proof of sample-then-lock [48] that caused prior work to overestimate security. This shows that the minimum of min-entropy over subsets is the relevant security measure rather than average min-entropy over subsets. Lastly, we tighten analysis, nearly doubling key length under reasonable parameter settings.
(2) **Zeta-then-lock.** We design a non-uniform selection algorithm, called ζ-sampling, that samples high quality bits more frequently. We additionally consider variants of this procedure that account for unpredictability of features and prove bounds on sampling complexity in certain idealized settings. Finally, we establish a negative result showing that sampling pairs of features does not significantly improve our feature extractors.
(3) **Feature Extractor.** We develop two new feature extractors that produce superior features in comparison with prior work. Our final feature extractor incorporates ideas from ζ-sampling, producing features with nearly the same quality.
(4) **Parameter Analysis.** We carry out extensive parameter analysis, showing that `ClearEyes-M` yields 105 bits of entropy and a 92% TAR.

2.1.1 Security Analysis. We redefine a fuzzy extractor [57] 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. It gives advice to Gen and Rep denoted as $\texttt{stats}_W$, the subsets $\mathcal{I}_1, ..., \mathcal{I}_\beta$. There are two goals:

(1) **Correctness.** For repeated readings from the same biometric, $w, w' \leftarrow (W, W')$, for $(\text{key}, p) \leftarrow \text{Gen}(w, \texttt{stats}_W)$,

$$\Pr[\text{key} = \text{Rep}(w', p, \texttt{stats}_W)] \geq \text{desired TAR}.$$

In the above, w, w' are sampled from a pair of correlated random variables W, W' where W' is the "noisy" version of W. The probability is over sampling from (W, W'), the creation of $\texttt{stats}_W$, and the randomness of (Gen, Rep). This is a change from normal fuzzy extractor security where correctness is guaranteed for all w, w' that are close enough and there is no definition of a "noisy" random variable W'.
(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 estimate the entropy of each subset. These subsets can be used for any user. Provably accurate entropy estimation [50, 58] 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 [43] (detailed in Section 5.2.1). Let EntTest be an entropy test for biometric values for a dataset DSet. That is, e = EntTest(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 sampled subsets; in some of our tests, this number is 5 bits smaller than the average min-entropy of subsets (see Table 5). Our evaluation shows that *one can filter out low entropy subsets without sacrificing too much TAR*. The main reason we publish subsets is that we've already filtered out subsets with low entropy.

2.1.2 Parameter Analysis and Cryptographic Efficiency. We start with $\beta = 250,000$ subsets, since the adversary can target the lowest entropy subsets, we filter the 50,000 subsets with the lowest entropy. Simhadri et al. [45] used $\beta = 10^6$. When using multiple readings, we consider 3 enrollments for a total of 600,000 subsets. Publishing subsets removes most of the randomness and running time from Gen as one only needs to pick a random key and sample digital lockers. Our feature extractor, resulting statistics, chosen subsets, and code are open-sourced [59]. (The dataset is licensed and is not included in our repository.)

Simhadri et al. [45] reported a Gen time of 220s and a Rep time of .54s using 32 threads with a parallel implementation on a server machine with 4 Xeon E5-2620 v4 processors, for an overall single-threaded time of 17.3s. However, that implementation is 6 years old. We implement the scheme in C using HMAC from OpenSSL as the underlying hash function. For 200K lockers across 10 runs, Setup takes 7.12s (with a variance of .114), Gen takes .203s (with a variance in 3.3×10^{-5}), and Rep takes .161s (with a variance of 1.9×10^{-5}) on a single thread of a mobile Intel Core i9-13900H processor compiled using GCC. We note that Setup only needs to be run once. When using multiple readings, `ClearEyes-M` still has both enrollment and authentication under 1s on a single thread.

Table 1 summarizes the parameters used in our systems. In addition, we list Simhadri et al.'s system [45] for comparison with timing using our implementation. We exclude Zhang et al.'s system [47] as it is not based on sample-then-lock. We use the following estimates:

$$\text{FAR} := \text{Total Unlocks} * 2^{-\text{Sec. Estimate}},$$

$$\text{Est. TAR per locker} := (1 - \text{Feature Ext. Mean Error Rate})^k.$$

The dependence between Feature Ext. Mean Error Rate and k is exponential. By dropping Mean Error Rate from .32 to .08 we increase locker size from 43 to 115 while increasing TAR.

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

$$p_{\mathtt{same},i} := \Pr[w_i = w'_i | (w, w') \leftarrow (W, W')], \quad \text{and}$$
$$p_{\mathtt{diff},i} := \Pr[w_i \neq w^*_i | w \leftarrow W, w^* \leftarrow W].$$

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 when $p_{\mathtt{same},i}, p_{\mathtt{diff},i}$ vary across i and are stable across irises.

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

$$\mathtt{LikeOnly} = p^{\zeta}_{\mathtt{same},i},$$
$$\mathtt{UnlikeRatio} = \left(\frac{p_{\mathtt{same},i}}{\max\{p_{\mathtt{diff},i}, 1 - p_{\mathtt{diff},i}\}}\right)^{\zeta},$$
$$\mathtt{UnlikeExp} = p^{\zeta / H_\infty(p_{\mathtt{diff},i})}_{\mathtt{same},i}.$$

where $\mathrm{H}_\infty(p_{\mathtt{diff},i}) = -\log(\max\{p_{\mathtt{diff},i}, 1 - p_{\mathtt{diff},i}\})$. 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. For our tested feature extractors, sampling by pairs of features does not improve the entropy vs. TAR tradeoff (see Table 2).

2.1.4 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).

We present two feature extractors, a first version combining angular loss [60] and triplet mining [42] that produces different quality features. Our second version incorporates ideas from ζ-sampling to condense to features with the same quality. It uses marginal unified cross entropy [61] loss [62] and terms to directly consider the intra-class compactness (decreasing the distance between different readings of same iris) and inter-class separation (increasing the distance between readings of different irises). Our first version of the feature extractor produces features of different quality and benefits from ζ-sampling, while our second version produces very similar features and ζ-sampling provides marginal gains for parameters of interest.

2.2 Further Related Work

We focus on computational security due to negative results on providing information-theoretic security [57, 63–66]. Fuzzy min-entropy is the necessary metric for security of a fuzzy extractor [63, 65]. 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 [63, 65] are based on: 1) a new subset product assumption [67], 2) general-purpose obfuscation techniques [68–76], and 3) information theoretic techniques requiring exponential time [63, 77, 78]. The subset product assumption is directly the security of the proposed construction.

3 CRYPTOGRAPHIC PRELIMINARIES

For a set of indices J, X_J is the restriction of X to the indices in J. 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 *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 two distribution ensembles $X := \{X_\lambda\}_{\lambda \in \mathbb{N}}, Y := \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ and a distinguisher D, we write the *computational distance* with respect to D between X and Y as

$$\Delta^c_D(X, Y; \lambda) = \left|\Pr[D(X_\lambda, 1^\lambda) = 1] - \Pr[D(Y_\lambda, 1^\lambda) = 1]\right|.$$

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 [18], and we include a setup algorithm that is used globally (Fuller [57, Definition 8]).

Definition 2. *Let $n = n(\lambda)$ be a polynomial in the security parameter λ and let $\mathcal{W} = \{(W^*_\lambda)\}_{\lambda \in \mathbb{N}}$ be an ensemble of distributions, where each W^*_λ is a distribution on the set $\{0,1\}^{n(\lambda)} \times \{0,1\}^{n(\lambda)}$. A triple of randomized procedures "setup" (*Setup*), "generate," (*Gen*) and "reproduce" (*Rep*) is a $(\mathcal{W}, \kappa := \kappa(\lambda), \delta)$-*computational fuzzy extractor *if the triple satisfies the following properties:*

Correctness. *For any λ*

$$\Pr\left[\mathrm{Rep}(w', p) = \mathsf{key} \;\middle|\; \begin{array}{l} \mathtt{stats}_W \leftarrow \mathsf{Setup}(\lambda) \\ (w, w') \leftarrow W^*_\lambda \\ (\mathsf{key}, p) \leftarrow \mathsf{Gen}(w, \mathtt{stats}_W) \end{array}\right] \geq 1 - \delta.$$

Security. *For all PPT D and $\lambda > 0$, let*

$$\mathtt{stats}_W \leftarrow \mathsf{Setup}(\lambda),$$
$$(w, w') \leftarrow W^*_\lambda,$$
$$(\mathsf{Key}, P) \leftarrow \mathsf{Gen}(w, \mathtt{stats}_W);$$

then $|\mathsf{Key}| = \kappa(\lambda)$ *and*

$$\Delta^{c}_{D}((\mathsf{Key}, P, \mathtt{stats}_W), (U_{\kappa(\lambda)}, P, \mathtt{stats}_W); \lambda) \leq \mathtt{ngl}(\lambda).$$

3.0.1 Remarks. We do not formally tackle the notion of reusability [54], private [53], or robust fuzzy extractors [17] in this work. Reusable fuzzy extractors allow one to enroll noisy readings of a biometric multiple times. A private fuzzy extractor does not leak any adversarially specified function of the input (as long as it is independent of P). Sample-then-lock is private and reusable and the use of a global Setup does not change this as long as one uses a sufficiently composable digital locker. Upgrades to make fuzzy extractors robust are known in multiple cryptographic models [25, 26, 54].

4 AUGMENTING SAMPLE-THEN-LOCK

In this section we make three improvements to sample-then-lock:

(1) We correct an error in the proof of [48, Theorem 1], identifying the minimum of min-entropy of subsets as the quantity of interest,
(2) We tighten the concrete security analysis, doubling the key length in most circumstances, and
(3) We introduce a new sampling algorithm for the setting when features have different quality.

Our first two improvements hold for any method of sampling subsets. Zeta-then-lock [19] "encrypts" the same key multiple times using different subsets of the biometric. We generate subsets globally in the Setup algorithm.

The required notion of encryption is a digital locker [51]. We first present the standard asymptotic definition of digital lockers and then discuss our assumptions of the impact 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 [55]. Notationally, it is an obfuscation of the function

$$I_{\mathsf{val},\mathsf{key}}(\mathsf{val}') = \begin{cases} \mathsf{key} & \mathsf{val}' = \mathsf{val} \\ \perp & \text{otherwise.} \end{cases}$$

We use the notation $\mathsf{ulock}_{\mathsf{val},\mathsf{key}} \leftarrow \mathsf{lock}(\mathsf{val}, \mathsf{key})$ to describe the produced obfuscation.

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

$$\Pr[\mathsf{ulock}_{\mathsf{val},\mathsf{key}}(\mathsf{val}) = \mathsf{key} | \mathsf{ulock}_{\mathsf{val},\mathsf{key}} \leftarrow \mathsf{lock}(\mathsf{val}, \mathsf{key})] \geq 1 - \gamma,$$

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

The above 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 function q such that for any sufficiently large λ, any polynomially-long sequence of values* $(\mathsf{val}_i, \mathsf{key}_i)$ *for $i = 1, \ldots, \beta$, and any auxiliary input $z \in \{0,1\}^*$,*

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

The above probability is over the randomness of A and S, S is allowed $q(\lambda)$ queries to its oracles.

Definition 3 is known as virtual grey-box obfuscation (because the simulator is allowed to run in unbounded time). It implies distributional indistinguishability which says that obfuscations of all distributions val where $\mathrm{H}_{\infty}(\mathsf{val}) \geq \omega(\log \lambda)$ are indistinguishable from one another. The definitions are equivalent if there are a constant number of digital lockers or the same val is used [25, 79–82].

Definition 3 is asymptotic. A different simulator is allowed for each polynomial function p making it difficult to argue what quality key is provided with respect to a particular adversary.

4.0.1 Known Constructions of Digital Lockers. Let HMAC be HMAC-SHA256. Our construction assumes that HMAC can be used to construct digital lockers. The "locking" algorithm outputs the pair 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 [51, 83] and Learning with Errors [84, 85]. The HMAC construction used in this work construction was shown to be secure in the random oracle model [86] by Lynn, Prabhakaran, and Sahai [52, Section 4]. Standard model (without random oracles) hash functions may suffice [51, Section 3.2], [87, Section 8.2.3].

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

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

$\mathsf{Rep}(w', \{p_i, h_i\}_{i=1}^{\beta}, \mathtt{stats}_W = \mathcal{I}_1, ..., \mathcal{I}_\beta)$:

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

4.1 Sample-then-lock Security Analysis

Let β denote the number of subsets and assume that $\mathtt{stats}_W = \mathcal{I}_1, ..., \mathcal{I}_\beta$ and is input to Gen and Rep. Pseudocode for Gen and Rep is in Figure 1.

The parameters k (size of each subset) and β (number of subsets) represent a tradeoff between correctness and security. Canetti et al. [19, Section 4] note that rather than using independent subsets they could be selected using a sampler [88]. Simhadri et al. [45]

noted that each subset on its own needs to be random. For a particular output of Setup define the minimum of the subset min-entropies:

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

We measure the minimum of the min-entropies over subsets. As we discuss shortly, this quantifies the adversary success probability when given oracle access to the ideal object. Our assumption is that the adversary with the cryptographic object has the same success. *We measure security by the quantity ν.*

Assumption 1. *Let $\mathsf{Val}_1, ..., \mathsf{Val}_\beta, Z$ be correlated random variables and let U_κ, U'_κ be uniformly chosen. Let*

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

Then for all A that run in time at most s the following holds:

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

4.1.1 Fixing and tightening [48, Theorem 1]. As mentioned above, Definition 3 is an inherently asymptotic definition.

Canetti et al. [48, Theorem 1] focuses on the simulator's success probability (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 in 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, [48, Lemma 2] is incorrect as stated. However, [48, 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 across subsets in contrast to Simhadri et al. [45] who consider the average min-entropy of $\mathsf{Val}_i | Z_i$. As mentioned, Zhu et al. [49] present an attack on Simhadri et al.'s system that targets the lowest entropy subset. In addition, we tighten Canetti et al.'s analysis [48] of adversarial success from $q(q+1)/2^\nu$ to $q/2^\nu$. This is important for estimating practical security.

Lemma 1. *Let $\mathsf{Val}_1, ..., \mathsf{Val}_\beta, Z$ be correlated random variables and let U_κ, U'_κ be uniformly random variables that are independent of all other variables. For an outcome z, let*

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

Then for any outcome z and any S making at most q queries,

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

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

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

Where probabilities are over coins of S and $U_\kappa, U'_\kappa, \mathsf{Val}_1, ..., \mathsf{Val}_\beta, Z$.

Proof of Lemma 1. We restate a Lemma on the amount average min-entropy decreases across choices of b [14, Lemma 2.2b]:

Lemma 2. *Let A, B be random variables,*

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

Equation 3 follows from Equation 2 by application of Lemma 2 taking a union bound over all β. We focus on proving Equation 2.

Fix any $u, u' \in \{0,1\}^\kappa$ (the lemma will follow by averaging over all u, u'). The only information about whether the values u, u' can be 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'.

Also assume the algorithm S is deterministic, a deterministic algorithm can choose the best randomness of a randomized algorithm (as S runs in unbounded time). Let Q_i be the set of queries to the oracle $I_{\mathsf{val}_i, U_\kappa}$. Then the probability that

$$\Pr\left[\vee_{i=1}^{\beta} \mathsf{val}_i \in Q_i \right] \le \sum_{i=1}^{\beta} \Pr[\mathsf{val}_i \in Q_i] \le \sum_{i=1}^{\beta} |Q_i| 2^{-\nu} = q 2^{-\nu}.$$

Where the last step proceeds since there are q total queries. Since the above is true for all pairs u, u' it is clearly true for any distribution over u, u'. □

4.1.2 Comparison with original analysis. The original analysis of Canetti et al. both used average min-entropy and had a weak bound on the concrete security after q queries. In particular, they bounded adversary success as $q(q+1)/2^\nu$ meaning that for $\nu = 105$ one achieved probability 1 of success after $q = 2^{53}$ queries. On the other hand, our analysis presents success of 2^{-52} for these parameters.

4.2 ζ-Subset Selection

We now turn to ζ-sampling. Our goal is to select subsets for a sample-then-lock better that outperform uniform subset selection when features have different quality.

The heart of ζ-sampling is to use statistics about the relative quality of each bit to guide how subsets are formed. Informally, one seeks features that are unlikely to have errors in repeated readings of the same iris while still displaying high entropy (for the adversary to guess).

Recall the following notation:

$$p_{\mathtt{same},i} := \Pr[w_i = w'_i | (w, w') \leftarrow (W, W')] \text{ and}$$
$$p_{\mathtt{diff},i} := \Pr[w_i \ne w^*_i | w \leftarrow W \wedge w^* \leftarrow W].$$

We consider three versions of the algorithm:

`LikeOnly`. Uses only $p_{\mathtt{same},i}$,
`UnlikeRatio`. Uses the ratio of $p_{\mathtt{same},i}/p_{\mathtt{diff},i}$, and

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

UnlikeExp. Uses $p_{\texttt{same},i}^{1/\mathrm{H}_\infty(p_{\texttt{diff},i})}$.

In UnlikeRatio the security and correctness of each bit are on the same scale, while in UnlikeExp variations in $p_{\texttt{diff},i}$ are exponentially more important. All versions of the algorithms use a sampling characteristic parameter ζ and are shown in Figure 2.

An increase in ζ causes a sharper curve on the probability that a bit is selected based on its $p_{\texttt{same},i}$. 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 $p_{\texttt{same}}$. Our analysis focuses on the setting when SelMethod = LikeOnly but we give intuition for the objective of ζ-sampling when SelMethod = UnlikeExp.

4.3 Analysis of ζ-sampling

4.4 The Abstract Problem Description

To provide a theoretical justification and analysis of the proposed family of subset selection algorithms above, we formulate an idealized version of the problem that posits a family of independent "features," each of which can be correctly predicted with known probability $p_i := 1 - p_{\texttt{same},i}$. We then analyze the success probability of the algorithm that selects features with probability proportional to p_i^{ζ}, and succeeds when the features so selected are distinct and, furthermore, can be simultaneously predicted. Throughout our formal analysis we assume that all features are independent, which is usually not true in practice. Our actual implementation of the ζ-norm algorithms in Sec. 5.2.1 additionally weights selection by $p_{\texttt{diff},i}$. This heuristic algorithm is the one we use in experiments in Section 7.

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.

4.4.1 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

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

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

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

We call an index whose corresponding X_i is equal to 1 "good" and other indices "bad." We then focus on two quantities of interest, determined by the random variables X_i. For a fixed set of values

$x_1, \ldots, x_n$ taken by the random variables X_i, consider a single selection q of A^ζ (according to $\mathcal{D}$); then we define

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

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

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

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

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

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 quantity in Eq. 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

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

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

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

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:

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

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 $9k^2\Gamma$. □

4.5 Analyzing Entropy weighted ζ-sampling

The algorithm in Figure 2 with `SelMethod = 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 e while maximizing the probability of prediction, this calls for selecting bits that maximize the ratio

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

This is equivalent to maximizing

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

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 $w_i = p_i^{\zeta/\mathrm{H}_\infty(p_{\mathtt{diff},i})}$ 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 $p_i^{1/\mathrm{H}_\infty(p_{\mathtt{diff},i})}$. 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.

4.6 Simple estimates of the optimal ζ

We note that

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

and

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

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

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

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

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

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

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

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

5 DATASETS AND METRICS

5.1 Dataset and Feature Extractor Training

Throughout, we use the IIT Delhi iris dataset [89] comprising of age group 14-55 years with 176 males and 48 females. This dataset consists of 224 individuals with most classes containing images of both eyes representing 435 biometrics (224 left irises and 211 right irises). Left and right eyes are considered independent biometrics [90]. IIT Delhi is captured at a near-infrared wavelength. The dataset consists of 2240 images. Based on instructions from the dataset author, we preprocess the dataset to remove redundant and featureless images. We train on all left irises as in ThirdEye [42]. 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 images of the left irises from all individuals.
- $\mathtt{stats}_W$**.** 40 of the 202 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 162 right eyes are used for computing test data.

Training, calculation of $\texttt{stats}_W$*, and testing are all class disjoint.* For histograms shown in Figure 3, all images for each biometric were taken from the union of $\texttt{stats}_W$ and testing. The rest of tables and figures only use testing data. Images have a resolution of 320×240, and are segmented (iris portion separated from background) before input to the feature extractors. Segmentation is performed using the techniques of Hofbauer et al. [91].

5.1.1 Dataset Issues. Like many academic datasets IITD is not balanced. It is collected from a single geographic region, has more men than women, and is not well distributed among ages. Similar issues exist with the UBIRIS [92], CASIA [93] and ND0405 [94] datasets. Demographic biases in biometric datasets and feature extractors [95] create accuracy imbalances that matter. Before commercial deployment of our algorithms, we encourage collection of a balanced dataset and measuring security and correctness on demographic populations.

5.2 Metrics

5.2.1 Entropy Test. Throughout this work, we use the standard method of Daugman [43] 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 EntTest(DSet):

(1) Compute a histogram of all distances (fractional Hamming between the binary vectors) between readings of different biometrics (the red histogram in Figure 3).
(2) Find the mean μ_{Unlike} and stdev. σ_{Unlike} of this histogram.
(3) Compute degrees of freedom
$$\mathsf{dF} = \mu_{Unlike}(1-\mu_{Unlike})/\sigma^2_{Unlike}.$$
(4) Min-entropy is
$$\mathrm{e} = \min\{-\log(\mu_{Unlike}), -\log(1-\mu_{Unlike})\} * \mathsf{dF}.$$

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

This leads us to our first assumption which along with Assumption 1 suffices for the security of the scheme.

Assumption 2. *For a dataset* DSet *the test* EntTest *accurately measures the min-entropy of the distribution of biometrics from which* DSet *is drawn.*

5.2.2 Limitations of the entropy test. All entropy tests are heuristic [50]. The above entropy test only depends on the first two moments of the Unlike distribution ($\mu_{Unlike}, \sigma_{Unlike}$). We additionally fed one reading of each biometric into the NIST SP 800-22 entropy assessment [96] which always produced a higher estimate.

5.2.3 Computing TAR. All assessments of TAR take as input a collection of subsets $\mathcal{I}_1, ..., \mathcal{I}_\beta$. We compute the following:

(1) Set $\mathsf{TAR}_{\mathsf{num}} = 0, \mathsf{TAR}_{\mathsf{denom}} = 0$.
(2) For each class:
   (a) Pick the first biometric (randomly) 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 $\mathsf{TAR}_{\mathsf{denom}} = \mathsf{TAR}_{\mathsf{denom}} + \gamma$.
   (d) For $j = 1$ to γ: if exists some i such that $w_{j,\mathcal{I}_i} = w^*_{\mathcal{I}_i}$, $\mathsf{TAR}_{\mathsf{num}} = \mathsf{TAR}_{\mathsf{num}} + 1$.
(3) Output $\mathsf{TAR} = \mathsf{TAR}_{\mathsf{num}}/\mathsf{TAR}_{\mathsf{denom}}$.

The above computation 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.

6 THE TWO FEATURE EXTRACTORS

We now describe our two feature extractor pipelines. Iris recognition has a long history that we don't review due to space limitations [42, 90, 97–100]. All training is done on *L2* features, however histograms are presented in Hamming distance with features comparison to the feature mean indicating if a feature is 0 or 1.

6.1 VersionOne: Triplets and Angular Margin

Our first feature extractor builds on ThirdEye [42]. ThirdEye [42] outlines a two round training pipeline, starting with a model with pre-trained ImageNet weights. Unlike ThirdEye [42], which used a ResNet-50 architecture [101], we use a DenseNet-169 architecture [102]. Our design trains in two stages:

6.1.1 Cross-Entropy. 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.

6.1.2 Review of Triplet Mining. Each triplet is comprised of three samples, 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:
$$\mathsf{TL} := \sum_i \mathsf{TL}_i = \sum_i \begin{pmatrix} m + L_2(x_{a,i}, x_{p,i}) \\ -L_2(x_{a,i}, x_{n,i}) \end{pmatrix}$$

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. The system takes the "soft-margin" of the above loss, described by Hermans et al. [103].

Instead of optimizing the L_2 distance, Liu et al. [60] 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.

6.1.3 Combining Angular and L_2 distance. Our second stage combines L_2 and angle minimization from angular margin. The second stage of training trains the last 20 layers of the model (including the ending weights that were used for classification). For each batch, these current model output features are used to produce triplets that are used to compute a distance-based loss function.[5]

Our design combines the ideas of L_2 margin maximization from triplet loss with angle minimization from angular margin. We compute a modified triplet loss that includes additional weighting for our distance terms and the *inner product* between all examples:

$$TL_{\mathbf{V1}} = \sum_i \begin{pmatrix} m + c_1 * L_2(x_{a,i}, x_{p,i}) \\ -L_2(x_{a,i}, x_{n,i}) \\ +c_2 |\mu_{IP}| \end{pmatrix}.$$

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, the margin is $m = .4$. Here μ_{IP} represents the average inner product between all pairs of points of different irises. $TL_{\mathbf{V1}}$ 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.

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. As we discuss in Section 7, the VersionOne feature extractor produces features of different quality effective where ζ-sampling is effective at trading between TAR and entropy. it produces a roughly $\leq$ 10 security bit increase in entropy over Simhadri et al. [45]. Our second version increases security by 60 more bits.

6.2 VersionTwo: Unified Cross Entropy with Hard Negative Mining, and Refinement

The second version of our feature extractor is designed to produce higher entropy features while "sampling" bits that are likely to be without error. Our second version is built from a EfficientNet-B0 architecture [104] with pre-trained ImageNet weights.

6.2.1 EfficientNet Review and Modifications. The first round of our pipeline employs a modified EfficientNet-B0 [104] architecture. We modify the output of the network to produce a L2-normalized 1024-dimensional feature vector. The L2 normalization also helps stabilize the training process, preventing large magnitude features. We train our model in two stages.

6.2.2 First Stage: Unified Cross Entropy Loss with Hard Negative Mining. During the first stage, we train using a composite loss function that combines two components: the Marginal Unified Cross-Entropy Loss [62] and a hard negative mining loss. We describe each component in turn below.

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

The first component, Marginal Unified Cross-Entropy [62], extends marginal cross-entropy [61] by adding a constraint: the similarity score between samples of the same class must be greater, by some margin, than the similarity score between samples of different classes.It is denoted by $L_{\text{uce-m}}$.

The second component is a hard negative mining loss, a variant of triplet loss that focuses only on the most challenging examples to improve class separation. For each anchor sample x_a in a batch, the algorithm finds the negative sample x_{hn} (from a different class) that has the highest cosine similarity. The loss function is then calculated as:

$$\mathcal{L}_{\text{hnm}} = \frac{1}{B} \sum_{a=1}^{B} \max\left(0, \cos(x_a, x_{hn}) - m_{neg}\right)$$

where B is the batch size and m_{neg} is a margin hyperparameter. The final composite loss for this stage, $\mathcal{L}_{\text{Stage1}}$, is a weighted sum: $\mathcal{L}_{\text{Stage1}} = \mathcal{L}_{\text{uce-m}} + \lambda \cdot \mathcal{L}_{\text{hnm}}$ where $\lambda = 0.5$ is a hyperparameter balancing the two terms.

This stage augments each image at train time with a single, randomly selected transformation. The set of augmentations includes geometric transformations (horizontal and vertical flips and rotations at various angles) and appearance-based transformations (image sharpening, blurring, simulated occlusion, and contrast enhancement).

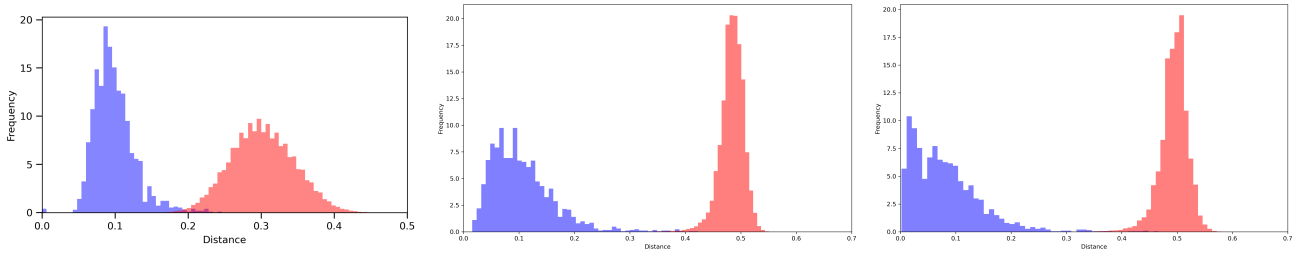
6.2.3 VersionTwoR*: Feature Refinement Inspired by ζ-sampling.* In the second stage of our pipeline, we introduce a lightweight feature refinement network inspired by ζ-sampling. The refinement network is an attention-based feed-forward neural network comprising Layer Normalization, two hidden layers with GELU activations and dropout, and a final linear layer to reduce the dimensionality to 512. A critical component of this network is a final tanh activation function, which maps the output features to the range [-1, 1]. This acts as a differentiable "soft binarization" of the features, which is essential for the subsequent loss calculation.

6.2.4 Soft Hamming Distance. The term "soft Hamming distance" refers to a differentiable proxy for the true (discrete) Hamming distance. Because the network's outputs from the tanh layer are continuous values in [-1, 1] rather than discrete bits, we cannot compute the standard Hamming distance directly during training, as the required binarization step is not differentiable. Instead, we use a proxy function based on cosine similarity that is differentiable and encourages the continuous outputs to move towards the binary extremes of -1 and 1. The soft Hamming distance, d_H, between two L2-normalized feature vectors x_i and x_j is calculated as:

$$d_H(x_i, x_j) = \frac{1}{2}(1 - \cos(x_i, x_j))$$

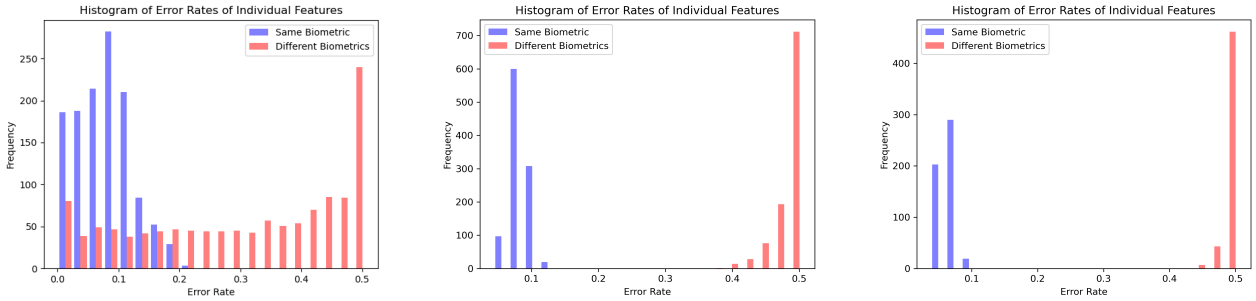
The total loss, $\mathcal{L}_{\text{Refinement}}$, is a weighted sum of two terms that separately consider positive (same-class) and negative (different-class) pairs: $\mathcal{L}_{\text{Refinement}} = \mathcal{L}_{\text{pos}} + \lambda_{\text{neg}} \cdot \mathcal{L}_{\text{neg}}$. The first term, $\mathcal{L}_{\text{pos}}$, is the mean of the soft Hamming distances for all positive pairs $\mathcal{P}$, pushing them towards zero:

$$\mathcal{L}_{\text{pos}} = \frac{1}{|\mathcal{P}|} \sum_{(i,j) \in \mathcal{P}} d_H(x_i, x_j)$$



(a) VersionOne. **(b)** VersionTwo. **(c)** VersionTwoR.

Figure 3: Distance comparisons for loss functions used in developing `ClearEyes`. Comparisons between readings of the same biometric are in blue. Comparisons between readings of different biometrics are in red. This figure combines data from $\texttt{stats}_W$ and testing.



(a) VersionOne **histogram of $p_{\texttt{same},i}$ and $p_{\texttt{diff},i}$ across i.** **(b)** VersionTwo **histogram of $p_{\texttt{same},i}$ and $p_{\texttt{diff},i}$ across i.** **(c)** VersionTwoR **histogram of $p_{\texttt{same},i}$ and $p_{\texttt{diff},i}$ across i.**

Figure 4: Different features have different error rates of $p_{\texttt{same},i}$ and $p_{\texttt{diff},i}$ and different gaps between these values. Figures 4a, 4b, and 4c consider data computed from all features for each stage feature extractor.

The second term, $\mathcal{L}_{\text{neg}}$, minimizes the mean squared error between the negative pair distances and a target distance, d_t (e.g., 0.5), for all negative pairs $\mathcal{N}$:

$$\mathcal{L}_{\text{neg}} = \frac{1}{|\mathcal{N}|} \sum_{(i,j)\in\mathcal{N}} (d_H(x_i, x_j) - d_t)^2$$

By optimizing this composite loss, the network learns to produce a compact, 512-dimensional binary embedding with low intra-class distance and a tightly clustered inter-class distance distribution centered around the target d_t.

During model inference we employ an ensemble approach that uses multiple augmentations of the input iris image, including horizontal flip, vertical flip, and small rotations of ±5 degrees.

All these augmentations are processed through the feature extraction pipeline and are combined using a bit-wise voting scheme where we select the most stable bit for each position.

This results in feature embeddings with tighter inter-class variance and consistent performance across various image quality conditions. We stress the small variation in features in Figure 3.

6.3 Computing $\texttt{stats}_W$

VersionOne feature extractor and VersionTwo (without refinement) produce binary features of length 1024 while VersionTwoR produces features of length 512. Our $\texttt{stats}_W$ consist of subsets $\mathcal{I}_1, ..., \mathcal{I}_\beta$. We use two real values for each feature from $[0, 1]$ that represent the "quality" of features. For each feature i we compute:

$$p_{\texttt{same},i} := \Pr[w_i \neq w_i' | (w, w') \leftarrow (W, W')]$$

$$p_{\texttt{diff},i} := \Pr\left[\max\begin{Bmatrix} w_i \neq w_i^* \\ w_i = w_i^* \end{Bmatrix} \middle| w \leftarrow W \wedge w^* \leftarrow W\right]$$

These probabilities are computed across the $\texttt{stats}_W$ dataset. We show the histogram of $p_{\texttt{same},i}$ and $p_{\texttt{diff},i}$ in Figure 3.

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

$$p_{\texttt{same},i,j} := \Pr[w_i = w_i' \wedge w_j = w_j' | (w, w') \leftarrow (W, W')]$$

$$p_{\texttt{diff},i,j} := \max_c \Pr\left[\begin{matrix} w_i \oplus w_i^* || \\ w_j \oplus w_j^* \end{matrix} = c \middle| w \leftarrow W \wedge w^* \leftarrow W\right]$$

Note the difference in feature quality between Figure 4a and Figure 4c. Since we apply zeta-sampling internally during the refinement stage (described in 6.2.3) for Figure 4c, external zeta sampling shows no effect compared to Figure 4a, as we demonstrate later in Figure 7.

7 EVALUATION

Our primary goal is to evaluate the TAR versus entropy tradeoff of a full system using irises processed by VersionOne and VersionTwoR

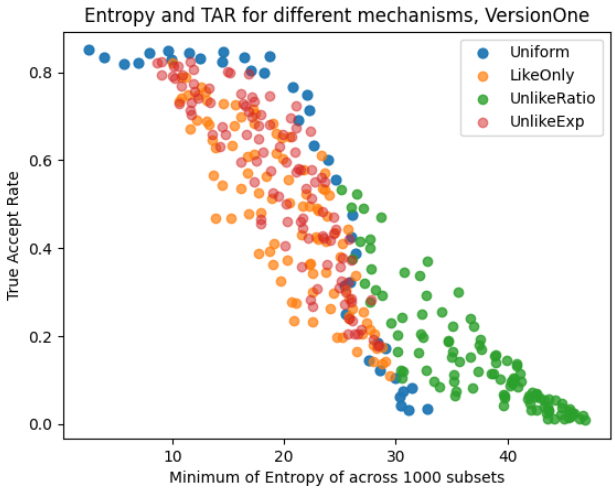


Figure 5: Scatter plot between entropy (across 1000 runs) and TAR for different values of ζ and k for VersionOne.

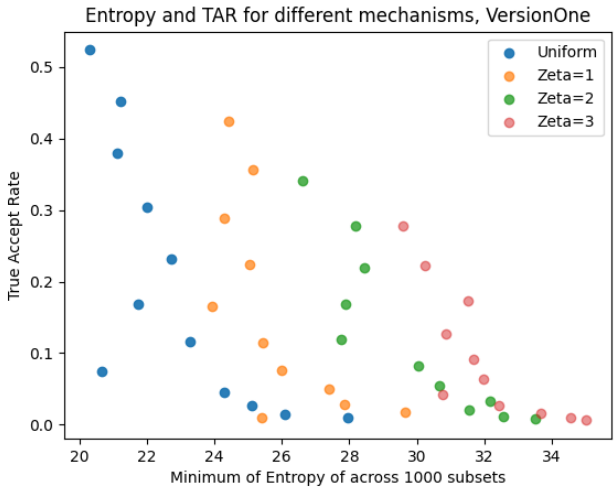


Figure 6: Change in TAR and Entropy as β increases from 2^6 to 2^{17}. ζ-sampling is able to retain entropy better as β increases. VersionOne of feature extractor. ζ sampling uses `UnlikeRatio`.

(Section 6), and placed into a *sample-then-lock* fuzzy extractor with subsets sampled using the best sampling methodology (Section 4). We use DSet to denote the test dataset. We explore three questions.

(1) The value of ζ sampling versus uniform sampling.
(2) The TAR versus entropy tradeoff.
(3) For promising parameters, a detailed investigation into the minimum of entropies across all subsets that would be used in practice. We publish the analyzed subsets as part of this work. These subsets (along with the trained feature extractors) are the output of Setup [59].

We report the minimum of assessed entropy using the `EntTest` from Section 5.2.1. When we choose promising parameters where we perform the entropy test over all 250,000 subsets. We consider k from 40 to 140 in increments of 5 and $\zeta \in \{0, .5, 1, 2, 3, 5, 8, 10, 15, 20\}$. Recall that $\zeta = 0$ corresponds to uniform sampling.

Figure 5 shows the TAR versus entropy tradeoff for VersionOne feature extractor. Uniform sampling is in Blue. `UnlikeRatio` is able to produce higher entropy features at similar TAR rates as uniform sampling. This is also shown in Table 2. Where `UnlikeRatio` can sample larger subsets (and thus achieve more entropy) for comparable TAR to uniform sampling. `UnlikeRatio` provides superior performance in comparison to `UnlikeExp` while `LikeOnly` is effective at producing high TAR but sacrifices entropy. ζ sampling provides a more drastic impact when the number of subsets, β, is small. To show this Figure 6 shows how TAR improves as β increases. It considers β increasing by powers of 2 from 2^6 to 2^{17}. Since uniform sampling is choosing random subsets, as β improves, one encounters lower entropy subsets. ζ-sampling better retains entropy, however the TAR improvement slows as it is picking more correlated subsets. As we describe in Section 4.6, when ζ sampling can reach a TAR value it reaches the TAR value with fewer subsets than smaller ζ values. However, achieving the largest TARs may require small ζ to ensure subset diversity.

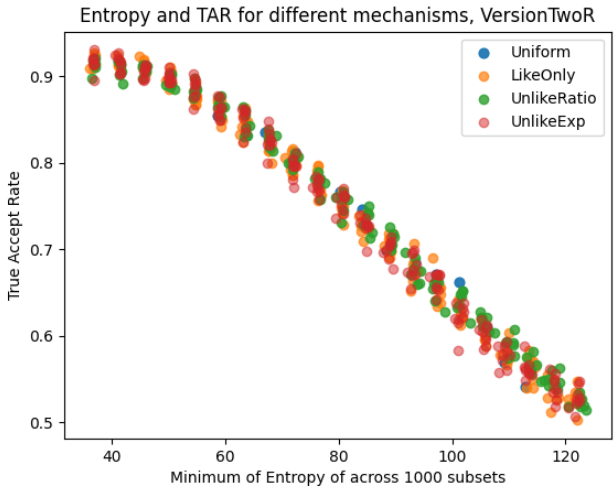


Figure 7: Scatter plot between entropy (across 1000 runs) and TAR for different values of ζ and k. VersionTwoR of Feature Extractor. Axes have different scales than Figure 5.

For the VersionTwoR feature extractor ζ sampling displays a much smaller improvement. This is shown in Figure 7 and Table 3. VersionTwoR was designed to ζ-sample internally so it produces features with the same quality. We note that while ζ-sampling is less effective, the entropy versus TAR tradeoff is vastly improved by VersionTwoR.

To understand if features are the same quality, we computed Pearson correlation between $p_{\texttt{same},i}$ on $\texttt{stats}_W$ and on the test dataset, it was only .14. In comparison, for the VersionOne feature

	Single Features											Pairs of 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	105	3	0.72	105	0.5	0.57	110	5	0.65	105	0.64	105	8	0.7	105	20	0.69	105	20	0.7
≥ 22	105	1	0.67	105	0.5	0.57	105	1	0.61	110	0.6	105	8	0.7	105	5	0.67	105	20	0.7
≥ 24	120	0.5	0.47	105	0.5	0.57	115	1	0.52	120	0.51	105	2	0.64	110	5	0.61	105	8	0.65
≥ 26	130	0.5	0.37	105	1	0.52	130	0.5	0.34	140	0.29	125	15	0.44	115	1	0.51	120	8	0.49
≥ 28	160	1	0.18	105	2	0.47	155	2	0.21	150	0.22	135	0.5	0.28	140	5	0.24	135	15	0.31
≥ 30				105	3	0.37				160	0.13	155	5	0.12	160	3	0.13	160	5	0.13
≥ 40				115	10	0.14														

Table 2: VersionOne **Feature Extractor. Best TAR for each entropy level.**

extractor Pearson correlation is .83. This explains the dramatically different quality of ζ-sampling.

Tables 2 and 3 show the highest achievable TAR for desired entropy levels for VersionOne and VersionTwoR respectively. We do not separate out uniform sampling for VersionTwoR as its performance is similar to ζ sampling. For the VersionTwoR the sampling method has little impact on TAR. However, the sampling method has a dramatic difference for VersionOne. Even for VersionOne, sampling by pairs does not improve over using single bits, indicating that correlation between features is higher dimensional. Sampling pairs 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.[6]

Ent	LikeOnly			UnlikeRatio			UnlikeExp		
Level	k	ζ	TAR	k	ζ	TAR	k	ζ	TAR
≥ 20	40	0.5	0.93	40	3.0	0.92	40	8.0	0.93
≥ 30	40	0.5	0.93	40	3.0	0.92	40	8.0	0.93
≥ 40	45	0	0.92	45	0	0.92	45	8.0	0.93
≥ 50	55	0.5	0.91	55	1.0	0.90	55	10.0	0.91
≥ 60	70	8.0	0.86	70	2.0	0.87	70	3.0	0.86
≥ 70	80	3.0	0.82	80	3.0	0.81	80	0	0.81
≥ 80	90	8.0	0.77	90	1.0	0.77	90	2.0	0.77
≥ 85	100	0	0.73	95	5.0	0.75	95	15.0	0.73
≥ 90	105	0.5	0.71	105	2.0	0.70	105	5.0	0.70
≥ 95	110	8.0	0.69	110	0	0.67	110	0	0.67
≥ 100	115	2.0	0.64	115	3.0	0.65	115	0	0.66

Table 3: VersionTwoR **Feature Extractor**

We consider four parameter regimes for deep dives all from the VersionTwoR feature extractor. These parameters (and detailed results) are found in Table 5.

7.1 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 understand the variation in entropy and choose to exclude low-entropy subsets. Entropy statistics are shown in Table 4. Note that the minimum of entropies, our security figure of merit, is as much as 5 bits lower than the average min-entropy across subsets which was used as a figure of merit in Simhadri et al. [45]. Simhadri et al. estimated entropy using only 10 subsets.

[6] 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.

					All 250K		All 200K		Obs.
k	ζ	Min	Avg	Max	Ent	FAR	Ent	FAR	FAR
70	3	62	66	70	62	41	65	44	0
90	8	79	84	88	79	58	83	62	0
110	3	95	101	107	95	74	100	79	0
115	2	100	106	112	100	79	105	84	0

Table 4: All methods use UnlikeExp **selection. Various entropy measurements across** 250**K subsets for all parameters chosen for deep dives. Average is the mean of entropies. 1000 represents the average min-entropy of the first 1000 sampled subsets. We consider the minimum of entropies for included subsets, excluding those with low entropy. Estimated FAR is reported as** $-\log$**, computed as Ent**-20.5 **as there are** $2^{20.5}$ **total lockers used in final parameters for** ClearEyes-M**. Observed FAR is computed on all** 250**K lockers.**

				250K		200K	
k	ζ	Enrolls	Auths	Ent.	TAR	Ent.	TAR
70	3	1	1	62	.85	65	.84
		1	4		.97		.97
		2	3		1.00		1.00
		3	2		.99		.99
90	8	1	1	79	.77	83	.74
		1	4		.94		.93
		2	3		.98		.98
		3	2		.99		.98
110	3	1	1	95	.67	100	.64
		1	4		.85		.84
		2	3		.93		.93
		3	2		.94		.92
115	2	1	1	100	.65	105	.64
		1	4		.84		.82
		2	3		.92		.91
		3	2		.93		.92

Table 5: TAR for various parameters, either all subsets or the highest entropy 200**K subsets.**

7.1.1 Recovering Average-Case Behavior. As described above, as needed one can remove 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. For our final chosen

parameters of $k = 115$, $\zeta = 2$, removing the lowest 50K entropy subsets increases entropy by 5 bits without reducing TAR, this reduces TAR by .01 (see Table 5). Across other parameter settings we see small decreases in TAR when removing the lowest entropy 50K subsets but it is consistently less TAR reduction than one might expect if those subsets were responsible for a disproportionate amount of TAR.

7.2 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 [105–107]. We adapt this concept by assuming there are five total readings to be split between enrollment and authentication. In Table 5, we enroll multiple distinct readings (1, 2, or 3) and attempt to authenticate with multiple readings (4, 3, or 2 respectively). Authentication is deemed successful if any authentication reading matches any enrollment reading. This boosts TAR by as much as .28. This technique appears equally effective when considering all 250K subsets or just the highest entropy 200K subsets. As a result, we chose the 200K highest entropy subsets with entropy of 105 and 92% TAR as our final parameters. These cell colors are presented in blue. We note that when enrolling 3 readings and authenticating with two readings this increases running time of Gen by factor of 3 and Rep by a factor of 6 as discussed in Section 2.1.2.

8 CONCLUSION

This work presents `ClearEyes` and `ClearEyes-M`, an iris key derivation system that yields 105 bits of security with TARs of 64% and 92% respectively. In addition, we can achieve the required 95% TAR for meeting the NIST biometric authentication guidelines [108] and the required 97% TAR for meeting the FIDO alliance biometric authentication guidelines [109] with 83 bits of entropy (second row in Table 5). Our scheme improves on the security and efficiency of prior work for iris key derivation.

Natural lines of future research include extending our results to different biometrics and understanding how the noise/entropy tradeoff is impacted by real-life use. In addition, we believe that ζ-sampling could find applications in fuzzy extractors for physically uncloneable functions where confidence information is used, see for example Herder et al. [110].

ACKNOWLEDGEMENTS

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

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

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

[19] 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.

[20] 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.

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

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

[23] 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.

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

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

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

[36] 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.

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] Jeroen Delvaux, Dawu Gu, Ingrid Verbauwhede, Matthias Hiller, and Meng-Day Yu. Efficient fuzzy extraction of puf-induced secrets: Theory and applications. In *International Conference on Cryptographic Hardware and Embedded Systems*, pages 412–431. Springer, 2016.

[42] 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.

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

[44] 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.

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

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] Ran Canetti and Ronny Ramzi Dakdouk. Obfuscating point functions with multibit output. In *Advances in Cryptology–EUROCRYPT 2008*, pages 489–508. Springer, 2008.

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

[53] 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.

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

[55] 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.

[56] 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.

[57] Benjamin Fuller. Impossibility of efficient information-theoretic fuzzy extraction. *Designs, Codes and Cryptography*, pages 1–27, 2024. https://eprint.iacr.org/2023/172.

[58] 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.

[59] Amey Shukla, Sohaib Ahmad, Luke Demarest, and Benjamin Fuller. Fuzzy extractors are practical, 2025. https://github.com/whyamey/feap/.

[60] 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.

[61] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018.

[62] Jiancan Zhou, Xi Jia, Qiufu Li, Linlin Shen, and Jinming Duan. Uniface: Unified cross-entropy loss for deep face recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20730–20739, 2023.

[63] 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.

[64] 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.

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

[66] Benjamin Fuller. Impossibility of efficient information-theoretic fuzzy extraction. *Designs, Codes and Cryptography*, 92(7):1983–2009, 2024.

[67] 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.

[68] 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.

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

[70] 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/.

[71] 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.

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

[73] 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.

[74] 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.

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

[76] 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.

[77] 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.

[78] 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.

[79] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In *Advances in Cryptology-—CRYPTO'97*, pages 455–469. Springer, 1997.

[80] Nir Bitansky and Ran Canetti. On strong simulation and composable point obfuscation. In *Advances in Cryptology–CRYPTO 2010*, pages 520–537. Springer, 2010.

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

[82] 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.

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

[84] 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.

[85] 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.

[86] 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.

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

[88] 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.

[89] Ajay Kumar and Arun Passi. Comparison and combination of iris matchers for reliable personal authentication. *Pattern recognition*, 43(3):1016–1026, 2010.

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

[91] Heinz Hofbauer, Fernando Alonso-Fernandez, Peter Wild, Josef Bigun, and Andreas Uhl. A ground truth for iris segmentation. In *2014 22nd international conference on pattern recognition*, pages 527–532. IEEE, 2014.

[92] H. Proenca, S. Filipe, R. Santos, J. Oliveira, and L.A. Alexandre. The UBIRIS.v2: A database of visible wavelength images captured on-the-move and at-a-distance. *IEEE Trans. PAMI*, 32(8):1529–1535, August 2010.

[93] Tieniu Tan and Zhenan Sun. Casia iris v4 interval. http://biometrics.idealtest.org/.

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

[95] Pawel Drozdowski, Christian Rathgeb, Antitza Dantcheva, Naser Damer, and Christoph Busch. Demographic bias in biometrics: A survey on an emerging challenge. *IEEE Transactions on Technology and Society*, 1(2):89–103, 2020.

[96] Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, et al. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*, volume 22. US Department of Commerce, Technology Administration, National Institute of …, 2001.

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

[98] J. Daugman. Probing the uniqueness and randomness of iriscodes: Results from 200 billion iris pair comparisons. *Proceedings of the IEEE*, 94(11):1927–1935, 2006.

[99] Maria De Marsico, Alfredo Petrosino, and Stefano Ricciardi. Iris recognition through machine learning techniques: A survey. *Pattern Recognition Letters*, 82:106–115, 2016.

[100] Yifeng Chen, Cheng Wu, and Yiming Wang. T-center: A novel feature extraction approach towards large-scale iris recognition. *IEEE Access*, 8:32365–32375, 2020.

[101] 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.

[102] 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.

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

[104] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[105] 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.

[106] 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.

[107] 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.

[108] Paul A Grassi, Michael E Garcia, and James L Fenton. NIST special publication 800-63-4 digital identity guidelines. *National Institute of Standards and Technology, Los Altos, CA*, 2024.

[109] Stephanie Schuckers, Greg Cannon, Nils Tekampe, Elham Tabassi, Meagan Karlsson, and Elaine Newton. Fido biometrics requirements. *Population*, 5(2-1):2–3, 2019.

[110] 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.