Cryptographic Authentication from the Iris

Sailesh Simhadri, James Steel, and Benjamin Fuller

University of Connecticut
Storrs, CT 06269
Email: first.last@uconn.edu

November 14, 2018

Abstract

Biometrics exhibit noise between repeated readings. Due to the noise, devices store a plaintext *template* of the biometric. This stored template is an appetizing target for an attacker. Due to this risk, the primary use case for biometrics is mobile device authentication (templates are stored within the mobile device's secure processor). There has been little adoption in client-server applications.

Fuzzy extractors derive a stable cryptographic key from biometrics (Dodis et al., Eurocrypt 2004). In this work we describe an iris key derivation system with 32 bits of security even when multiple keys are derived from the same iris.

We are fully aware that 32 bits of security is insufficient for a secure system. The goal of this work is to inspire researchers to design multi-factor authentication systems that uses our scheme as one component. Our system is based on repeated hashing which simplifies incorporating multiple factors (such as a password).

Our starting point a recent fuzzy extractor due to Canetti et al. (Eurocrypt 2016). Achieving satisfactory parameters requires modifying and coupling the image processing and cryptographic algorithms. Our scheme is implemented in C and Python and is open-sourced. On a moderately powerful server, authentication usually completes within .30s.

1 Introduction

Authentication schemes combine factors such as passwords, one-time codes, security questions, and social relationships [BHVOS12]. Since service providers are frequently compromised, salted hashes of the factors are stored at providers (instead of the plaintext value). Some providers use key derivation functions to derive cryptographic keys and then protect sensitive data using these keys. The goal of these protections is to limit an adversary that compromises the server to only verifying a guess of the authentication factors. Depending on the *entropy* of the factors, we can obtain bounds on how long it will take an adversary to correctly guess users' private information.

Biometrics are used as authentication tokens on mobile devices (phones and tablets). In these systems, a template of the biometric reading is stored in a secure processor. When the user attempts to authenticate, a new reading is collected. The secure processor checks if the new reading is close enough to the stored template. The secure processing unit then bootstraps the operating system, allowing the user to unlock their phone. Since the template is stored "in the clear," a secure processing component is necessary. This increases cost. Furthermore, it means that deploying biometric authentication in a client-server settings is risky. The client-server setting is still the majority of Internet authentication.

Two complementary lines of research emerged to mitigate this risk: interactive protocols and schemes that create a single value that allows for authentication (that is, non-interactive protocols). The interactive setting is well understood with secure solutions for a variety of biometrics using a variety of cryptographic techniques [BDK+05, DKRS06, BG11, EHKM11, DKK+12, BCP13, BDCG13, DCH+16, DHP+18]. The main goal in this line of research is building efficient enough protocols for deployment. Importantly, interactive protocols do not consider server

compromise in scope of the threat model. Their focus is on ensuring an adversary that pretends to be either the client or server gains minimal information by engaging in the protocol.

The non-interactive setting is not understood despite years of research. For many physical distributions with noise on repeated readings, the only protocols that are known to be secure run in exponential time or whose security relies on semantically secure graded encodings [BCKP14, BCKP17]. The protocols that run in exponential time require a full description of probability distribution, making it unlikely they can be made efficient [HR05, FRS16, WCD+17]. On the other hand, semantically secure graded encodings [PST13] are closely related to indistinguishability obfuscation which uses heavy cryptographic machinery [GGH+13b, GGH13a]. The security of indistinguishability obfuscation is a matter of vigorous debate [CHL+15, MSZ16, GPSZ17, MZ17].

The focus of this work is building non-interactive and efficient authentication from biometrics. Specifically, we focus on the iris due its high entropy, stability throughout life, epigentic nature, and proliferation of sensors collecting iris images [PPJ03]. We use the formalism of fuzzy extractors introduced by Dodis, Ostrovsky, Reyzin, and Smith [DRS04, DORS08]. Much of the below discussion applies to the related primitives of fuzzy commitment [JW99] and secure sketches [DORS08].

Fuzzy Extractors Fuzzy extractors derive stable keys from a biometric. Fuzzy extractors consist of two algorithms Gen, or generate, and Rep, or reproduce. The Gen algorithm takes an initial reading of the biometric, denoted w, deriving a key Key and a value Pub. The Rep algorithm is used at authentication time taking Pub and a later reading of the biometric, denoted w'. If the two readings of the biometric are similar enough then the same Key should be output by the algorithm. The security of a fuzzy extractor is analyzed assuming the adversary knows Pub.

Provable, cryptographic security is crucial for biometric storage systems because biometrics cannot be changed or updated. Thus, a compromise affects an individual for their entire life. Furthermore, since individuals have a limited number of biometrics, allowing an adversary to learn a single biometric value will likely compromise a user in multiple applications. We thus focus on a strengthening of fuzzy extractors called *reusable* fuzzy extractors that allows derivation of multiple keys and multiple public values from the same biometric. We formally define our security model in Section 3.

Researchers designed a first generation of fuzzy extractors that share the same core construction and security analysis. Suppose that the original reading of a biometric is drawn from a distribution W. If W were uniformly distributed and could be reproduced exactly one could build the following algorithm: $\mathsf{Gen}(w)$: 1) sample a random Key and 2) set $\mathsf{Pub} = \mathsf{Key} \oplus w$. If W were stable and uniform, then w acts as a one-time pad. However, if a subsequent reading of the biometric, denoted by w', is only similar to w, the correct key cannot be found using Pub and w'. This problem can be rectified by first encoding Key using an error correcting code. The new construction is: 1) sample a random Key, 2) encode Key using an error-correcting code to obtain c and 3) set $\mathsf{Pub} = \mathsf{Key} \oplus w$. Consider the value

$$\mathsf{Pub} \oplus w' = c \oplus w \oplus w' = c \oplus (w \oplus w').$$

If w and w' are similar enough then $w \oplus w'$ is "small" and can be viewed as an error term. This means that c can be recovered using the decoding algorithm of the error correcting code. This construction is called the *code-offset construction* and security follows by one-time pad analysis. If biometrics were drawn from the uniform distribution the problem would be solved.

Unfortunately, biometrics are drawn from nonuniform distributions. Daugman's seminal paper on iris recognition [Dau04] transformed iris images into a fixed length 2048 bit vector that is mostly stable under environmental variation. Daugman estimates this string produces a distribution that has 249 bits of entropy.[1] Repeated readings of the same iris differ in between $10 - 30\%$ of bits of the vector.

The analysis of the code-offset construction has seen two extensions 1) when bits of w are independent and 2) when bits may be correlated. We are not aware of a biometric where bits are independent so we focus on the second case. Unfortunately, if bits may be correlated known analysis methods are crude. Suppose that a distribution W over an n-bit space has k bits of entropy. One can extend the one-time pad argument to say that Pub leaks at most $n - k$ bits about Key. Error-correcting codes are a well studied object with clear bounds on their size. If one wishes

[1] Any distribution limited to people on the earth can be described using 33 bits. The estimate of 249 should be understood as the randomness involved in creating a new iris.

to tolerate t bits of error between w and w', one needs an error-correcting code that tolerates t errors. Such a code has at most $2^{n-h_2(t/n)*n}$ codewords.[2]

With $n-k$ bits of leakage security is analyzed by calculating

$$n - h_2(t/n) * n - (n-k) = k - h_2(t/n) * n.$$

In many cases $h_2(t/n) * n$ is larger than k, so it is not clear how to analyze the secrecy of Key. Daugman reports error rates close to 10% in a controlled environment. For more realistic datasets the error rate is 30%. In either case, $h_2(t/n) * n \geq h_2(.1) * 2048 \approx 874$ is larger than the estimated entropy of 249. To the best of our knowledge it is not known how to analyze the first generation of fuzzy extractors to argue security for the iris. (We discuss prior work in Section 2.)

Recently, a second generation of fuzzy extractors emerged using cryptographic tools [FMR13, CFP+16, ACEK17, ABC+18, WLH18, WL18]. These constructions only provide security against computationally bounded adversaries (the code-offset construction provides security against computationally unbounded adversaries). Some of these constructions provide meaningful security when W has low entropy. However, this security requires W to have additional structure beyond entropy. There have been no empirical evaluations of whether biometrics exhibit this structure. Furthermore, these constructions are stated in asymptotic form and it is not clear what properties they provide for actual biometrics.

1.1 Our contribution

We build the first key derivation system that provides meaningful albeit moderate provable security from the iris. Our scheme has been implemented and open-sourced. The combination of cryptographic and statistical analysis estimates a security level of 32 bits. As a point of comparison, recent estimates place password entropy at 34 bits [KSK+11].

To be clear, we do not believe this security level is sufficient for a stand alone system. Our hope is that this work serves as a catalyst for system designers to incorporate our construction into multi-factor authentication systems and that the overall system provides strong security. We provide some initial discussion in Section 4.1.

The starting point for our construction is the recent sample-then-lock scheme of Canetti et al. The idea of the scheme is simple: to hash the biometric in an "error-tolerant" way. Hashing the full biometric doesn't work (due to biometric noise). Instead, multiple random subsets of the biometric are hashed. That is, sample a random subset of bits, denoted $\mathcal{I}$, and hash w restricted to the bits of $\mathcal{I}$, denoted $\mathsf{Hash}(w_{\mathcal{I}})$, and use this value as a pad for a cryptographic Key. That is, store $\mathsf{Key} \oplus \mathsf{Hash}(w_{\mathcal{I}})$. This process is repeated with multiple subsets $\mathcal{I}_j$ and the same Key. Correctness follows if it is likely that in at least one subset $\mathcal{I}_j$, $w_{\mathcal{I}_j} = w'_{\mathcal{I}_j}$. Importantly, the security analysis requires for a random subset $\mathcal{I}_j$ of bits that $w_{\mathcal{I}_j}$ has entropy with high probability over the choice of $\mathcal{I}_j$ (see Definition 6). This strengthens the requirement that the whole vector W has entropy. We first estimate that subsampling iris bits produces a distribution with entropy.[3] However, we find that the naive combination of iris processing and the fuzzy extractor provides inadequate security and efficiency. The core of our technical contribution is two-fold:

1. Modifying the fuzzy extractor to support a more efficient implementation (and a modified security proof).
2. Modifying the iris image processing to maximize security of the resulting scheme.

A major part of this work is statistical analysis to verify that irises exhibit the required structure for security of the fuzzy extractor. In addition to this security analysis we report system correctness, storage requirements, and timing. All of our analysis uses the ND-0405 iris data set [PSJO+06, BF16] which is a superset of the NIST Iris Challenge Evaluation Dataset [PBF+08]. Throughout our work we explicitly state what assumptions are needed for security of the scheme to hold.

[2]The quantity $h_2(t/n) * n$ is the binary entropy of t/n multiplied by n. The quantity $h_2(t/n) * n$ is larger than t (when $t \leq .5n$). (For example, if $t = .1n$ then $h_2(t/n) * n \approx .427n$.)

[3]Measuring entropy is inherently heuristic. Provably accurate entropy estimation [VV10, VV11] requires an exponentially large number of samples in the actual entropy of the distribution.

We use the same heuristic that occurs roughly in previous biometric research. Roughly, the distances between transformed irises of different individuals are compared with the distances that would be produced by a distribution with known entropy. This heuristic is discussed in Section 6.

Scheme	Limitation
Code Offset [Boy04]	Assumes noise uncorrelated to iris
LWE Decoding [ACEK17]	Assumes noise uncorrelated to iris
Diffie-Hellman [WLH18]	Assumes noise (largely) uncorrelated to iris
Pseudo. isometry [ABC+18]	Applicable for set difference metric
Sample-then-lock [CFP+16]	Sublinear correction capacity

Table 1: Recent constructions of reusable fuzzy extractors. The code offset and LWE decoding schemes leak information about the difference between repeated readings w_i, w_j which may be correlated to the individual readings. The Diffie-Hellman construction requires a small amount of correlation, it is secure if each w_i has high entropy conditioned on the differences between all pairs of readings.

Organization The rest of this work is organized as follows, Section 2 reviews prior work in the area, in Section 3 we review basic definitions and nonstandard cryptographic tools, Section 4 describes our scheme, Section 5 the resulting software, Section 6 describes iris image processing and the transform used as input to our scheme, Section 7 evaluates the performance and correctness of our system, Section 8 concludes.

In the Appendix, we provide additional statistical analysis and describe a second version of our system that communicates additional information from the image processing to the cryptographic system. This additional information is called "confidence information." This system provides a higher bit level of security but requires an additional assumption about the independence of the confidence information and the iris.

2 Prior work

We split our discussion of prior work into two parts, cryptographic theory and systems.

Reusable Fuzzy Extractors We reviewed interactive solutions and traditional fuzzy extractors in the introduction. Thus, we focus our review of theory to reusable fuzzy extractors. Boyen [Boy04] defined reusable fuzzy extractors in 2004 and showed information-theoretic reusability requires a large decrease in security [Boy04, Theorem 11].[4] Essentially, Boyen showed the security loss of $n-k$ described in the introduction is *necessary* (and cannot be avoided by clever analysis). Applied works showed that many fuzzy extractors were not reusable [STP09, BA12, BA13], meaning that the negative result of Boyen was not only a theoretic issue.

Recent work providing security only against computationally bounded attackers [FMR13] may be able to sidestep Boyen's negative result. Most reusable fuzzy extractors were described in the last two years (summarized in Table 1). A key consideration in reusable fuzzy extractors is the type of correlation assumed between enrollments W_i and W_j. In many constructions it is assumed that $W_i \oplus W_j$ does not leak information about W_i or W_j. This assumption has not been verified in practice. This assumption is made by the first three schemes in Table 1. Alamelou et al. construct a reusable fuzzy extractor is for a different distance metric (set difference metric), their system is not applicable for the iris [ABC+18]. The only construction that appears viable is the sample-then-lock construction (our starting point).

Iris Key Derivation Systems Many previous works have used a fuzzy extractor in combination with the iris. These works report key strengths that are troubling for a variety of reasons.

Hao et al. [HAD06] use the code-offset construction with a code with 2^{140} codewords with the Iriscode transform of Daugman [Dau04]. The standard analysis of the introduction provides no guarantee for this code as $140 < (n-k) = 2048 - 249 = 1799$. Hao et al. claim a key strength of 140 without further justification. Hao et al. then argue an adversary providing a random iris would succeed with probability 2^{-44}. This corresponds to an adversary that does not have access to Pub (plaintext template storage suffices in this model).

[4]The actual result of Boyen applies to *secure sketches* which imply fuzzy extractors. A secure sketch is a frequently used tool to construct a fuzzy extractor. Our construction does not utilize a secure sketch.

Bringer et al. [BCC+07] do not explicitly describe a key length but they report a nonzero false accept rate which implies a small effective key strength. Reporting a nonzero false accept rate is common in iris key derivation despite claimed key lengths > 40 bits (see discussion [PRC15, ICF+15]). Using the birthday bound, false acceptances should appear when the tested dataset size approaches the square root of the claimed key size (i.e. $> 2^{20}$). No published iris datasets have close to a million individuals.

Kanade et al. [KCK+08] claim a fuzzy extractor construction but they report the entropy of the iris as over 1000 bits, much higher than other estimates. Other research states that each bit of the iris transform is independent [GKTF16] which is demonstrably not true for standard iris techniques (see for example our statistical analysis in Section 6).

The above discussion is necessarily incomplete (see the survey of Bowyer et al. [BHF13, Section 6]). It demonstrates a large gap between theoretical fuzzy extractor constructions and their use, justifying a rigorous analysis of iris key derivation that makes assumptions explicit and accurately estimates security.

In concurrent work, Cheon et al. [CJKL18] also implemented and evaluated a practical reusable fuzzy extractor. However, this work contains a fundamental flaw in its security argument. At a high level, the authors incorrect argue that many small random oracles (of polynomial size) can't be exhausted by an unbounded adversary. This flaw has been communicated to and acknowledged by the authors. To the best of our knowledge, no revision has since been made.

3 Definitions and Cryptographic Tools

We use capital letters to refer to random variables. 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.

Definition 1. *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 2. *The* average *min-entropy of X given Y is*

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

The *statistical distance* between random variables X and Y with the same domain is $\Delta(X,Y) = \frac{1}{2}\sum_x |\Pr[X = x] - \Pr[Y = x]|$. For a distinguisher D we write the *computational distance* between X and Y as $\delta^D(X,Y) = |\mathbb{E}[D(X)] - \mathbb{E}[D(Y)]|$ (we extend it to a class of distinguishers $\mathcal{D}$ by taking the maximum over all distinguishers $D \in \mathcal{D}$). We denote by $\mathcal{D}_s$ the class of randomized circuits which output a single bit and have size at most s.

We use the version of fuzzy extractors that provides security against computationally bounded adversaries [FMR13]. Dodis et al. provide comparable definitions for information-theoretic fuzzy extractors [DORS08, Sections 2.5–4.1].

Definition 3. *[FMR13, Definition 4] Let $\mathcal{W}$ be a family of probability distributions over metric space $(\mathcal{M}, \mathsf{dis})$. A pair of randomized procedures "generate" (Gen) and "reproduce" (Rep) is an* $(\mathcal{M}, \mathcal{W}, \kappa, t)$-computational fuzzy extractor *that is $(\epsilon_{sec}, s_{sec})$-hard with error δ if Gen and Rep satisfy the following properties:*

- *Generate procedure Gen on input $w \in \mathcal{M}$ outputs a key, $\mathsf{key} \in \{0,1\}^\kappa$ and a helper string $p \in \{0,1\}^*$.*
- *The reproduction procedure Rep takes an element $w' \in \mathcal{M}$ and a bit string $p \in \{0,1\}^*$ as inputs.*
- Correctness: *if $\mathsf{dis}(w, w') \le t$ and $(r, p) \leftarrow \mathsf{Gen}(w)$,*

$$\Pr_{p,\mathsf{Rep}}[\mathsf{Rep}(w', p) = r] \ge 1 - \delta.$$

- Security*: for any distribution $W \in \mathcal{W}$, define $(R, P) \leftarrow \mathsf{Gen}(W)$ then*

$$\delta^{\mathcal{D}_{s_{sec}}}((R, P), (U_\kappa, P)) \le \epsilon_{sec}.$$

Using a biometric multiple times Individuals are born with a limited number of biometrics. If a biometric's privacy is compromised it cannot be regenerated or refreshed. A desirable property of a fuzzy extractor is that an individual can enroll their biometric with multiple service providers and retain security. Informally, each cryptographic key should be secure if an adversary knows all public helper values and all other derived keys.

Definition 4. *Let $\mathcal{W}$ be a family of distributions over $\mathcal{M}$. Let $(\mathsf{Gen}, \mathsf{Rep})$ be a $(\mathcal{M}, \mathcal{W}, \kappa, t)$-computational fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$-hard with error δ. Let $(W^1, W^2, \ldots, W^\rho)$ be ρ correlated random variables such that each $W^j \in \mathcal{W}$. Let D be an adversary. Define the following game for all $j = 1, ..., \rho$:*

- ***Sampling*** *The challenger samples $w^j \leftarrow W^j$, $u \leftarrow \{0,1\}^\kappa$.*
- ***Generation*** *The challenger computes $(r^j, p^j) \leftarrow \mathsf{Gen}(w^j)$.*
- ***Distinguishing*** *The advantage of D is*

$$\Pr[D(r^1, ..., r^{j-1}, r^j, r^{j+1}, ..., r^\rho, p^1, ..., p^\rho) = 1] - \Pr[D(r^1, ..., r^{j-1}, u, r^{j+1}, ..., r^\rho, p^1, ..., p^\rho) = 1].$$

$(\mathsf{Gen}, \mathsf{Rep})$ is $(\rho, \epsilon_{sec}, s_{sec})$-reusable if for all $D \in \mathcal{D}_{s_{sec}}$ and for all $j = 1, ..., \rho$, the advantage is at most ϵ_{sec}.

Digital Lockers Our construction uses digital lockers, which are computationally secure symmetric encryption schemes that retain security even when used multiple times with correlated and weak (i.e., nonuniform) keys [CKVW10]. A digital locker is an algorithm lock which takes a key and a value, producing an algorithm unlock, unlock reproduces the value if and only if the same key is provided as input. Somewhat confusingly, we will lock a key using part of the biometric. That is, the locked value as a cryptographic key. Thus, we refer to the locked value as the *key* and the value used to open the lock as the *combination*. Digital lockers have two important properties:

1. The only way to obtaining any information about the key is by guessing the combination.
2. It is possible to detect the wrong combination with high probability.

Digital lockers can be constructed from variants of the Diffie-Hellman assumption [CD08]. Let $\mathtt{HMAC}$ be HMAC-SHA256. Our construction assumes that $\mathtt{HMAC}$ can be used to construct digital lockers. The "locking" algorithm outputs the pair

$$\mathsf{nonce}, \mathtt{HMAC}(\mathsf{nonce}, w) \oplus (0^{128} || \mathsf{Key}),$$

where nonce is a nonce, $||$ denotes concatenation, 0^{128} is the all zeros string of length 128, a security parameter. 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. This construction was proposed in [BR93] and shown to be secure in the random oracle model by Lynn, Prabhakaran, and Sahai [LPS04, Section 4] It is plausible that in the standard model (without random oracles) hash functions provide the necessary security [CD08, Section 3.2], [Dak09, Section 8.2.3]. We now present the full formal definition [BC10]:

Definition 5. *The pair of algorithm $(\mathsf{lock}, \mathsf{unlock})$ with security parameter λ is an ℓ-composable secure digital locker with error γ if the following hold:*
Correctness *For any pair $\mathsf{key}, \mathsf{val}$,*

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

Also, for any $\mathsf{key}' \neq \mathsf{key}$,

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

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

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

where S is allowed $q(\lambda)$ oracle queries to the oracles

$$\{\mathsf{idealUnlock}(\mathsf{key}_i, \mathsf{val}_i)\}_{i=1}^{\ell}.$$

Technical Remark: Unfortunately, the security definition of digital lockers (Definition 5) is "inherently" asymptotic. A different simulator is allowed for each distance bound $p(s)$ making it difficult to argue what quality key is provided with respect to a particular adversary.

4 Our Scheme

Our construction builds on the construction of Canetti et al. [CFP+16]. The high level idea is to encrypt the same key multiple times using different subsets of w. Pseudocode for the algorithm is below:

$\mathsf{Gen}(w)$**:**

1. Sample random 128 bit Key.
2. For $i = 1, ..., \ell$:
   (i) Choose $1 \leq j_{i,1}, ..., j_{i,k} \leq |w|$
   (ii) Choose 512 bit hash key h_i.
   (iii) Set $c_i = \mathtt{HMAC}(h_i, w_{j_{i,1}}, ..., w_{j_{i,k}})$.
   (iv) Set $p_i = (0^{128} || \mathsf{Key}) \oplus c_i$.
3. Output $(\mathsf{Key}, p_i, v_i, h_i)$.

$\mathsf{Rep}(w', p_1, ..., p_\ell, v_1, ..., v_\ell, h_1, ..., h_\ell)$**:**

1. For $i = 1, ..., \ell$:
   (i) Set $c_i = \mathtt{HMAC}(h_i, w'_{j_{i,1}}, ..., w'_{j_{i,k}})$.
   (ii) If $(c_i \oplus p_i)_{1..128} = 0^{128}$ then output $(c_i \oplus p_i)_{129..256}$.
2. Output $\perp$.

In the description above, $x_{a..b}$ denotes the restriction of a vector to the bits between a and b. The parameters k and ℓ represent a tradeoff between correctness and security.

For the scheme to be correct at least one of the ℓ subsets should have no error with high probability. Canetti et al. show it is possible to set ℓ if the expected error rate is sublinear in $|w|$. That is, when $d(w, w')/|w| = o(|w|)$. We set ℓ and k in Section 6.

A single digital locker requires storage of 32 bytes for the output of the hash 64 bytes for each hash key h_i. In addition, the public value must store the randomly sampled locations. The two natural solutions for this are 1) storing a mask of size $|w|$ for each subset or 2) a location set of size $\log |w| * k$ for each subset. Using either approach, in our analysis, storing subsets required more space that the hash outputs and keys. This led to our main modification of the cryptographic scheme.

Canetti et al. [CFP+16, Section 4] note that rather than using independent subsets they could be selected using a sampler [Gol11]. We observe the security argument holds as long as each subset is random on its own. That is, the different subsets can be arbitrarily correlated. We will use this fact to reduce the storage requirement of the scheme. Before stating our condition we introduce the necessary condition for security on the distribution W.

Definition 6. *Let $W = W_1, \ldots, W_n$ be a distribution over $\{0,1\}^n$). For k, α, we say that W is a source with* α-entropy k-samples *if $\tilde{\mathrm{H}}_\infty(W_{j_1}, \ldots, W_{j_k} \,|\, j_1, \ldots, j_k) \geq \alpha$ for uniformly random $1 \leq j_1, \ldots, j_k \leq n$.*

We now state security of the modified scheme.

Theorem 1. *Let λ be a security parameter, Let $\mathcal{W}$ be a family with α-entropy k-samples for $\alpha = \omega(\log \lambda)$. Suppose the HMAC construction is a secure digital locker. Let $\mathcal{I}_j$ be the jth subset generated in* Gen*. The above fuzzy extractor is secure if each individual $\mathcal{I}_j$ is uniformly distributed (but different subsets $\mathcal{I}_j, \mathcal{I}_\ell$ are potentially correlated). More formally, for any $s_{sec} = \mathtt{poly}(\lambda)$ there exists some $\epsilon_{sec} = \mathtt{ngl}(\lambda)$ such that* sample-then-lock *is a $(\mathcal{Z}^n, \mathcal{W}, \kappa, t)$-computational fuzzy extractor that is $(\epsilon_{sec}, s_{sec})$-hard with error $\delta = \mathtt{negl}(\lambda)$.*

No claim about correctness is made if $\mathcal{I}_j$ and $\mathcal{I}_\ell$ are correlated.

We only show security when Gen is run once, reusability follows using the same argument as in Canetti et al. [CFP+16].

Proof. Let $V_1, ..., V_\ell$ be random variables corresponding to W restricted to the bits selected in subset $\mathcal{I}_i$. Similarly, let P_i be the random variable corresponding to the public part of the output produced in iteration i. Let R denote the distribution over output key values. Lastly, let U denote the uniform distribution over $\{0,1\}^{|\mathsf{key}|}$.

We show for all $s_{sec} = \mathtt{poly}(\lambda)$ there exists $\epsilon_{sec} = \mathtt{ngl}(\lambda)$ such that $\delta^{\mathcal{D}_{s_{sec}}}((R, \{P_i\}_{i=1}^{\ell}), (U, \{P_i\}_{i=1}^{\ell})) \leq \epsilon_{sec}$. Fix some polynomial s_{sec} and let D be a distinguisher of size at most s_{sec}.

We proceed by contradiction: supposing

$$|\,\mathbb{E}[D(R, \{P_i\}_{i=1}^{\ell})] - \mathbb{E}[D(U, \{P_i\}_{i=1}^{\ell})]|$$

is not negligible. Suppose there is a polynomial $p(\cdot)$ such that for all λ_0 there exists some $\lambda > \lambda_0$ such that

$$|\,\mathbb{E}[D(R, \{P_i\}_{i=1}^{\ell})] - \mathbb{E}[D(U, \{P_i\}_{i=1}^{\ell})]| > 1/p(\lambda).$$

By Definition 5, there is a polynomial q and an unbounded time simulator S (making at most $q(\lambda)$ queries to the oracles $\{\mathsf{idealUnlock}(v_i, r)\}_{i=1}^{\ell}$) such that

$$\begin{aligned}\frac{1}{3p(\lambda)} \geq |\,&\mathbb{E}[D(R, P_1, ..., P_\ell)] \\ &- \mathbb{E}\left[S^{\{\mathsf{idealUnlock}(v_i,r)\}_{i=1}^{\ell}}\left(R, \{\mathcal{I}_i\}_{i=1}^{\ell}, k, |\mathsf{key}|\right)\right]|\end{aligned} \tag{1}$$

This is also true if we replace R with an independent uniform random variable U over $\{0,1\}^{|\mathsf{key}|}$. We now prove the following lemma, which shows that S cannot distinguish between R and a independent U.

Lemma 1. *Let all variables be as above. Then*

$$\left|\,\mathbb{E}\left[S^{\{\mathsf{idealUnlock}(v_i,r)\}_{i=1}^{\ell}}\left(R, \{\mathcal{I}_i\}_{i=1}^{\ell}, k, |\mathsf{key}|\right)\right] - \mathbb{E}\left[S^{\{\mathsf{idealUnlock}(v_i,r)\}_{i=1}^{\ell}}\left(U, \{\mathcal{I}_i\}_{i=1}^{\ell}, k, |\mathsf{key}|\right)\right]\,\right| \leq \frac{q(q+1)}{2^\alpha} \leq \frac{1}{3p(\lambda)}$$

where q is the maximum number of queries S can make.

Proof. Fix some $u \in \{0,1\}^{|\mathsf{key}|}$. The only information about whether the value is r or u can obtained by S through the query responses. First, modify S slightly to quit immediately if it gets a response not equal to $\perp$. There are $q+1$ possible values for the view of S on a given input (q of those views consist of some number of $\perp$ responses followed by the first non-$\perp$ response, and one view has all q responses equal to $\perp$). By [DORS08, Lemma 2.2b], $\tilde{\mathrm{H}}_\infty(V_i|View(S), \{\mathcal{I}_j\}) \geq \tilde{\mathrm{H}}_\infty(V_i|\{\mathcal{I}_j\}) - \log(q+1) \geq \alpha - \log(q+1)$. Therefore, at each query, the probability that S gets a non-$\perp$ answer (equivalently, guesses V_i) is at most $(q+1)2^{-\alpha}$. Since there are q queries of S, the overall probability is at most $q(q+1)/2^\alpha$. Then since 2^α is $\mathtt{ngl}(\lambda)$, there exists some λ_0 such that for all $\lambda > \lambda_0$, $q(q+1)/2^\alpha \leq 1/(3p(\lambda))$. □

The overall theorem follows using the triangle inequality with equation 1, equation 1 with R replaced with U, and Lemma 1 yielding $\delta^D((R,P),(U,P)) \leq 1/p(\lambda)$. This completes the proof of Theorem 1. □

This theorem gives us a mechanism for saving on storage size. However, using the same subset for each locker will destroy the correctness argument. Looking ahead to Section 5, instead of choosing independent subsets, the implementation chooses a master subset and then generate permutations π_j to create new subsets based on public cryptographic keying material.

Unlinkability Unlinkability prevents an adversary from telling if two enrollments correspond to the same physical source [CS08, KBK+11]. Our construction satisfies unlinkability (as long as the digital locker is not broken) as the only stored information is padded hash outputs and the subsets.

4.1 Adding more factors

Many multi factor authentication systems do not achieve "additive security." Consider a strawman authentication system: 1) a user inputs a password and 2) an iris. Currently, the password would be hashed and compared to stored hash and the iris compared to a template. One of these comparisons has to be done first. In either case based on time or error messages it is often possible to perform a brute force search on each factor separately.

A noiseless password can be used as part of the input to any fuzzy extractor and strengthen key derivation. However, previous fuzzy extractors separate the error-correction from the key derivation process using two distinct primitives called secure sketch [DORS08] and randomness extractor respectively [NZ93]. The secure sketch is responsible for mapping w' back to the input w, while the randomness extractor converts entropy into a random key. In such a process, the password can only be incorporated into the randomness extractor (and not used in the secure sketch). Many secure sketches report an error when $d(w, w') > t$ (known as being well formed). If such an error is visible to the adversary they can separate searching w' and searching for the password. Hiding such timing channels is notoriously difficult.

Our construction does not suffer from this problem. "Error-correction" and key derivation are performed simultaneously. The password can be prepended as input to each hash invocation without affecting storage or computational requirements. Recent estimates place password entropy at 34 bits [KSK+11].

5 Implementation

We implemented our construction in both Python and C and both implementations are open sourced [FSS18]. Previous implementations of fuzzy extractors required expertise in error-correcting codes. Our construction only requires repeated evaluation of a hash function. The Python implementation is simple and functions primarily as a comparison point for our C implementation. Our C implementation is optimized, using low-level bit and cryptographic operations.

The entire Python library is 100 lines of code with dependencies on numPy (for array manipulation), random, and hashlib. Our Gen code is single threaded because the majority of execution time is spent in system calls to achieve the randomness needed for the subsets $j_{i,1}, ..., j_{i,k}$. The Rep functionality is embarrassingly parallel. We implemented a parallel version that simply partitions the hashes to be performed. Rep succeeds when one of these threads returns. Unfortunately, neither implementation is fast enough with authentication taking seconds (see Section 7).

We also developed an optimized C implementation designed for fast Rep performance. As Rep is used at every authentication its speed is more important than Gen which is only used when a user enrolls with a new service. For this implementation we used Libsodium [BLS12] as the cryptographic backend and HMAC-SHA-512 to instantiate the digital locker. This library makes use of low level bit level operations for quickly packing and selecting bits the iris vector. In preliminary testing a major obstacle to fast Rep was disk load time. Recall, each subset selected in Gen requires storage of 96 bytes plus the subset itself. We eliminate storing the subset (relying on Theorem 1 for security). The new scheme works as follows:

1. Choose a master subset $\mathcal{I}$ uniformly at random where $|\mathcal{I}| = k$.
2. For each locker j generate a permutation $\pi_j : \{0,1\}^{|w|} \to \{0,1\}^{|w|}$.
3. Apply π_j to each element of $\mathcal{I}$ to get $\mathcal{I}_j$.

To efficiently generate permutations we do the following:

1. Select a single master CHACHA20 key
2. Encrypt the permutation number j, creating $\log|w| * |w|$ bits of output c.
3. We split c into $\log|w|$ bit sections $c_1, ..., c_{|w|}$.
4. Define $\pi_j(i) = c_i$

The output of CHACHA20 is not a permutation: it is not guaranteed that $\log|w|$ consecutive bits do not repeat. Furthermore, looking ahead to Section 6, our iris processing results in a vector of 12000 bits. The above algorithm

only works if $|w|$ is a power of 2. We adapt our algorithm by adding a check for each section c_i, if $c_i > |w|$ or c_i is repeated it is discarded. To compensate for these two failure conditions it is necessary to produce more than $|w|$ sections. Producing 2000 additional sections was sufficient to always output a permutation in our experiments.

This modification reduces overall storage to a single CHACHA20 key, the single randomly generated subset, and 96 byte per subset storage. Generating these permutations takes additional computation. One can tradeoff between storing all subsets and a single master subset, storing some fraction of subsets and regenerating the rest.

We are not aware of how to reduce the 96 byte per subset storage. An idea is to use a single nonce, we were not able to argue security of this modified scheme. We leave this as an open problem.

6 Iris Image Processing and Setting Parameters

This section provides a brief overview of iris image processing and the transform used in our system. Iris image processing is an entire field [BHF08], we cannot do it justice here. Our scheme can be used with techniques that produce a binary vector with Hamming errors (fraction of bits that are the same).

The starting point for our transform is open-source OSIRIS package [KMSD17]. We use this package as a starting point as it is open source and uses representative techniques.[5] OSIRIS takes a near infrared iris image and produces a 32768 bit vector w. The stages of OSIRIS are:

1. Iris and Pupil Localization: This step finds the inner and outer boundaries of the iris accounting for pupil dilatation and occlusions due to the eyelid or eyelashes.

2. Iris Unwrapping: The iris is converted into a 2D matrix (using the rubber band transform). This array is indexed by (r, θ) which is the polar position of the pixel in the original image.

3. Featurization: 2D Gabor filters [GM84] centered at different positions are convolved with the image yielding a complex values at locations (r, θ). This produces a 64×512 vector of complex valued numbers.

4. Binarization: Complex numbers are quantized based on sign to produce two bits.

The OSIRIS library includes six transforms. These transforms are the real and imaginary components of three different sets of Gabor filters. Our experiments showed the histogram with the lowest error rate (for images of the same iris) was Transform 5. We thus used Transform 5 for all of our analysis. The resulting iris biometric has several desirable properties:

1. Different individuals have fractional Hamming distance tightly distributed around .5.

2. Previous estimates place entropy at 249 bits [Dau04].

3. Images from the same individual have a mean fractional Hamming distance of $.11 - .32$. Daugman reports .11, we observe a mean error rate of .32 using OSIRIS and the ND-0405 dataset [PSJO$^+$06, BF16].

We observe an mean error rate of 32% using the ND-0405 Iris data set [PSJO$^+$06, BF16]. Daugman reports mean error rates of 11%, but we are unaware of any subsequent work that achieves as low an error rate as 11%.[6] In addition, the construction requires subset entropy (see Definition 6). We check this condition first statistical analysis before trying to set parameters (otherwise the construction is unlikely to be fruitful).

All statistical analysis is performed using the ND-0405 dataset [BF16] which is a superset of the NIST Iris Challenge Evaluation Dataset [PBF$^+$08]. The ND-0405 dataset includes 356 persons and 64964 total images.

Our analysis includes *intraclass* comparisons which are comparisons of the Hamming distance between two transformed images of the same iris and *interclass* comparisons which are comparisons of the Hamming distance between two transformed images of different irises. The ND-0405 dataset contains images from the left and right eye of the same individual. These are treated as interclass comparisons.

Figure 1 shows the histograms for fractional Hamming distance between two images of the same individual (*same*) and different individuals (*different*) for the dataset. This histogram is produced by computing the fractional

[5] Like many learning tasks, iris recognition is currently transitioning to deep learning based approaches [AA08, SHN$^+$11].

[6] The security/correctness tradeoff of our system immediately improves with an iris transform with lower error rate.

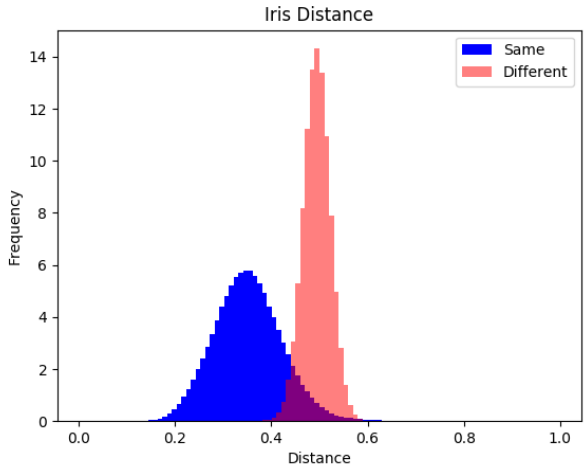


Figure 1: Distribution of distances for the data set.

Hamming distance of every iris with every other iris (for a total of $\approx 10^9$ comparisons). The fractional Hamming distances were then grouped into *interclass/different* comparisons corresponding to the same iris and *intraclass/same* comparisons corresponding to different irises. The error rate of the data is defined as the expected fractional Hamming distance between two images of the same iris. We observed a mean error rate of .32. For different irises, we observe the interclass mean and interclass variance as $\mu = .494$ and $\sigma = .0008$.

The standard method for estimating the entropy of the iris [Dau04] is to compare the interclass histogram with a Binomial distribution with the same mean μ and variance σ. If the observed distribution and the Binomial distribution have very similar histograms, then the observed distribution is assumed to have the same entropy as the Binomial distribution. This technique is necessarily a heuristic.

We computed this heuristic generating a binomial distribution with mean $\mu = .494$ and variance $\sigma = .0008$. The statistical distance between the interclass histogram and the binomial distribution was computed with a total statistical distance of .005. The difference between the two probability mass functions is in the Appendix in Figure 6. From the figure and the total statistical distance the binomial distribution appears a good fit for the observed distribution. Thus, we use the entropy of the Binomial distribution as a stand in for the entropy of the observed distribution. The entropy of the Binomial is calculated using the following equations (where dF stands for degrees of freedom):

$$\begin{aligned} \mathrm{dF} &= \frac{\mu(1-\mu)}{\sigma} = 311 \\ \text{entropy} &= (-\mu \log \mu - (1-\mu)\log(1-\mu)) * \mathrm{dF} \\ &= 311. \end{aligned}$$

Our entropy estimate is different from Daugman's. It is common for this estimate to vary across data sets, this estimate is capturing useful information of the underlying biologic process and noise which is less useful. However, since the construction has to "correct" the noise, the noise should also be counted for security.

6.1 Entropy of Subsamples

We now turn to the sufficient condition for security: do random samples of the iris have entropy? In the worst case, sampling only preserves the entropy rate of a distribution which for OSIRIS is $311/32768 \approx 1\%$.

Iris entropy is believed to be geographically distributed throughout the iris. The OSIRIS output is produced by convolving a fixed Gabor filter at overlapping regions of the unwrapped iris. So one would expect nearby bits to be

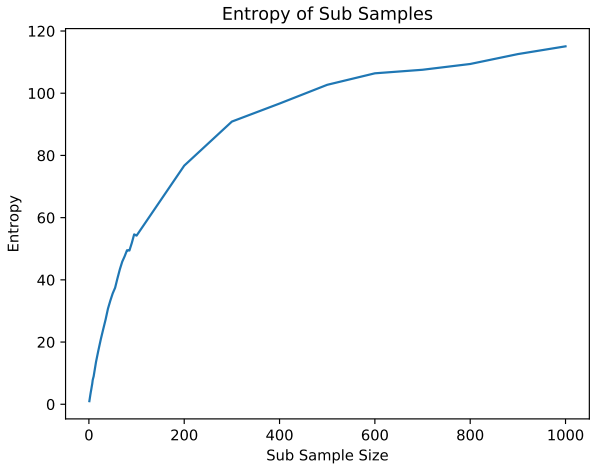


Figure 2: In the worst case subsampling only preserves error rate. For the iris, subsampling greatly increases entropy rate from 1% to over 80%.

correlated. If only nearby bits are correlated, subsampling random bits will increase the entropy rate. To test this hypothesis, we performed the following analysis (for subset size k) with 10 trials for each subset size:

1. Randomly sample k distinct positions.
2. Compute the intraclass and interclass histograms for the dataset restricted to these positions.
3. Compute the μ and σ for the interclass histogram. (Using the same method as in Figure 1).
4. Compute the entropy e_i for trial i.
5. Compute the overall entropy as $e = -\log \mathbb{E}_i \, 2^{-e_i}$

In this last step we average the entropy calculation using average min-entropy (Definition 2). This technique is preferable to averaging the entropies e_i. We are targeting security which requires that the entropy should be high in all settings, not just on average. Consider five possible events where the entropy conditioned on the events is $1, 100, 100, 100, 100$ respectively. Then the "average entropy" is ≈ 80 while the average min-entropy is ≈ 3. However, clearly in this situation the individual with an entropy of 1 is in trouble. We find the average of entropies and the *average min-entropy* differ substantially. In some rare events, positions are chosen close together yielding a low entropy.

This analysis was performed for subset sizes $k \in \{1, 2, ..., 10\} \cup \{15, 20, ..., 100\} \cup \{200, 300, ..., 1000\}$ with 10 trials for each size.

Since we are randomly subsampling from a distribution that fits the binomial well, the distribution was assumed to also fit a binomial distribution. Results are in Figure 2. We note that the entropy rate is significantly higher than the worst case of 1%. At some points in Figure 2 the entropy rate exceeds 80%.

6.2 Choosing reference parameters and optimizing

In this subsection, we define some reference parameters for an instantiation of the scheme. This is done as a baseline for comparison for the optimization that follows.

For our construction the two tunable parameters are the number of subsets, ℓ, and the number of bits in each subset, k. Increasing k improves security but hurts correctness, increasing ℓ improves correctness but costs time and storage. The two parameters are related by

$$1 - (1 - (1 - \text{error rate})^k)^\ell = \Pr[\text{correct}].$$

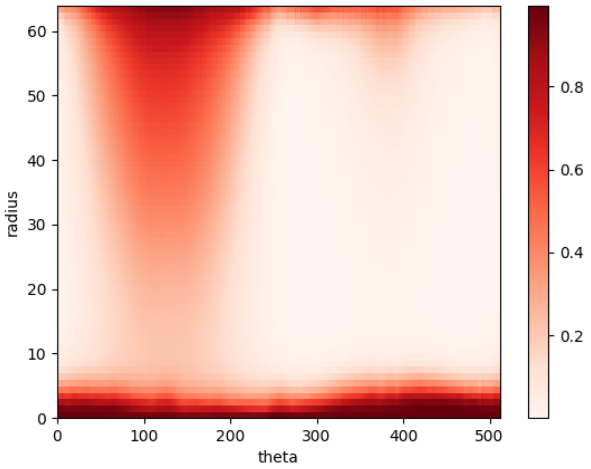


Figure 3: Heatmap of polar coordinate position in the iris (r, θ). The intensity of the pixel indicates the probability that the bit will be "masked" by the transform. The inside and outside of the iris (small and large radius respectively) are likely to be masked as well as the top of the iris (when theta is between 100-200).

We will set the number of lockers $\ell = 10^6$. This results in storage of approximately 100 MB which is dominated by the per locker storage of the HMAC key and output. We assume a *correctness target* of 50% true positive rate. While this number is unacceptable for an authentication system, correctness rate is an "s-curve" in error rate. Correctness increases quickly once it hits 50%, achieving correctness of $1 - 2^{-x}$ for some x requires multiplicatively increasing the number of lockers by 2^{x-1}. So 93.75% correctness requires 8×10^6 lockers (roughly 800MB of storage). We consider these parameters fixed.

Optimizing the transform A technique commonly used to improve iris transforms is called *masking*. (Bowyer et al. survey iris processing techniques [BHF08].) In most iris transforms in addition to the binary vector w the transform additionally outputs a second vector $mask$. Bits set in $mask$ indicate an error in the transform perhaps due to an eyelash or eyelid (known as an occlusion). Rather than comparing the Hamming distance $d(w, w')$, the authentication only compares locations i where $mask_i = 0 = mask'_i$. The intuition behind the mask vector is that occluded locations are expected to have higher error rates and should be ignored.

A possible way to incorporate $mask$ into *sample-then-lock* is to only sample from positions that are not masked. This technique limits "comparison" to locations where $mask_i = 0$. However, $mask$ may be correlated to the underlying values w, so choosing subsets in this way may leak information to the attacker.

Instead, we note that locations to be masked are not uniformly distributed throughout the iris. Rather masked bits usually occur on the top, bottom, inside and outside of the iris [HBF09]. In Figure 3, the radius and theta are polar locations with respect to the "center" of the iris. These polar coordinates are mapped to the rectangular space with the y axis representing r and the x axis representing θ. The radius r represents the distance from the pupil and θ represents the angle from 0°. This figure was calculated by using OSIRIS to compute a mask vector for every image in the ND-0405 dataset. Then, for every location (r, θ) the total number of times that the bit was masked was divided by 64964 (the number of images in the ND-0405 dataset) to normalize the value to between $[0, 1]$. As noted in prior work, the inner and outer rings of the iris are frequently masked in addition to bits with $100 \leq \theta \leq 200$ (the top of the iris which is frequently occluded by the top eyelid).

We will use this observation to design a new iris transform that restricts of the 32768 vector restricted to locations that are unlikely to be masked. We denote by pr_{mask} the vector shown visually in Figure 3). To find the right restriction we did the following for a threshold $thres \in \{1, .0975, .095, .0925, ..., .05, .025\} \cup \{.015\}$.

1. Restrict the input locations to positions j where $pr_{mask,j} > thres$.

2. Compute the mean error rate restricted to these bits.

3. Compute the maximum subset size k such that

$$1 - (1 - (1 - \text{error rate})^k)^{10^6} \geq .5.$$

4. Repeat 10 times:

   (a) Sample k random bits $\mathcal{I}$ from possible locations (where $pr_{mask,j} > thres$).

   (b) Restrict the input dataset to locations in $\mathcal{I}$. Compute interclass histogram across the entire dataset.

   (c) Compute $\mu_{thres,i}, \sigma_{thres,i}$ for trial i.

   (d) Compute the entropy $e_{thres,i}$ for trial i.

5. Compute the overall entropy as $e_{thres} = -\log \mathbb{E}_i \, 2^{-e_{thres,i}}$

The outcome of this analysis is in Table 2. We compare this approach with restricted to bits that demonstrate the highest error rate in Appendix 9. Both approaches result in similar parameters.

We include the 12000 bits that are least likely to be masked as our "iris transform." This was the size that allowed the highest subset size where entropy was close to the maximum. From this point forward, we assume a new iris processing transform that yields a 12000 output corresponding to all bits that are masked at most 5% of the time. For this choice the corresponding subset size for a correctness target of 50% and 10^6 subsets is 43 and estimated entropy of 32 bits.

7 Evaluation

In this section we evaluate the running time and correctness of our system. The basis of our security argument is Theorem 1 and Table 2 which give a necessary condition for security and the estimated entropy of subsets being used in our construction respectively.

7.1 Performance

This performance analysis was performed on a Dell Precision Tower 7000 Series with 4 Xeon E5-2620 v4 processors and 64GB of RAM. The computation was parallelism bound.

We report performance numbers for both the Python and C implementations. In the Python implementation Gen takes 220s. We implemented a parallel version of Rep which takes 12s. Since Rep must be performed on every authentication this is not fast enough for most use cases. These performance numbers do not include disk read time, which was greater than the computation time.

For the C implementation, we consider the speed of three different operations, Gen, Rep and subset generation. We do not include time for subset generation in Gen and Rep. Furthermore, we do not include disk read time. The reported times for Rep assumes the data structure is already in memory. Depending on the use case the data structure for Pub may be stored in memory, on disk, or regenerated as needed. Importantly, subset generation is independent of the iris value and can be performed ahead of time (e.g., prior to an employee starting their shift).

On average, Gen takes 6.40s, Rep takes .54s, and subset generation takes 12.93s. This data is averaged across 100 runs. The distribution of Rep times is in Figure 4. Over 72% of the time Rep completes in less than .30s. We believe the drastic difference in times separates between correct and incorrect cases. When a match is found in some thread the main thread can immediately continue. When no matches are found the main thread must wait for all threads to terminate.

Pr of mask	Number of Bits	Subsample Size	Entropy
1	32768	32	28
0.975	32341	33	30
0.95	32109	33	30
0.925	31887	33	30
0.9	31810	33	29
0.875	31695	33	30
0.85	31542	33	29
0.825	31420	33	30
0.8	31256	33	29
0.775	31099	33	29
0.75	30913	33	29
0.725	30725	33	30
0.7	30528	33	29
0.675	30283	33	29
0.65	30008	34	30
0.625	29736	34	30
0.6	29455	34	29
0.575	29113	34	30
0.55	28745	34	30
0.525	28340	34	30
0.5	27910	34	30
0.475	27483	35	30
0.45	27039	35	30
0.425	26618	35	31
0.4	26115	35	30
0.375	25618	36	31
0.35	25097	36	30
0.325	24462	36	31
0.3	23861	37	32
0.275	23196	37	30
0.25	22381	37	31
0.225	21353	38	32
0.2	20109	39	31
0.175	19144	39	32
0.15	18139	40	32
0.125	17089	40	32
0.1	15953	41	33
0.075	14572	42	32
0.05	12718	43	32
0.025	9661	44	30
0.015	7619	45	30

Table 2: Security of *sample-then-lock* when restricting to bits that are unlikely to be masked.

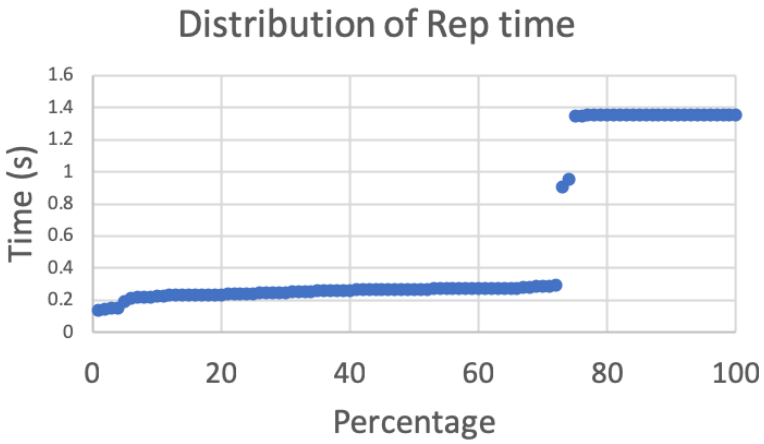


Figure 4: Cumulative distribution of time to run Rep for C implementation. Does not include time to regenerate subsets or load from hard disk. The "inflection point" is samples that did not unlock as they had to wait for all threads to finish all processing all lockers.

7.2 Correctness

We tested correctness across the data corpus with our Python implementation. Our implementation was tested with these parameters: 1) starting with 12000 bits that are unlikely to be masked 2) using a subset size of 43 bits. Specifically, Gen was run based on the first alphanumeric image from a particular iris, followed by Rep on all other images of that iris. Our target correctness was 50% across the corpus. Our observed mean correctness was higher at 60%. As expected correctness is highly correlated with the error rate of the underlying iris. This correlation is demonstrated in Figure 5.

Cautionary note for short iris transforms Above we claimed that our system can be instantiated with most iris image processing systems as input. There is one important caveat for this claim. The lock-then-sample construction needs many bits in the source to create many unique and dissimilar subsets. The construction computes $\ell = 10^6$ random subsets. For subsets of size k, we expect

$$\Pr[\text{correct}] = 1 - (1 - (1 - \text{error rate})^k)^\ell.$$

The work of Canetti et al. [CFP+16] incorrectly assumes that taking a bit without error does not effect the probability of the next bit having an error. This assumption is not true if the subset size approaches the total number of bits. To understand the smallest size iris transform that could be used with our construction, we consider selection sizes $sel = 2^{\{6,7,\ldots,13\}}$ and performed the following:

1. For each eye input an iris code w.
2. Restrict w to a random subset of size sel.
3. Run Gen with the restricted w.
4. Execute Rep on all images of the same iris (restricted to the locations in sel).
5. Record how frequently Rep is successful.
6. Average across images of an iris.
7. Average across irises.

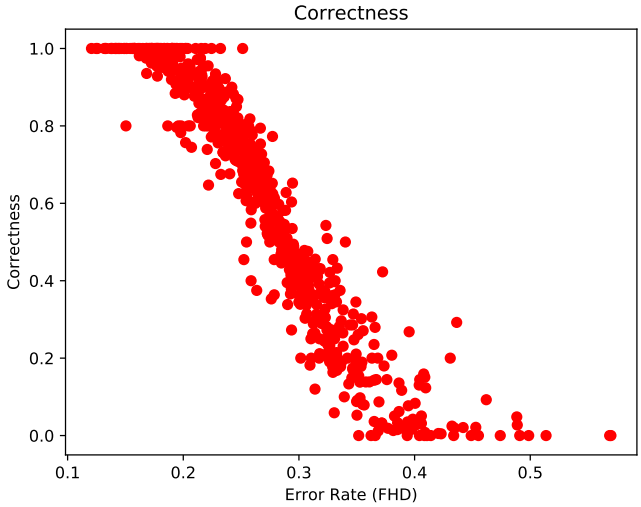


Figure 5: Correlation between correctness of Rep and the error rate of an individual's eye. The mean error rate across the corpus is 60%. Based on subset size of 43 bits with the 12000 bits that have probability of being masked of at less than 5%.

Subsample Size	64	128	256	512	2^{10}	2^{11}	2^{12}	2^{13}
Correctness (%)	17	41	52	56	60	60	60	60

Table 3: Relationship between the size of the input to Gen and the correctness of the system. The length of the input w is the first column and the probability that Rep unlocks is second column. The asymptotic behavior of sample-then-lock is not observed if the input to Gen is too small. Analysis performed using Python implementation.

The results are presented in Table 3. We found that $2^{10} = 1024$ bits were needed to achieve target correctness. For this experiment we analyzed how frequently the system was correct for each of the 356 eyes in the ND-0405 dataset. The Rep algorithm was then run on every other sample of the same eye in the corpus. (On average this was ≈ 200 trials per eye.) We only performed one trial of this experiment for each selection size. Restricting trials was necessary to keep the computation tractable.

8 Conclusion

In this work we have described the first key derivation system from the human iris that provides meaningful albeit modest security. An important feature of the system is that an individual can enroll with multiple devices and services. In addition, the necessary assumptions for security are clearly articulated in Theorem 1. The required distributional properties are directly evaluated in Table 2.

Our system is based on repeated evaluation of a cryptographic hash. We do not consider hash functions that are difficult to compute on GPUs. There are many promising memory hard hash functions such as scrypt [PJ16] and argon2i [Jos15]. Using these constructions in sample-then-lock is nontrivial as the hash function must be computed many times by the system. Ideally, one could use a hash function that is easy to compute in parallel with fast access to a large memory but hard for GPUs. We are unaware of any such candidates.

Our security is based on the error rates produced by the open source OSIRIS iris processing library. We expect higher security for a transform with lower error rates. Biometric authentication is a fact of life. This work explores how secure cryptographic techniques can be made for real biometrics. While our system does not achieve "cryptographic" security levels, we believe they are in reach. We hope this work encourages further research into the

iris and other biometric modalities. Lastly, porting to mobile platforms is a natural goal. We believe satisfactory performance on mobile devices requires new cryptographic and architectural techniques. We leave this as future work.

Acknowledgements

Mariem Ouni and Tyler Cromwell contributed to some software described in this work. We thank Leonid Reyzin and Alexander Russell for helpful discussions and insights.

References

[AA08] Rahib H Abiyev and Koray Altunkaya. Personal iris recognition using neural network. *International Journal of Security and its Applications*, 2(2):41–50, 2008.

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

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

[BA12] Marina Blanton and Mehrdad Aliasgari. On the (non-) reusability of fuzzy sketches and extractors and security improvements in the computational setting. *IACR Cryptology ePrint Archive*, 2012:608, 2012.

[BA13] Marina Blanton and Mehrdad Aliasgari. Analysis of reusability of secure sketches and fuzzy extractors. *IEEE transactions on information forensics and security*, 8(9-10):1433–1445, 2013.

[BC10] Nir Bitansky and Ran Canetti. On strong simulation and composable point obfuscation. In *Advances in Cryptology–CRYPTO 2010*, pages 520–537. Springer, 2010.

[BCC+07] Julien Bringer, Hervé Chabanne, Gérard Cohen, Bruno Kindarji, and Gilles Zémor. Optimal iris fuzzy sketches. In *Biometrics: Theory, Applications, and Systems, 2007. BTAS 2007. First IEEE International Conference on*, pages 1–6. IEEE, 2007.

[BCKP14] Nir Bitansky, Ran Canetti, Yael Tauman Kalai, and Omer Paneth. On virtual grey box obfuscation for general circuits. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, 2014.

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

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

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

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

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

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

[BHF08] Kevin W Bowyer, Karen Hollingsworth, and Patrick J Flynn. Image understanding for iris biometrics: A survey. *Computer vision and image understanding*, 110(2):281–307, 2008.

[BHF13] Kevin W Bowyer, Karen P Hollingsworth, and Patrick J Flynn. A survey of iris biometrics research: 2008–2010. In *Handbook of iris recognition*, pages 15–54. Springer, 2013.

[BHVOS12] Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *IEEE Symposium on Security and Privacy*, pages 553–567. IEEE, 2012.

[BLS12] Daniel J Bernstein, Tanja Lange, and Peter Schwabe. The security impact of a new cryptographic library. In *International Conference on Cryptology and Information Security in Latin America*, pages 159–176. Springer, 2012.

[Boy04] Xavier Boyen. Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM conference on Computer and communications security*, CCS '04, pages 82–91, New York, NY, USA, 2004. ACM.

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

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Obfuscating point functions with multibit output. In *Advances in Cryptology–EUROCRYPT 2008*, pages 489–508. Springer, 2008.

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

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

[CJKL18] Jung Hee Cheon, Jinhyuck Jeong, Dongwoo Kim, and Jongchan Lee. A reusable fuzzy extractor with practical storage size: Modifying Canetti et al.s construction. In *Australasian Conference on Information Security and Privacy*, pages 28–44. Springer, 2018.

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

[CS08] F Carter and A Stoianov. Implications of biometric encryption on wide spread use of biometrics. In *EBF Biometric Encryption Seminar (June 2008)*, 2008.

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

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

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

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

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

[DKRS06] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, and Adam Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 232–250. Springer Berlin Heidelberg, 2006.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008.

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

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

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

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

[FSS18] Benjamin Fuller, Sailesh Simhadri, and James Steel. Computational fuzzy extractors. `https://github.com/benjaminfuller/CompFE`, 2018.

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

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

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

[GM84] Alexander Grossmann and Jean Morlet. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM journal on mathematical analysis*, 15(4):723–736, 1984.

[Gol11] Oded Goldreich. A sample of samplers: A computational perspective on sampling. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 302–332. Springer, 2011.

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

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

[HBF09] Karen P Hollingsworth, Kevin W Bowyer, and Patrick J Flynn. The best bits in an iris code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):964–973, 2009.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

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

[HRvD+17] Charles Herder, Ling Ren, Marten van Dijk, Meng-Day Mandel Yu, and Srinivas Devadas. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. *IEEE Transactions on Dependable and Secure Computing*, 14(1):65–82, 2017.

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

[Jos15] Simon Josefsson. The memory-hard argon2 password hash function. *memory*, 2015.

[JW99] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Sixth ACM Conference on Computer and Communication Security*, pages 28–36. ACM, November 1999.

[KBK+11] Emile JC Kelkboom, Jeroen Breebaart, Tom AM Kevenaar, Ileana Buhan, and Raymond NJ Veldhuis. Preventing the decodability attack based cross-matching in a fuzzy commitment scheme. *Information Forensics and Security, IEEE Transactions on*, 6(1):107–121, 2011.

[KCK+08] Sanjay Kanade, Danielle Camara, Emine Krichen, Dijana Petrovska-Delacrétaz, and Bernadette Dorizzi. Three factor scheme for biometric-based cryptographic key regeneration using iris. In *Biometrics Symposium, 2008. BSYM'08*, pages 59–64. IEEE, 2008.

[KMSD17] Emine Krichen, Anouar Mellakh, Sonia Salicetti, and Bernadette Dorizzi. OSIRIS (open source for IRIS) reference system, 2017.

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

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

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

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

[NZ93] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, pages 43–52, 1993.

[PBF+08] P Jonathon Phillips, Kevin W Bowyer, Patrick J Flynn, Xiaomei Liu, and W Todd Scruggs. The iris challenge evaluation 2005. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, pages 1–8. IEEE, 2008.

[PJ16] Colin Percival and Simon Josefsson. The scrypt password-based key derivation function. Technical report, 2016.

[PPJ03] Salil Prabhakar, Sharath Pankanti, and Anil K Jain. Biometric recognition: Security and privacy concerns. *IEEE Security & Privacy*, 1(2):33–42, 2003.

[PRC15] Vishal M Patel, Nalini K Ratha, and Rama Chellappa. Cancelable biometrics: A review. *IEEE Signal Processing Magazine*, 32(5):54–65, 2015.

[PSJO+06] P. Jonathan Phillips, W. Todd Scruggs, Alice J. O'Toole, Patrick J. Flynn, Kevin W. Bowyer, Cathy L. Schott, and Matthew Sharpe. FRVT 2006 and ICE 2006 large-scale experimental results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

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

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[SHN+11] Fadi N Sibai, Hafsa I Hosani, Raja M Naqbi, Salima Dhanhani, and Shaikha Shehhi. Iris recognition using artificial neural networks. *Expert Systems with Applications*, 38(5):5940–5946, 2011.

[STP09] Koen Simoens, Pim Tuyls, and Bart Preneel. Privacy weaknesses in biometric sketches. In *IEEE Symposium on Security and Privacy*, pages 188–203. IEEE, 2009.

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

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

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

[WL18] Yunhua Wen and Shengli Liu. Robustly reusable fuzzy extractor from standard assumptions. In *Advances in Cryptology –ASIACRYPT*, 2018.

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

[WZ17] Daniel Wichs and Giorgos Zirdelis. Obfuscating compute-and-compare programs under lwe. *IACR Cryptology ePrint Archive*, 2017:276, 2017.

[Zha16] Mark Zhandry. The magic of elfs. In *Annual Cryptology Conference*, pages 479–508. Springer, 2016.

9 Additional Statistical analysis

This section contains additional statistical analysis that supports the conclusions in the body of the paper.

In Section 6 we computed the statistical distance between the observed interclass distribution and a binomial distribution. This statistical distance is displayed in Figure 6.

In Section 6.2, we showed how to optimized subset entropy by using the probability of location being masked. Roughly, this was using masking to predict future error rate. We also performed this analysis with future error rate directly. This analysis was performed with similar methodology as used to produce Table 2 with the following changes:

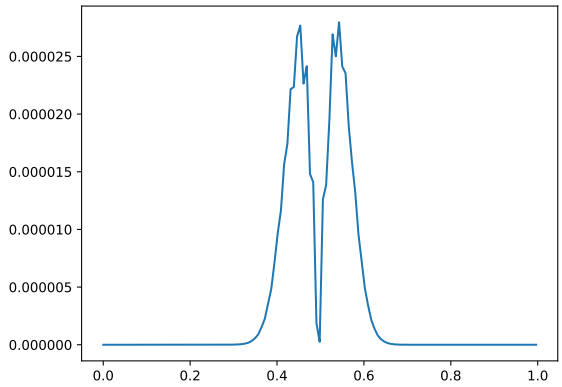


Figure 6: The statistical distance between the interclass comparisons of the base data set and a binomial distribution with the same mean and variance. The x-axis is the Hamming distance (resp. the normalized value of the binomial distribution). The y-axis is the statistical distance between the two distributions.

Error Rate Thres.	# of Bits	Subsample Size	Entropy
1	32768	32	29
0.475	32635	32	29
0.45	30610	33	29
0.425	26518	35	30
0.4	23417	37	29
0.375	19742	39	32
0.35	16594	41	32
0.325	13723	43	33
0.3	10659	45	31
0.275	6684	47	29
0.25	1876	50	20

Table 4: Security of *sample-then-lock* when restricting to bits that exhibit lower error rate across the data corpus.

1. We computed a vector $err\_rate \in [0,1]^{32768}$.
2. We consider a threshold $thres \in \{1, .99, .98, .97, ..., 0\}$.

The results are summarized in Table 4. There is a strong correlation between these locations with high masking probability and high error probability. In both cases we see the entropy of the system increase and then decrease. Our hypothesis is that this is due to having a small area to choose bits from increasing the probability of choosing nearby and correlated bits. This optimum point appears to occur at the same number of bits ≈ 12000.
Note: The reader may notice that the top entries in Tables 2 and 4 do not agree despite them corresponding to the same experiment: subsampling from all 32768 bits of the transform. This difference is due to noise introduced by choosing random subsets of size 32. The numbers differed by .3, the difference of 1 is due to rounding.

10 Further Connecting Image Processing and Cryptography

In this section we describe a second version of our system that passes additional information from the image processing to the cryptographic scheme. This additional information is used to select subsets that are less likely

to have an error. The security of this scheme is incomparable with the scheme presented in the body of this work. It has a higher "level" of security but requires an additional assumption about the way iris information is distributed (Assumption 1). Auxiliary information from a noisy source that guides error-correction is called *confidence information* and has been previously used in the context of physical unclonable functions [HRvD+17].

The confidence information The OSIRIS transform is formed by convolution of a 2D Gabor filter with the iris image at a number of positions. This convolution results in an array of complex numbers $a_i + b_i\mathbf{i}$, one for each location where the convolution was centered (the convolution is repeated at a number of positions). Then $a_i + b_i\mathbf{i}$ is converted to two bits, $\mathtt{sign}(a_i)$ and $\mathtt{sign}(b_i)$. The fractional Hamming distance is then computed across these bits.

The magnitude of each coefficient is correlated to error rate of this location in subsequent readings. That is, when $|a_i|$ is large a newly captured image of that iris is less likely to have an error in location i. This information can be used to guide subset selection. However, this confidence information may be correlated to w, thus revealing subsets that are biased by confidence information could leak about w.

We are not aware of any iris authentication that uses confidence information. Plaintext systems that compare w and w' achieve satisfactory false accept and false reject rates without such information. Our analysis only refers to a_i because we are using a transform where b_i is discarded. In order to use $|a_i|$ in our system, we must show the following properties:

1. Utility: There exists a correlation between $|a_i|$ and error rate in $\mathtt{sign}(a_i)$ in subsequent readings.
2. Correctness: It is possible to restrict to high confidence bits without negatively impacting correctness of *sample-then-lock*.
3. Security: That $|a_i|$ is not correlated with $\mathtt{sign}(a_i)$.[7]

We now explore these properties in turn.

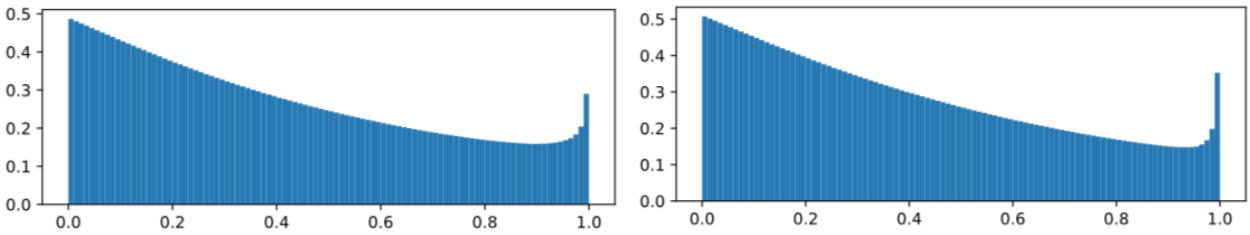
Utility We begin by showing that $|a_i|$ is correlated with the error rate of $\mathtt{sign}(a_i)$. That is, there exists some range of values $[ca, cb]$ such that

$$\Pr[\mathtt{sign}(a_i)' \neq \mathtt{sign}(a_i) || a_i| \in [ca, cb]] < \Pr[\mathtt{sign}(a_i)' \neq \mathtt{sign}(a_i)].$$

We first compute a frequency histogram of magnitude across the entire data corpus. We derive 100 buckets that represent the smallest 1% of magnitudes, $1\% - 2\%$, etc. using the following procedure:

1. Input an individual image from the data set.
2. Compute the vectors $\mathtt{sign}(a_i)$ and $|a_i|$.
3. Restrict to bits that are unlikely to be masked (where probability of masking is less than 5%).
4. For each bit i
   (a) Compute the bucket of $|a_i|$.
   (b) Compute the error rate of this bit with respect to other images of this eye.
   (c) Add this error rate to the bucket for i.
5. Average across images of this eye.
6. Average across different eyes.

[7] As will be discussed (in Assumption 1) $|a_i|$ may be correlated with $\mathtt{sign}(a_j)$ for $i \neq j$.



(a) Nonnegative values: Histo. of confidence vs. error rate. (b) Negative values: Histogram of confidence vs. error rate.

Figure 7: Histogram of correlation between error rate (y-axis) and magnitude of confidence information. Nonnegative confidence values are on the left histogram. Negative confidence values are in the right histogram. In each histogram the height of a bar is the number of bits with magnitudes in a range. The x-axis is the percentile of values with a confidence less than a given value.

We then compute a histogram of $|a_i|$ vs. the observed error rates $\Pr[\texttt{sign}(a_i)' \neq \texttt{sign}(a_i)]$. We found that this histogram differs based on whether a_i is nonnegative or negative. This determines whether the bit will be mapped to a 1 or 0 respectively. Thus, we present two different histograms. The analysis was recomputed restricting to only nonnegative bits (and then only negative bits).

These histograms are presented in Figure 7. The error rates for low confidence bits are as high as .50. Bits with magnitudes in the .85-.93 bins have average error rate as low as .13. We note that the two histograms are not the same, the negative histogram has a sharper "slope" and has a lower error rate in .85-.93 range. Thus, the confidence value is correlated with the error rate.

We incorporate confidence into our system by selecting a negative and a nonegative positive range that contain the same number of total bits across the corpus. That is, we select equal areas in the two histograms Figure 7.[8] This approach then yields two ranges $[p_{min}, p_{max}]$ and $[n_{min}, n_{max}]$.

As stated in Section 7.2 a transform of at 1024 bits is needed to maintain correctness. Intuitively, taking the minimum size range of confident bits should result in a system with minimum error rate and thus highest security. However, this intuition is incorrect. We performed the following analysis for $bins \in \{10, 20, 30, 40, 50, ..., 100\}$.

1. Compute a range $[p_{min,bins}, p_{max,bins}]$ that includes $bins$ with the smallest total integral across the data set.
2. Compute a range $[n_{min,bins}, n_{max,bins}]$ with the same number of entries as $[p_{min,bins}, p_{max,bins}]$ and minimum integral.
3. For each range perform 10 trials of:
   (a) For each iris in the data corpus:
      i. Take the first image according to some ordering (we used the first alphabetical file).
      ii. For this file compute the set $\mathcal{I}$ of bits that are unlikely to be masked and where a_i lies within the selected ranges.
      iii. Compute the intra and inter class fractional Hamming distance restricted to $\mathcal{I}$.
   (b) Average across all individuals
   (c) Compute the interclass mean μ_{bins} and variance σ_{bins}.
   (d) Calculate the degrees of freedom and entropy as above.
   (e) Compute the average min-entropy.

The result of this analysis is in Table 5. This table contains the computed ranges as well as the maximum subset size and the resulting entropy. 10 bins of the histogram corresponds to 1200 bits, 20 bins to 2400 bits and all 100 bins to 12000 bits.

[8] To do this we select a number of bins in the nonnegative histogram that have the minimum error rate and then select a range of negative values with the same total count and minimum integral. This usually means selecting part of a bin for negative values.

# Bits	$[n_{min}, n_{max}]$	$[p_{min}, p_{max}]$	Subset Size	Security
12000	[-21900, -26]	[6,21900]	43	35
10800	[-21900,-36]	[68,21900]	46	37
9600	[-21900, -99]	[131, 21900]	50	39
8400	[-21900, -165]	[198, 21900]	55	41
7200	[-2132,-228]	[263,2132]	60	43
6000	[-2132, -307]	[343, 2132]	65	45
4800	[-2132, -399]	[437, 2132]	70	44
3600	[-2132, -513]	[540, 1726]	75	42
2400	[-1726, -654]	[676,1522]	80	40
1200	[-1522, -879]	[828,1206]	83	35

Table 5: Security and subset sizes based on confidence ranges that include *# bits* in expectation. The positive and negative ranges are the real values that in expectation select that many bits.

The minimum number of bins, 10, produced the lowest effective error rate and thus the highest subsample size $k = 83$. Surprisingly, the entropy of the input subset is **not** maximized by taking ten bins. We found that including 6000 confident bits produced an input entropy of 45 with a subset size of 65. There are a number of possible causes for this behavior, i) highly confident bits tend to be in close proximity and thus are redundant or ii) when restricting to a smaller set a "bad" area of an image is more likely to be resulting in one trial with low entropy which will drop average min-entropy. We have not identified a root cause.

When we set ranges to select 6000 confident bits on average it is very rare for an image to have less than 1024 bits in this range. Thus, we do not expect correctness to be effected by restricting to bits in this range.

A priori, there is no reason to expect that high confidence bits have a similar probability density function as a binomial distribution. To establish this fact we first compute a histogram for inter/intra class distance. We do this using the following method:

1. Input an individual image from the data set.
2. Restrict the image to locations whose magnitudes lie within $[343, 2132]$ or $[-2132, -307]$ (and are not filtered by masking). Denote this location set by $\mathcal{I}$. Note that $|\mathcal{I}|$ may be larger or smaller than 6000.
3. Compute interclass and intraclass histograms for the entire corpus restricted to $\mathcal{I}$.
4. Average across images from the same iris.
5. Average across irises.

These histograms are in Figure 8. We note these histograms are visibly different from those in Figure 1. Interclass comparison have a mean of $\mu = .495$ and variance of $\sigma = .00264$. We compared the interclass distribution with a binomial distribution with mean $\mu = .495$ and variance $\sigma = .00264$. The total statistical distance is .087 and is shown in Figure 9.

Security Using confident bits has a subtle but important impact on security. The adversary is able to see which bits are selected in subsets (the locations are considered public). This means that the adversary can infer which bits are high confidence. We ensured that a bit is equally likely to be 1 and 0 based on being high confidence. However, this does not rule out other types of correlations. For example it might be that when bit 0 is high confidence it always takes the same value as bit 1000. Since subsets are now being computed based on the value of the iris (instead of randomly) this means that the following assumption is necessary for security of the scheme:

Assumption 1. *Let $(W, \mathcal{I})$ be a pair of random variables corresponding to sampling an iris and the location of that iris's high confidence bits. All information about $W_{\mathcal{I}} \stackrel{def}{=} \mathtt{sign}(\vec{a})_{\mathcal{I}}$ is contained in the values of other irises restricted to the locations in $\mathcal{I}$.*

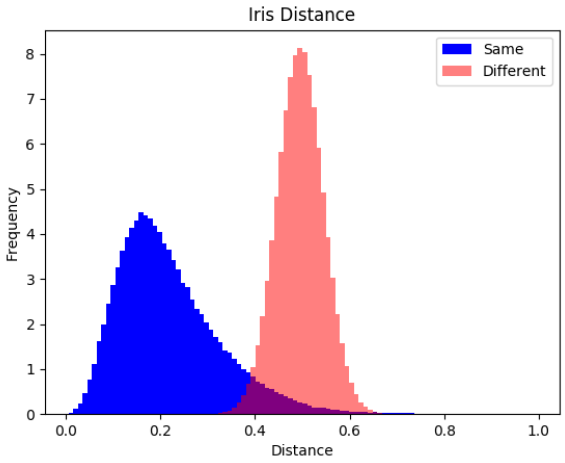


Figure 8: Distribution of distances for the data set restricted to high confidence bits. This figure is formed by selecting all locations of an image that have magnitude between $[343, 2132]$ or $[-2132, -307]$ and then computing the fractional Hamming distance with all other images in the data set restricted to these locations.

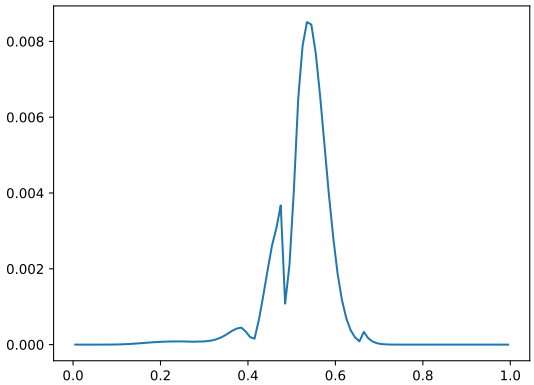


Figure 9: The statistical distance between the interclass comparisons restricted to "confident" bits compared to a binomial distribution with the same mean and variance. The x-axis is the Hamming distance (resp. the normalized value of the binomial distribution). The y-axis is the statistical distance between the two distributions.

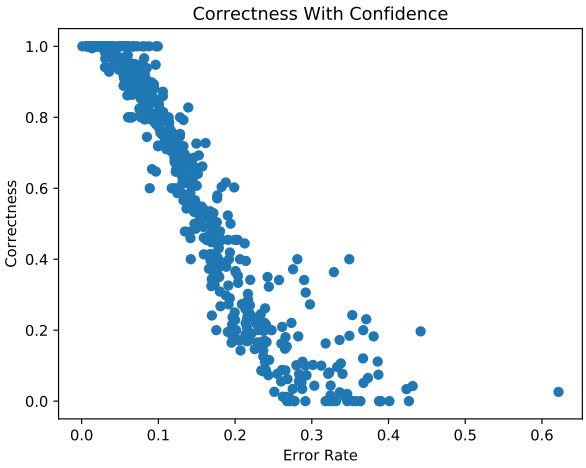


Figure 10: Correlation between correctness of Rep and the error rate of an individual's eye. The mean error rate across the corpus is 60%. Based on subset size of 81. Subsets selected by first 1) removing bits that are likely to be masked and 2) only selecting from bits in high confidence range.

This assumption also implies that it is safe to reuse confidence information. If all information about the value $\texttt{sign}(\vec{a})_I$ is contained in the value of other irises this immediately implies that having multiple high confidence sets is not helpful.

This assumption is in addition to the assumption that HMAC serves as a strong digital locker (Theorem 1) and that the entropy estimate in Table 5 is accurate.

Unlinkability While the construction in the body of the paper is unlinkable, this construction may not be, the selected subsets may be used to link enrollments of an individual's biometric.checking for the size of overlap between sampled positions.

Correctness For evaluating confidence, we tested the system with a confidence range expected to yield 2000 bits per iris. This is because this was the minimum size range where 90% of images had at least 1024 bits (where correctness of the scheme begins to degrade). We tested this range instead of 6000 confident bits because fewer bits are likely to hurt correctness. If correctness is preserved for 2000 bits it will also be preserved for 6000 bits.

Our Python implementation was tested with these parameters: 1) starting with 12000 bits that are unlikely to be masked 2) filter all bits whose confidence is outside both $[pa, pb] = [676.25, 1386.75]$ and $[na, nb] = [-1726.00, -673.12]$. and 2) using a subset size of 65 bits. Our mean correctness was 59%. As before, the correctness is highly correlated with the error rate of the underlying iris. This correlation is demonstrated in Figure 10.