Secure Biometrics Via Syndromes

Emin Martinian, Sergey Yekhanin, Jonathan S. Yedidia
Mitsubishi Electric Research Labs
Cambridge, MA 02139
{martinian,yedidia}@merl.com, yekhanin@mit.edu

Abstract

We consider the secure biometric storage problem and develop a solution using syndrome codes. Specifically, biometrics such as fingerprints, irises, and faces are often used for authentication, access control, and encryption instead of passwords. While it is well known that passwords should never be stored in the clear, current systems often store biometrics in the clear and are easily compromised. We describe the secure biometric storage problem by discussing the insecurities in current systems, the reasons why password hashing/encryption algorithms are unsuitable for biometrics, and propose a secure biometric storage framework based on syndrome codes and the Slepian-Wolf theorem.

1 Introduction

Biometrics such as fingerprints, voice prints, irises, and faces are becoming increasingly attractive tools for authentication and access control [1]. As replacements for passwords, biometrics have a number of advantages. First biometrics are inherently linked to the user and cannot be forgotten, lost, or given away. Second, appropriately chosen biometrics have high entropy and are less susceptible to brute force attacks than poorly chosen passwords. Finally, biometric authentication requires very little user expertise and can be used for widespread deployment.

Despite their advantages, however, care must be taken in securely storing biometrics. For example, Fig. 1 illustrates the biometric based encryption architecture used by commercially available products such as laptops, mobile phones, and portable hard drives. In such systems, an encryption key is derived from the user's biometric and used either to encrypt data stored on the device or to control access. To allow decryption or authorized access, the original biometric is stored on the device. To decrypt the data or gain access, a second biometric reading is taken from the user and compared with the biometric stored on the device. If the two biometrics are close enough, then access is granted or the original biometric is used to decrypt the data.

The major flaw in such systems is that the original biometric (which is the key) is stored on the device. Specifically, if an attacker gains physical access to the device (*e.g.*, by removing the hard drive from a laptop), then the attacker gains access to the original biometric. For data encryption systems, the attacker can then use the original biometric directly as the decryption key to compromise security. For access control systems, the attacker can use the biometric in a variety of ways ranging from creating fake biometrics matching the original, to modifying the output of the sensor designed to measure the user's biometric.
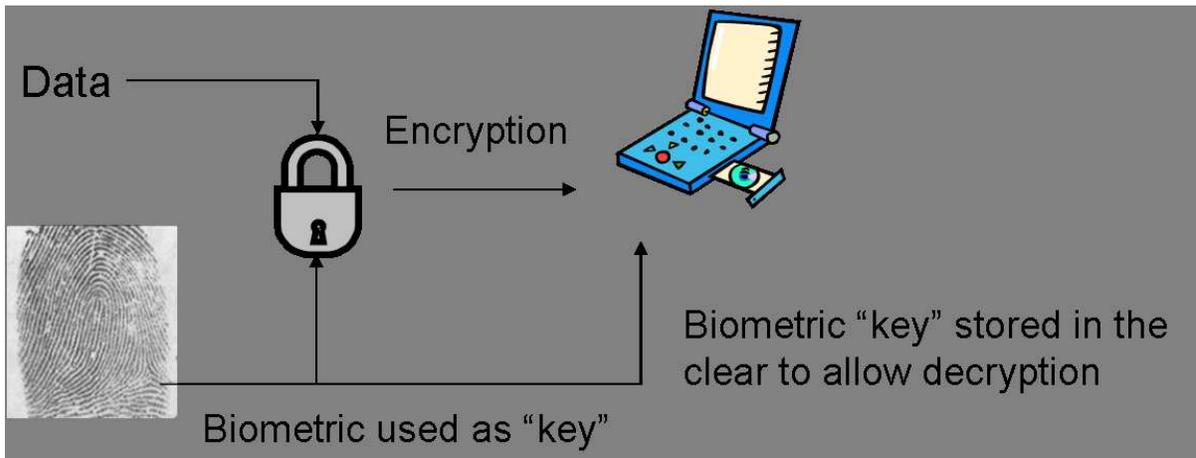
Figure 1: Typical (and insecure) biometric based encryption architecture an encryption key is derived from the user's biometric and used either to encrypt data stored on the device or to control access. To allow decryption or authorized access, the original biometric is stored on the device.

A potentially more serious security concern occurs when someone uses the same biometric in many systems or when many user biometrics are stored on a single system. Specifically, once an attacker acquires the original biometric, he can use it to compromise the security of many different systems. This potential for identify theft is much more serious for biometrics than passwords since if a password is stolen, it can be easily changed, but if a biometric such as a fingerprint is stolen it is difficult or impossible to change.

1.1 A Comparison To Passwords

Before delving into the secure biometrics problem, it is useful to consider a similar problem that arises while storing passwords. Specifically, if a system authenticates a user by requesting a password P, the system must have some way to verify whether the correct password is provided. The most straightforward (and insecure) way to do this would be to store P directly on the access control system. Of course, if P is stored in the clear, then any attacker who gains access to the password database completely compromises security.

Almost every password based system resolves this problem by storing the result of passing P through a hash function $\mathcal{H}(\cdot)$ instead of P and granting access only if the password P' provided by a user satisfies $\mathcal{H}(P) = \mathcal{H}(P')$. By using a one-way hash function that is difficult to invert, this approach prevents an attacker who gains access to the password database containing $\mathcal{H}(P)$ from obtaining P.

The key difference between a password and a biometric is that a password is always the same every time it is used while biometrics suffer from measurement noise and thus are different every time they are measured. Therefore storing the hash of a biometric instead of the biometric itself generally fails.

1.2 Background

To the best of our knowledge, Davida, Frankel, and Matt first considered the use of error correction coding as a solution to the secure biometrics problem [2]. Later, in 2002 Juels

and Sudan [3] introduced the idea of a fuzzy vault to formalize the use of error correcting codes for such applications. This spurred a variety of authors to apply the fuzzy vault idea to biometrics. In particular, several constructions for fingerprint biometrics were proposed [4–6] but all yielded high error rates. Also, several researchers have explored the cryptographic aspects of this problem in more depth [7–9]. On the information theory side, many authors have considered common randomness problems where different parties observe correlated random variables and attempt to agree on a shared secret key [10–13]. Our formulation and proposed solution builds is inspired by both sets of work.

1.3 Outline

Our main contributions in this paper are to formulate the secure biometrics problem from an information theoretic perspective, to develop a practical construction based on low density codes and belief propagation, and to report experimental results for iris biometrics.

We begin by defining a model for the secure biometric storage problem in Section 2. Section 3 describes our proposed solution from an information theoretic perspective, while Section 4 discusses how this framework can be implemented in practice and describes experimental results on iris biometrics. Finally, we close with some concluding remarks in Section 5.

2 Problem Model

Essentially, a biometric measurement acts as a shared secret between the user and an access control system. The secure biometric storage problem arises from the fact that in practical scenarios neither the user nor the access control has direct access to this shared secret.

Specifically, while the user essentially carries his biometric with him, he never knows the true value of the biometric and can only obtain noisy measurements. In particular, he can never recreate the exact measurement he provided to an access control system, but can only obtain noisy versions of the original measurement.[1]

Second while the access control system is designed only to grant access to a user possessing the shared secret, it cannot store this secret directly since doing so compromises security if an attacker gains knowledge of the system. Specifically, if an attacker gains knowledge of the access control system and the information it stores, the attacker should be unable to impersonate a different user or to gain access to encrypted data. Therefore the access control system must store an indirect version of the shared secret.

2.1 Measurement Model:

We model the secure biometric storage problem as follows. We imagine that each user has a "true biometric" $\mathbf{b}$, which is a vector of length n drawn according to some distribution $P_{\mathbf{b}}(\mathbf{b})$. Furthermore, we model the biometric for user i as statistically independent of the biometric for user j. To account for the inherent noise in obtaining a biometric

---

[1]Obviously, the user could store the exact measurement he provided to an access control system and carry this data around. But the main purpose of biometrics is precisely to avoid carrying or remembering a shared secret and to automatically generate the secret from the biometric. Consequently, our model assumes that the user does not carry or remember any information.

measurement $\mathbf{m}$, we model the measurement channel that maps a true biometric into a measured biometric via a conditional distribution $P_{\mathbf{m}|\mathbf{b}}(\mathbf{m}|\mathbf{b})$.

2.2 Enrollment and Authentication

We divide the problem into two phases: the enrollment phase, and the authentication phase.

2.2.1 Enrollment Phase:

In the enrollment phase, a user is selected and the true biometric $\mathbf{b}$ is determined by nature. Next, a measurement is made to obtain $\mathbf{m}$. A key extraction function $\mathbf{x}(\cdot)$ then maps the initial measurement into a "biometric key" $\mathbf{z} = \mathbf{x}(\mathbf{m})$, which acts as the shared secret between the user and the access control system. Next, $\mathbf{z}$ is mapped into the "secure biometric" S by an encoder $f(\cdot)$. The access control system only stores S and does not directly store $\mathbf{m}$ or $\mathbf{z}$.

2.2.2 Authentication Phase:

In the authentication phase, the same user requests access or decryption and provides another biometric measurement $\mathbf{m}'$. A decoder $g(\cdot, \cdot)$ combines the secure biometric S with the measured biometric $\mathbf{m}'$ and either produces an estimate of the shared secret $\hat{\mathbf{z}}$ or a special symbol $\varnothing$ indicating that authentication fails. Once the decoder has produced the shared secret it can determine whether to grant access, decrypt data encrypted with the shared secret, *etc.*

2.3 Performance Measures

2.3.1 Probability of Authentication Failure

We define $\mathcal{E}_{\text{auth}}$ as the event that the decoder fails to recover the shared secret for a legitimate user:

$$\mathcal{E}_{\text{auth}} \triangleq \{\mathbf{z} \neq g(\mathbf{m}', f(\mathbf{m}))\}. \tag{1}$$

Ideally, $\Pr[\mathcal{E}_{\text{auth}}]$ should be as small as possible. [2]

2.3.2 Probability of Security Failure

Most cryptographic attacks generally make many attempts to guess the desired secret and so measuring the probability that a *single* attack succeeds is not particularly meaningful. Instead, security is usually assessed by measuring how many attempts an attack algorithm must make to have a reasonable probability of success. As a result, security failure is somewhat more complicated to define than authentication failure.

Let $\mathcal{L} = \mathcal{A}_t[\cdot]$ be a list of 2^{tn} guesses for S produced by an attack algorithm $\mathcal{A}_t[\cdot]$ that uses knowledge of $\mathbf{x}(\cdot)$, $f(\cdot)$, $g(\cdot, \cdot)$, and S, but not $\mathbf{m}$. We define $\mathcal{E}_{\text{sec},t}$ as the event

---

[2]When the biometric is used to generate an encryption key $\Pr[\mathcal{E}_{\text{auth}}]$ is the appropriate quantity. But in access control applications, $\mathbf{z}$ may be compared to $\mathbf{x}(\mathbf{m}')$ to determine if access should be granted. In such cases, further authentication errors are possible, but we do not consider these here due to space constraints and plan to address this issue in future work.

that the list contains the biometric used to produce S:

$$\mathcal{E}_{\mathrm{sec},t} \triangleq \{\mathbf{z} \in \mathcal{A}_t\left[\mathbf{x}(\cdot), f(\cdot), g(\cdot, \cdot), S\right]\}. \tag{2}$$

We refer to a scheme with $\Pr[\mathcal{E}_{\mathrm{sec},t}] = \epsilon$ as having $n \cdot t$ bits of security with confidence $1 - \epsilon$ since with probability $1 - \epsilon$ an attacker must search a keyspace of $n \cdot t$ bits to crack the system security.

2.4 Goals

The goal of the secure biometrics problem is to construct an encoder and decoder to obtain the best combination of robustness (as measured by $\mathcal{E}_{\mathrm{auth}}$) and security (as measured by $\mathcal{E}_{\mathrm{sec},t}$). In general, there will be a trade-off between these two. For example, if $\Pr[\mathcal{E}_{\mathrm{sec},40}] = \epsilon$ and $\Pr[\mathcal{E}_{\mathrm{auth}}] = 2^{-30}$ at one operating point, increasing the security might yield another operating point at $\Pr[\mathcal{E}_{\mathrm{sec},50}] = \epsilon$ and $\Pr[\mathcal{E}_{\mathrm{auth}}] = 2^{-20}$. A similar trade-off between the probability of false alarm and the probability of missed detection arises in many detection problems and is generally characterized by the receiver operating characteristic (ROC).

For various idealized models, as the block length n increases, it is possible to show that $\Pr[\mathcal{E}_{\mathrm{sec},t}]$ undergoes a phase transition in t. Specifically, there will exist a value of t below which $\Pr[\mathcal{E}_{\mathrm{sec},t}]$ goes to zero as n increases and above which this probability goes to one as n increases. Consequently, we find it more meaningful to measure security by focusing on the maximum value of t yielding a fixed $\Pr[\mathcal{E}_{\mathrm{sec},t}]$ than by focusing on $\Pr[\mathcal{E}_{\mathrm{sec},t}]$ directly.

For authentication failure, the error exponent $-(1/n)\log \Pr[\mathcal{E}_{\mathrm{auth}}]$ provides a similar logarithmic performance measure. So for a fixed $\epsilon > 0$, we define the security-robustness region $\mathcal{R}_\epsilon$ as the set of pairs (t, γ) where t bits of security are possible with an authentication failure probability of γ:

$$\mathcal{R}_\epsilon = \{(t, \gamma)|\Pr[\mathcal{E}_{\mathrm{sec},t}] \leq \epsilon, \gamma \leq -(1/n)\log \Pr[\mathcal{E}_{\mathrm{auth}}]\}. \tag{3}$$

The goal of the secure biometrics problem is to maximize $\mathcal{R}_\epsilon$. Specifically, as illustrated in Fig. 2 we will consider one secure biometric system to be superior to another secure biometrics system if the security-robustness region for the former is strictly larger than the latter.

3 Secure Biometrics Via Information Theory

To illustrate the intuition for our proposed solution, we first describe a system and estimate its performance using information theory and random codes. Later we describe a practical implementation based on low density parity check codes.

3.1 Enrollment Phase

We model the key extraction function $\mathbf{x}(\cdot)$ as a quantizer and denote the quantized output as $\hat{\mathbf{x}} = \mathbf{x}(\mathbf{m})$ where $\hat{\mathbf{x}}$ is a vector of length n with each component taking values in a finite set $\hat{\mathcal{X}}$. For example, in an iris system, $\mathbf{x}(\cdot)$ could be a function mapping the continuous image of an iris into a vector of bits.
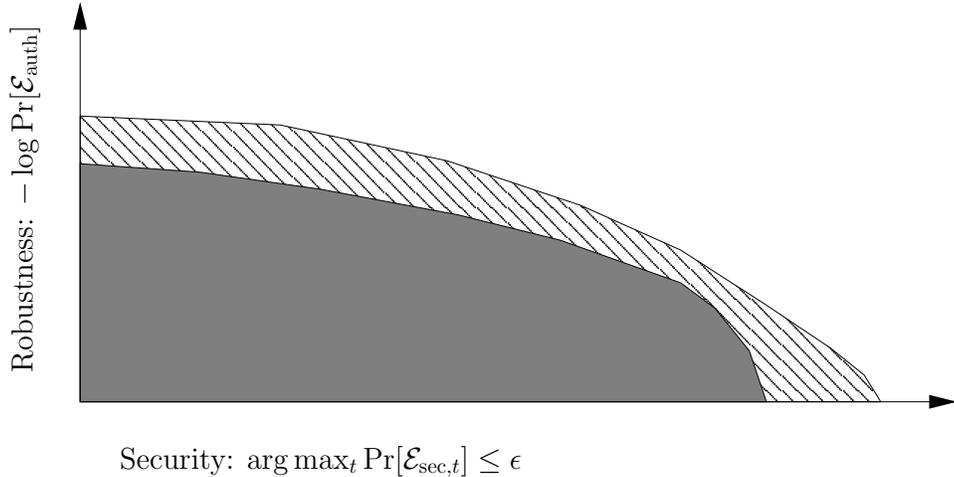
Figure 2: Example security-robustness regions. The horizontal axis represents the number of bits of security while the vertical axis represents robustness. The system corresponding to the striped region has a larger security-robustness region and is therefore superior to the system corresponding to the shaded region.

For the encoder, we use a rate R random hash function. Specifically, we imagine that $f(\cdot)$ is constructed by randomly assigning each possible quantized vector $\hat{\mathbf{x}} \in \hat{\mathcal{X}}^n$ to an integer selected uniformly from $\{1, 2, \ldots, 2^{nR}\}$. The secure biometric is $S = f(\mathbf{x}(\mathbf{m}))$.

3.2 Authentication Phase

In the authentication phase, a user provides a biometric $\mathbf{m}'$ and claims to be user i. The decoder $g(S, \mathbf{m}')$ searches for a quantized vector $\hat{\mathbf{z}} \in \hat{\mathcal{X}}^n$ such that is $\hat{\mathbf{z}}$ is jointly typical[3] with $\mathbf{m}'$ and matches the hash (*i.e.*, $f(\hat{\mathbf{z}}) = S$). If a unique $\hat{\mathbf{z}}$ is found, then the decoder outputs this result. Otherwise, an authentication failure is declared and the decoder returns $\varnothing$.

3.3 Analysis

First, according to the Slepian-Wolf Theorem [14], the decoder will succeed with probability approaching 1 as n increases provided that $R > H(\hat{\mathbf{x}}|\hat{\mathbf{z}})$. Thus, $\Pr[\mathcal{E}_{\text{auth}}]$ approaches zero and for long block lengths, authentication failures become increasingly unlikely. The theory of error exponents for channel coding [15] can be used to derive sharper results such as the asymptotic form of $\log \Pr[\mathcal{E}_{\text{auth}}]$.

Next we consider the probability of successful attack, *i.e.*, how well an attacker can estimate $\mathbf{z}$ given the secure biometric S. According to the asymptotic equipartition property [14], under fairly mild technical conditions it is possible to show that conditioned on $S = f(\hat{\mathbf{x}})$, $\hat{\mathbf{x}}$ is approximately uniformly distributed over the typical set of size $2^{H(\hat{\mathbf{x}}|S)}$. We can compute $H(\hat{\mathbf{x}}|S)$ via

$$H(\hat{\mathbf{x}}|S) = H(\hat{\mathbf{x}}, S) - H(S) = H(\hat{\mathbf{x}}) - H(S) = H(\hat{\mathbf{x}}) - nR \qquad (4)$$

and use this to estimate the security of a system provided $H(\hat{\mathbf{x}})$ is known.

---

[3]For readers less familiar with information theory, two vectors $\mathbf{a}$ and $\mathbf{b}$ are jointly typical with respect to a distribution $p(\mathbf{a}, \mathbf{b})$, if the empirical entropies of $\mathbf{a}$, $\mathbf{b}$, and $(\mathbf{a}, \mathbf{b})$ are close to the true entropies with respect to $p(\cdot, \cdot)$. See [14] for details.

We conclude this section with the following proposition which yields an inner bound to the security-robustness region:

Proposition 1. *Let* **b** *and* **m** *be sequences of random variables generated according to the i.i.d. distribution*

$$p_{\mathbf{m},\mathbf{b}}(\mathbf{m},\mathbf{b}) = \prod_{i=1}^{n} p_{m|b}(m_i|b_i)p_b(b_i), \tag{5}$$

then for any $\epsilon > 0$ as $n \to \infty$ we can find the following inner bound to the security-robustness region $\mathcal{R}_\epsilon$ by taking a union over all possible key extraction functions $x(\cdot)$ and rates R of (4) and the Slepian-Wolf coding error exponent [16]:

$$\mathcal{R}_\epsilon \supset \cup_{x(\cdot),R}\{t,\gamma|t \leq H(\hat{x}) - R, \gamma \leq E_{\mathrm{SW},p_{\hat{x},m}(\cdot,\cdot)}(R)\} \tag{6}$$

where $E_{\mathrm{SW},p(\cdot,\cdot)}(R)$ is the Slepian-Wolf coding coding error exponent [16] for a source pair with distribution $p(\cdot,\cdot)$ at rate R.

4 A Secure Biometrics System Using Syndromes

In this section, we describe a prototype implementation of our secure biometrics system for iris recognition and discuss experimental results on the CASIA [17] database. Specifically, we replace the random hash function output of 3 with the syndrome from an LDPC code.

4.1 Enrollment Phase

Our encoder $f(\cdot)$ consists of the following steps:

1. We start with an image of a user's eye, detect the location of the iris, unwrap it into a rectangular region, and use a bank of Gabor filters to extract a bit sequence, which we denote as **m**. These steps are performed using the matlab implementation from [18].

2. Our key extraction procedure $\mathbf{x}(\cdot)$ produces **z** from **m** by discarding the bits at certain fixed positions that we determined to be unreliable based on our training data. The resulting $\mathbf{z} = \mathbf{x}(\mathbf{m})$ consists of the 1806 most reliable bits from **z**.

3. We map the bit string **z** into the secure biometric S by computing the syndrome of **z** with respect to a low density parity check (LDPC) code. Specifically, we select a random parity check matrix **H** from a good low rate degree distribution obtained via density evolution [19] and compute $S = \mathbf{H} \cdot \mathbf{z}$.

4.2 Authentication Phase:

Our decoder $g(\cdot,\cdot)$ consists of the following steps:

1. As in the enrollment phase, we start with an image of a user's eye, detect the location of the iris, unwrap it into a rectangular region, and use a bank of Gabor filters to extract a bit sequence, which we denote as $\mathbf{m}'$. These steps are performed using the matlab implementation from [18].

2. We again discard the least reliable bits as in step 1 of Section 4.1 and use the resulting $\mathbf{x}(\mathbf{m}')$ as the input to a belief propagation decoder that attempts to find a sequence $\hat{\mathbf{z}}$ satisfying $\mathbf{H} \cdot \hat{\mathbf{z}} = S$.

3. If the belief propagation decoder succeeds, then the output of $g(\mathbf{m}', S)$ is the resulting $\hat{\mathbf{z}}$. Otherwise, an authentication failure is declared and the output of $g(\mathbf{m}', S)$ is $\varnothing$.

4.3 Inter-bit Memory and New BP Rules

We found that the bit sequences extracted from the irises in our database contained significant inter-bit correlation. Specifically, let $p_{i,j}$ be the probability of an iris bit taking the value i followed by another bit with the value j. If the bits extracted from an iris were independent and identically distributed, we would expect $p_{i,j} = 1/4$ for all $(i,j) \in \{0,1\}^2$. Instead we measured the following probabilities:

$$p_{0,0} = 0.318, \; p_{0,1} = 0.166, \; p_{1,0} = 0.166, \; p_{1,1} = 0.349. \tag{7}$$

When we ignored this memory, we obtained unacceptably poor performance and therefore we modified our belief propagation decoder to exploit memory. Specifically, conventional belief propagation methods pass messages from variable nodes to check nodes and back again using sum-product formulas (*e.g.*, as described in [20]). In contrast, we added "correlation nodes" between every pair of variable nodes in the factor graph. Since the correlation nodes are essentially just a different type of check node, adding them preserves the bipartite structure of the factor graph.

We describe our changes to the traditional sum-product rules by defining new message update equations for the messages to and from the new correlation nodes (the message update equations for messages to and from the other nodes are unchanged). In the notation of Kschischang et al., if $\mu_{f \to y}(x)$ is the incoming message for state x to variable node y from check f and $\mu_{y \to f}(x)$ is the outgoing message from variable node y to check or correlation node f, then the message from a variable node y to a neighboring check or correlation node f is

$$\mu_{y \to f}(x) = \prod_{c \in \mathcal{N}(y) \backslash y} \mu_{c \to y}(x) \tag{8}$$

where $\mathcal{N}(a) \backslash b$ is the set of nodes that are neighbors of a excluding node b. The only change from [20] is that y may now refer to correlation nodes as well as traditional check nodes. The message from a check node f to a variable node y is exactly the same as the usual check to variable rule (see [20] for details). The main change is the rule for an outgoing message from a correlation node f to the variable node y on its left

$$\mu^{(L)}_{y \leftarrow f}(0) = p_{0,0} \cdot \mu_{f \to y}(0) + p_{0,1} \cdot \mu_{f \to y}(1) \text{ and } \mu^{(L)}_{y \leftarrow f}(1) = p_{1,0} \cdot \mu_{f \to y}(0) + p_{1,1} \cdot \mu_{f \to y}(1) \tag{9}$$

and the message from a correlation node f to the variable node y on its right

$$\mu^{(R)}_{y \leftarrow f}(0) = p_{0,0} \cdot \mu_{f \to y}(0) + p_{1,0} \cdot \mu_{f \to y}(1) \text{ and } \mu^{(R)}_{y \leftarrow f}(1) = p_{0,1} \cdot \mu_{f \to y}(0) + p_{1,1} \cdot \mu_{f \to y}(1). \tag{10}$$

4.4 Experimental Results

To test our prototype system, we used the CASIA iris database [17]. The iris segmentation algorithm used by our system was only able to correctly detect the iris in 624 out

of 756 images [18, Chapter 2.4]. Since our goal was to focus on the secure biometrics problem not on iris segmentation, we worked only with the 624 iris that were segmented successfully. Furthermore, we separated these 624 into 312 training images (*e.g.*, in order to measure parameters such as $p_{i,j}$) and 312 test images (on which we report results).

Fig. 3 reports our performance results on the 312 image test set from the CASIA iris database. The horizontal axis represents security (the t parameter in Section 2.3.2) while the vertical axis represents $-\log_2 \Pr[\mathcal{E}_{\text{auth}}]$ where $\Pr[\mathcal{E}_{\text{auth}}]$ is the probability of authentication failure for a legitimate user. Better systems correspond to points in the upper right, but as the figure shows, there is a trade-off between security and robustness. Specifically, if a rate R LDPC code is used in step 3 of Section 4.1, then S contains nR bits. Under the idealized model where the iris data consists of i.i.d. Bernoulli$(1/2)$ bits, our approach yields approximately $1806 \cdot R$ bits of security with confidence approaching 1. Decreasing R yields higher security, but lower robustness so the security-robustness region can be estimated by varying this parameter.

Note that if the biometric is stored in the clear then we obtain a probability of authentication failure of 0.0012 (*i.e.*, the leftmost point in the graph).[4] Thus, we see that with essentially no change in the probability of authentication failure relative to an insecure scheme, we achieve almost 50 bits of security.
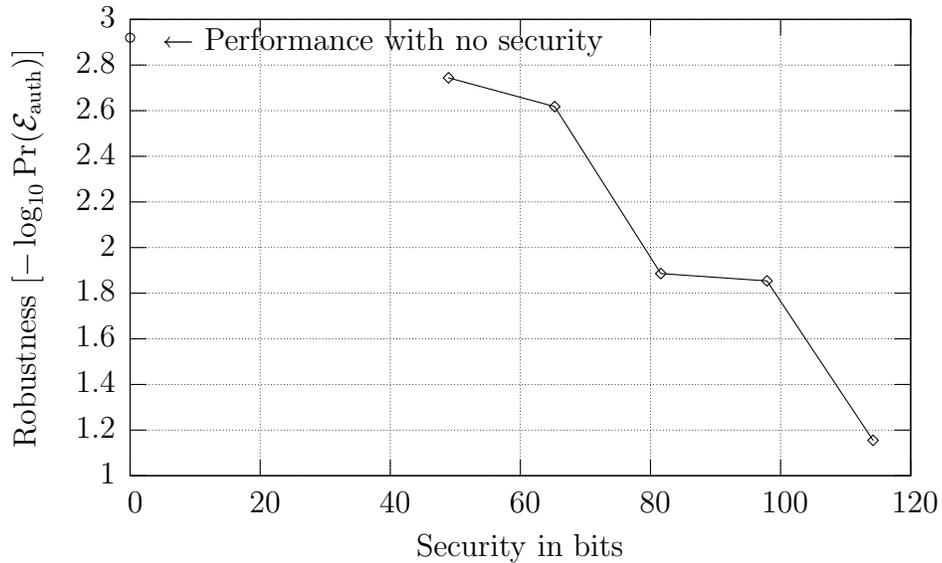

Figure 3: Performance results on 312 images in our test set from the CASIA iris database. The horizontal axis represents security (the t parameter in Section 2.3.2) while the vertical axis represents $-\log_2 \Pr[\mathcal{E}_{\text{auth}}]$ where $\Pr[\mathcal{E}_{\text{auth}}]$ is the probability of authentication failure for a legitimate user. The solid line shows performance for our system, while the leftmost point is authentication performance without security.

4.4.1 Information Theoretic vs. Computational Security

We note that while we believe our approach provides a secure method of storing biometrics, the true level of security is somewhat difficult to evaluate. Specifically, the original

---

[4]When a biometric key is stored in the clear there can be no error in recovering the key for decryption. Instead, the leftmost point in Fig. 3 represents the probability of false rejection for an access control system that grants access if the measured biometric is close enough to the biometric stored in the clear.

length of the bit sequence extracted from an iris in our system is 1806 and the length of the syndrome produced by our encoder is $1806 - t$ where t is a point on the horizontal axis of Fig. 3. *If* the original biometrics were independent and identically distributed sequences of uniformly random bits, then the probability of guessing the true biometric from the syndrome would be about 2^{-t} (*i.e.*, information theoretic security of t bits).

As we point out in Section 4.3, however, there is significant inter-bit memory in iris biometrics. In particular, according to the statistics for $p_{i,j}$ that we measured, the entropy of an 1806 bit measurement is only about 90% of 1806. Consequently, if our syndrome was a truly random hash of the input biometric, it would contain $1806 - t$ bits of information about the biometric. Since $1806 - t > 90\%$ for all reasonable values of $\Pr[\mathcal{E}_{\text{auth}}]$, this suggests that an attacker with unbounded computational resources might be able to determine the true syndrome more quickly than by randomly searching a key space of size 2^t.

We are not aware of any computationally feasible methods of improving upon random guessing. The most obvious method to attempt would be for the attacker to guess a random bit string for the measured biometric and attempt belief propagation decoding. Such a strategy would probably require fewer attempts than a random guessing strategy, but the time required for belief propagation decoding might outweigh such an advantage.

In practice, we do not believe this issue is a serious impediment in real systems since similar security issues are present for virtually all other encryption systems. For example, it is well known that the widely used RSA encryption algorithm can be broken more quickly than by exhaustively searching the key space. Therefore, our main purpose in raising this issue is to spur further analysis for more accurate security estimates.

5   Concluding Remarks

In this paper, we discussed a model for the secure biometrics problem, described an information theoretic solution based on the Slepian-Wolf theorem, proposed a practical implementation using syndrome codes, and discussed experimental results of a prototype on iris biometrics. In particular, for the CASIA database, our proposed solution achieves about 50 bits of security at essentially the same authentication error rates as an insecure system. Higher levels of security can be achieved if larger authentication error rates are allowed. To our knowledge, these results represent the first experimental results for any method of securing iris data and the first method of securing any biometric data that can provide security at the roughly same authentication error rate as an insecure system.

References

[1] A. K. Jain, S. Pankanti, S. Prabhakar, L. Hong, and A. Ross, "Biometrics: a grand challenge," *Proc. Interntaional Conference on Pattern Recognition*, vol. 2, pp. 935–942, August 2004.

[2] G. I. Davida, Y. Frankel, and B. J. Matt, "On enabling secure applications through off-line biometric identification," in *Proc. IEEE Symposium on Security and Privacy*, pp. 148–157, May 1998.

[3] A. Juels and M. Sudan, "A fuzzy vault scheme," in *Proc. International Symposium on Information Theory*, p. 408, 2002.

[4] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcardbased fingerprint authentication," in *Proc ACM SIGMM workshop on biometrics methods and applications*, 2003.

[5] S. Yang and I. M. Verbauwhede, "Secure fuzzy vault based fingerprint verification system," in *Asilomar Conference on Signals, Systems, and Computers*, vol. 1, pp. 577–581, November 2004.

[6] U. Uludag and A. Jain, "Fuzzy fingerprint vault," in *Proc. Workshop: Biometrics: Challenges Arising from Theory to Practice*, pp. 13–16, August 2004.

[7] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Adv. in Cryptology — Eurocrypt 2004, LNCS*, vol. 3027, pp. 523–540, Springer-Verlag, 2004.

[8] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith, "Secure remote authentication using biometric data," in *Advances in Cryptology (EUROCRYPT 2005)*, vol. 3494 of *LNCS*, pp. 147–163, Springer-Verlag, 2005.

[9] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *EUROCRYPT 2005: 457-473*, pp. 457–473, 2005.

[10] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography I: Secret sharing," *IEEE Transactions on Information Theory*, vol. 39, pp. 1121–1132, July 1993.

[11] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography II: CR capacity," *IEEE Transactions on Information Theory*, vol. 44, pp. 225–240, January 1998.

[12] S. Venkatesan and V. Anantharam, "The common randomness capacity of a network of discrete memoryless channels," *IEEE Transactions on Information Theory*, vol. 46, pp. 367–387, March 2000.

[13] I. Csiszar and P. Narayan, "Secrecy capacities for multiple terminals," *IEEE Transactions on Information Theory*, vol. 50, pp. 3047–3061, December 2004.

[14] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: John Wiley & Sons, Inc., 1991.

[15] R. G. Gallager, *Information Theory and Reliable Communication*. John Wiley and Sons, Inc., 1968.

[16] I. Csiszar, "Linear codes for sources and source networks: Error exponents, universal coding," *IEEE Transactions on Information Theory*, vol. 28, pp. 585–592, July 1982.

[17] "CASIA iris image database collected by institute of automation, chinese academy of sciences."

[18] L. Masek, "Recognition of human iris patterns for biometric identification." Bachelors Thesis, University of Western Australia, 2003.

[19] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, pp. 619–637, February 2001.

[20] F. R. Kschischang, B. J. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, pp. 498–519, February 2001.