MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Securing Biometric Data

Anthony Vetro, Stark Draper, Shantanu Rane, Jonathan Yedidia

TR2008-081 December 2008

Abstract

Securing access to physical locations and to data if of primary concern in many personal, commercial, governmental and military contexts. Classic solutions include carrying an identifying document or remembering a password. Problems with the former include forgeries while problems with the latter include poorly-chosen or forgotten passwords. Computer-verifiable biometrics, such as fingerprints and iris scans, provide an attractive alternative to conventional solutions. Biometrics have the advantage that, unlike passwords, they do not have to be remembered and, unlike identifying documents, they are difficult to forge. However, they have characteristics that raise new security challenges.

Edited Book on Distributed Source Coding

Securing Biometric Data

Anthony Vetro, Stark C. Draper, Shantanu Rane, and Jonathan S. Yedidia

***Abstract*— This chapter discusses the application of distributed source coding techniques to biometric security. A Slepian-Wolf coding system is used to provide a secure means of storing biometric data that provides robust biometric authentication for genuine users and guards against attacks from imposters. A formal quantification of the trade off between security and robustness is provided as a function of the Slepian-Wolf coding rate. Prototype secure biometric designs are presented for both iris and fingerprint modalities. These designs demonstrate that it is feasible to achieve information-theoretic security while not significantly compromising authentication performance (measured in terms of false-rejection and false-acceptance rates) when compared to conventional biometric systems. The methods described in this chapter can be applied to various architectures, including secure biometric authentication for access control and biometric-based key generation for encryption.**

***Index Terms*— Biometric, security, Slepian-Wolf coding, syndrome, iris, fingerprint, error correcting codes, LDPC codes, belief propagation decoding, statistical model, feature extraction, feature transformation, minutiae, helper data, fuzzy vault, factor graph, access control, authentication, encryption, cryptographic hash, robust hash, false accept rate, false reject rate, equal error rate.**

I. Introduction

A. Motivation and Objectives

Securing access to physical locations and to data is of primary concern in many personal, commercial, governmental and military contexts. Classic solutions include carrying an identifying document or remembering a password. Problems with the former include forgeries while problems with the latter include poorly-chosen or forgotten passwords. Computer-verifiable biometrics, such as fingerprints and iris scans, provide an attractive alternative to conventional solutions. Biometrics have the advantage that, unlike passwords, they do not have to be remembered and, unlike identifying documents, they are difficult to forge. However, they have characteristics that raise new security challenges.

This work was performed while all authors were with the Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge MA 02139. This work was presented in part at the Allerton Conf. Comm. Control Comput., Monticello IL, Sept 2005; in part at the UCSD Workshop on Inform. Theory and Apps., San Diego CA, Jan 2007; in part at the IEEE Int. Conf. Acoust. Speech Sig. Proc., Honolulu HI, Apr 2007; in part at the IEEE Int. Symp. Inform. Theory, Toronto CA, Jun 2008; and in part at the Comp. Vision Pattern Recog. (CVPR) Biometrics Workshop, Anchorage AL, Jun 2008.

A. Vetro is with the Mitsubishi Electric Reserach Laboratories, Cambridge MA, 02139 USA (e-mail: avetro@merl.com).

S. Draper is with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison WI 53706 USA (e-mail: sdraper@ece.wisc.edu).

S. Rane is with the Mitsubishi Electric Reserach Laboratories, Cambridge MA, 02139 USA (e-mail: rane@merl.com).

J. Yedida is with the Mitsubishi Electric Reserach Laboratories, Cambridge MA, 02139 USA (e-mail: yedidia@merl.com).

The key characteristic differentiating biometrics from passwords is measurement noise. Each time a biometric is measured, the observation differs, at least slightly. For example, in the case of fingerprints, the reading might change because of elastic deformations in the skin when placed on the sensor surface, dust or oil between finger and sensor, or a cut to the finger. Biometric systems must be robust to such variations. Biometric systems deal with such variability by relying on pattern recognition. To perform recognition in current biometric systems, the biometric measured at enrollment is stored on the device for comparison with the "probe" biometric collected later for authentication. This creates a security hole: an attacker who gains access to the device also gains access to the biometric. This is a serious problem since, in contrast to passwords or credit card numbers, an individual cannot generate new biometrics if their biometrics are compromised.

The issue of secure storage of biometric data is the central design challenge that is addressed in this chapter. Useful insight into desirable solution characteristics can be gained through consideration of password-based authentication. In order to preserve the privacy of passwords in the face of a compromised database or personal computer, passwords are not stored "in-the-clear". Instead, a cryptographic "hash" of one's password is stored. The hash is a scrambling function that is effectively impossible to invert. During authentication a user types in their password anew. Access is granted only if the hash of the new password string matches the stored hash of the password string entered at enrollment. Because of the non-invertibility of the hash, password privacy is not compromised even if the attacker learns the stored hash. Unfortunately, the variability inherent to biometric measurement means that this hashing solution cannot be directly applied to biometric systems – enrollment and probe hashes would hardly ever match.

The aim of the secure biometric systems detailed in this chapter is to develop a hashing technology robust to biometric measurement noise. In particular, we focus on an approach that uses "syndrome" bits from a Slepian-Wolf code [1] as a "secure" biometric. The syndrome bits on their own do not contain sufficient information to deduce the user's enrollment biometric (or "template"). However, when combined with a second reading of the user's biometric, the syndrome bits enable the recovery and verification of the enrollment biometric. A number of other researchers have attempted to develop secure biometric systems with similar characteristics, and we will review some of these proposals in Section II.

B. Architectures and System Security

There are two fundamental applications for secure biometric technology: access control and key management. In the former,

the system modulates access through inspection of a candidate user's biometric. In the latter, the system objective is to extract a stable encryption key from the user's biometric. While access-control and key-management are different goals, the syndrome-encoding and recovery techniques we discuss apply to both. In an access-control application, the recovered biometric is verified by comparison with a stored hash of the original in a manner identical to password-based systems. In a key-management application, the (now recovered) original serves as a shared secret from which an encryption (decryption) key can be generated.

While secure biometric technology addresses one security threat facing biometric systems, it should be kept in mind that a variety of threats exist at various points in the biometric subsystem chain. For instance, individual modules can be forged or tampered with by attackers. Examples include a fake feature extraction module that produces pre-selected features that allow an intruder to gain access, or a fake decision-making entity that bypasses the authentication subsystem altogether. In remote authentication settings, where biometric measurements are collected at a remote site, not co-located with the stored enrollment data, other weak points exist. Dishonest entities such as servers that impersonate a user or perform data mining to gather information could be the source of successful attacks. Furthermore, in remote settings, the communication channel could also be compromised and biometric data could be intercepted and modified. Not all these threats are guarded against with secure biometric templates. Some can be dealt with using standard cryptographic techniques. But, in general, system designers need to be aware of all possible points of attack in a particular system.

In view of the above threats, a few desirable properties regarding biometric system security are listed as follows:

- Availability: Legitimate users should not be denied access
- Integrity: Forging fake identity should be infeasible
- Confidentiality: Original biometric data should be kept secret
- Privacy: Database cross-matching should reveal little information
- Revocability: Revocation should be easy

C. Chapter Organization

The rest of this chapter is organized as follows. In Section II, related work in this area is described to give readers a sense for alternative approaches to the secure biometrics problem. Section III formally quantifies the trade-off between security and robustness for the class of secure biometric systems that we consider, and introduces the syndrome-coding-based approach. In Section IV, we describe a prototype system developed for iris biometrics. In Sections V and VI, two different approaches for securing fingerprint data are described. The first is based on a statistical modeling of the fingerprint data. The second approach involves transforming the fingerprint data to a representation with statistical properties that are well-suited to off-the-shelf syndrome codes. A summary of this new application of distributed source coding is given in Section VII, including a discussion on future research opportunities and potential standardization.

II. Related Work

One class of methods for securing biometric systems is "transformation-based". Transformation-based approaches essentially extract features from an enrollment biometric using a complicated transformation. Authentication is performed by pattern matching in the transform domain. Security is assumed to come from the choice of a good transform which masks the original biometric data. In some cases the transform itself is assumed to be kept secret and design considerations must be made to ensure this secrecy. Particularly in the case when the transform itself is compromised, it is difficult to prove rigorously the security of such systems. Notable techniques in this category include cancelable biometrics [2], [3], score matching-based techniques [4], and threshold-based biohashing [5].

The main focus of this chapter is on an alternative class of methods that are based on using some form of "helper data." In such schemes, user-specific helper data is computed and stored from an enrollment biometric. The helper data itself and the method for generating this data can be known and is not required to be secret. To perform authentication of a probe biometric, the stored helper data is used to reconstruct the enrollment biometric from the probe biometric. However, the helper data by itself should not be sufficient to reconstruct the enrollment biometric. A cryptographic hash of the enrollment data is stored to verify bit-wise exact reconstruction.

Architectural principles underlying helper data–based approaches can be found in the information-theoretic problem of "common randomness" [6]. In this setting, different parties observe dependent random quantities (the enrollment and the probe) and then through finite-rate discussion (perhaps intercepted by an eavesdropper) attempt to agree on a shared secret (the enrollment biometric). In this context, error correction coding (ECC) has been proposed to deal with the joint problem of providing security against attackers, while accounting for the inevitable variability between enrollment and probe biometrics. On the one hand, the error correction capability of a error-correcting code can accommodate variations between multiple measurements of the same biometric. On the other hand, the check bits of the error correction code perform much the same function as a cryptographic hash of a password on conventional access control systems. Just as a hacker cannot invert the hash and steal the password, he cannot use the check bits to recover and steal the biometric.

An important advantage of helper data–based approaches relative to transformation–based approaches is that the security and robustness of helper data–based schemes are generally easier to quantify and prove. The security of transformation-based approaches are difficult to analyze since there is no straightforward way to quantify security when the transformation algorithm itself is compromised. In helper data–based schemes, this information is known to an attacker, and the security is based on the performance bounds of error correcting codes, which have been deeply studied.

To the best of our knowledge, Davida, Frankel, and Matt were the first to consider the use of ECC in designing a secure biometrics system for access control [7]. Their approach seems

to have been developed without knowledge of the work on common randomness in the information theory community. They describe a system for securely storing a biometric and focuses on three key aspects: security, privacy, and robustness. They achieve security by signing all stored data with a digital signature scheme and achieve privacy and robustness by using a systematic algebraic error-correcting code to store the data. A shortcoming of their scheme is that the codes employed are only decoded using bounded distance decoding. In addition, the security is hard to assess rigorously and there is no experimental validation using real biometric data.

The work by Juels and Wattenberg [8] extends the system of Davida, et al. [7] by introducing a different way of using error-correcting codes. Their approach is referred to as "fuzzy commitment". In the enrollment stage the initial biometric is measured and a random codeword of an error correcting code is chosen. The hash of this codeword along with the difference between an enrollment biometric and the codeword are stored. During authentication, a second measurement of the user's biometric is obtained, then the difference between this probe biometric and the stored difference is obtained, and error correction is then carried out to recover the codeword. Finally, if the hash of the resulting codeword matches the hash of the original codeword, then access is granted. Since the hash is difficult to invert, the codeword is not revealed. The value of the initial biometric is hidden by subtracting a random codeword from it, so the secure biometric hides both codeword and biometric data. This scheme relies heavily on the linearity/ordering of the encoded space to perform the difference operations. In reality, however, the feature space may not match such linear operations well.

A practical implementation of a fuzzy commitment scheme for iris data is presented in [9]. The authors utilize a concatenated-coding scheme in which Reed-Solomon codes are used to correct errors at the block level of an iris (e.g., burst errors due to eyelashes), while Hadamard codes are used to correct random errors at the binary level (e.g., background errors). They report a false reject rate of 0.47% at a key length of 140 bits on a small proprietary database including 70 eyes and 10 samples for each eye. As the authors note, however, the key length does not directly translate into security and they estimate a security of about 44 bits. It is also suggested in [9] that passwords could be added to the scheme to substantially increase security.

In [10] Juels and Sudan proposed the fuzzy vault scheme. This is a cryptographic construct that is designed to work with unordered sets of data. The fuzzy vault scheme essentially combines the polynomial reconstruction problem with ECC. Briefly, a set of t values from the enrollment biometric are extracted, and a length κ vector of secret data (i.e., the encryption key) is encoded using an (n, k) ECC. For each element of the enrollment biometric, measurement-codeword pairs would be stored as part of the vault. Additional random "chaff" points are also stored with the objective of obscuring the secret data. In order to unlock the vault, an attacker must be able to separate the chaff points from the legitimate points in the vault, which becomes increasingly difficult with a larger number of chaff points. To perform authentication, a set of values from a probe biometric could be used to initialize a codeword, which would then be subject to erasure and error decoding to attempt recovery of the secret data.

One of the main contributions of the fuzzy vault work was to realize that the set overlap noise model described in [10] can effectively be transformed into a standard errors and erasures noise model. This allowed application of Reed-Solomon codes, which are powerful codes and analytically tractable enough to obtain some privacy guarantees. The main shortcoming is that the set overlap noise model is not realistic for most biometrics since feature points typically vary slightly from one biometric measurement to the next rather than either matching perfectly or not matching at all.

Nonetheless, several fuzzy vault schemes applied to various biometrics have been proposed. Clancy, et al. [11] proposed to use the $X - Y$ location of minutiae points of a fingerprint to encode the secret polynomial, and describe a random point-packing technique to fill in the chaff points. The authors estimate 69 bits of security and demonstrate a false reject rate of 30%. Yang and Verbauwhede [12] also used the minutiae point location of fingerprints for their fuzzy vault scheme. However, they convert minutiae points to a polar coordinate system with respect to an origin that is determined based on a similarity metric of multiple fingerprints. This scheme was evaluated on a very small database of 10 fingers and a false reject rate of 17% was reported.

It should also be noted that there do exist variants of the fuzzy vault scheme that do not employ ECC. For instance, the work of Uludag, et al. [13] employs cyclic redundancy check (CRC) bits to identify the actual secret from several candidates. Nandakumar, et al. [14] further extended this scheme in a number of ways to increase the overall robustness of this approach. On the FVC2002-DB2 database [15], this scheme achieves 9% false reject rate (FRR) and 0.13% false accept rate (FAR). The authors also estimate 27-40 bits of security depending on the assumed distribution of minutiae points.

As evident from the literature, error-correcting codes indeed provide a powerful mechanism to cope with variations in biometric data. While the majority of schemes have been proposed in the context of fingerprint and iris data, there also exist schemes that target face, signature and voice data. Some schemes that make use of multi-biometrics are also beginning to emerge. Readers are referred to review articles on biometrics and security for further information on work in this area [16], [17].

In the sections that follow, the secure biometrics problem is formulated in the context of distributed source coding. We first give a more formal description of the problem set-up, and then describe solutions using techniques that draw from information theory, probabilistic inference, signal processing and pattern recognition. We quantify security and robustness and provide experimental results for a variety of different systems.

III. Overview of Secure Biometrics using Syndromes

A. Notation

We denote random variables using sans-serif and random vectors using bold sans-serif, x and $\mathbf{x}$, respectively. The corresponding sample values and vectors are denoted using serifs x and $\mathbf{x}$, respectively. The length of vectors will be apparent from context or, when needed, indicated explicitly as, e.g., x^n for the n-length random vector $\mathbf{x}$. The ith element of a random or sample vector is denoted as x_i or x_i, respectively. Sets are denoted using calligraphic font, e.g., the set of sample values of x is denoted $\mathcal{X}$, its n-fold product $\mathcal{X}^n$, and $|\cdot|$ applied to a set denotes its cardinality. We use $H(\cdot)$ to denote entropy; its argument can be either a random variable or its distribution; we use both interchangeably. For the special case of a Bernoulli-p source we use $H_B(p)$ to denote its entropy. Along the same lines, we use $I(\cdot;\cdot)$ and $I(\cdot;\cdot|\cdot)$ to denote mutual and conditional mutual information, respectively.

B. Enrollment and Authentication

As depicted in Fig. 1, the secure biometrics problem is realized in the context of a Slepian-Wolf coding framework. In the following, we describe the system operation in terms of an access-control application. During enrollment, a user is selected and their raw biometric $\mathbf{b}$ is determined by nature. The biometric is a random vector drawn according to some distribution $p_{\mathbf{b}}(\mathbf{b})$. A joint sensing, feature extraction, and quantization function $f_{\rm feat}(\cdot)$ maps the raw biometric into the length-n enrollment biometric $\mathbf{x} = f_{\rm feat}(\mathbf{b})$. Next, a function $f_{\rm sec}(\cdot)$ maps the enrollment biometric $\mathbf{x}$ into the secure biometric $\mathbf{s} = f_{\rm sec}(\mathbf{x})$ as well as into a cryptographic hash of the enrollment $\mathbf{h} = f_{\rm hash}(\mathbf{x})$. The structure of the encoding function $f_{\rm sec}(\cdot)$ reveals information about $\mathbf{x}$ without leaking too much secrecy. In contrast, the cryptographic hash function $f_{\rm hash}(\cdot)$ has no usable structure and is assumed to leak no information about $\mathbf{x}$. The access control point stores $\mathbf{s}$ and $\mathbf{h}$, as well as the functions $f_{\rm sec}(\cdot)$ and $f_{\rm hash}(\cdot)$. The access control point does not store $\mathbf{b}$ or $\mathbf{x}$.

In the authentication phase, a user requests access and provides a second reading of their biometric $\mathbf{b}'$. We model the biometrics of different users as statistically independent. Therefore, if the user is not the legitimate user $p_{\mathbf{b}',\mathbf{b}}(\mathbf{b}',\mathbf{b}) = p_{\mathbf{b}}(\mathbf{b}')p_{\mathbf{b}}(\mathbf{b})$. On the other hand, if $\mathbf{b}'$ comes from the legitimate user $p_{\mathbf{b}',\mathbf{b}}(\mathbf{b}',\mathbf{b}) = p_{\mathbf{b}'|\mathbf{b}}(\mathbf{b}'|\mathbf{b})p_{\mathbf{b}}(\mathbf{b})$, where $p_{\mathbf{b}'|\mathbf{b}}(\cdot|\cdot)$ models the measurement noise between biometric readings. The features extracted from this second reading are $\mathbf{y} = f_{\rm feat}(\mathbf{b}')$. Instead of working with $p_{\mathbf{b}',\mathbf{b}}(\mathbf{b}',\mathbf{b})$, we choose to work with $p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})$. The feature extraction function $f_{\rm feat}(\cdot)$ induces the distribution $p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})$ from $p_{\mathbf{b}',\mathbf{b}}(\mathbf{b}',\mathbf{b})$. Per the preceding discussion, if the user is legitimate $p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y}) = p_{\mathbf{x}}(\mathbf{x})p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x})$, and if the user is illegitimate, then $p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y}) = p_{\mathbf{x}}(\mathbf{x})p_{\mathbf{x}}(\mathbf{y})$.[1]

[1]We comment that Fig. 1 can be thought of as somewhat specific to a single observation. If one had multiple observations of the underlying biometric, one could symmetrize the joint distribution by assuming that each observation of the underlying biometric (including the enrollment) was through a noisy channel. The current setting simplifies the model and is sufficient for our purposes.

The decoder $g_{\rm dec}(\cdot,\cdot)$ combines the secure biometric $\mathbf{s}$ with the probe $\mathbf{y}$ and either produces an estimate of the enrollment $\hat{\mathbf{x}} = g_{\rm dec}(\mathbf{s},\mathbf{y})$ or a special symbol $\varnothing$ indicating decoding failure. Finally, the stored $\mathbf{h}$ is compared to $f_{\rm hash}(\hat{\mathbf{x}})$. If they match, access is granted. If they do not, access is denied.[2]

C. Performance Measures: Security and Robustness

The probability of authentication error (false rejection) is

$$P_{\rm FR} = \Pr\left[\mathbf{x} \neq g_{\rm dec}(\mathbf{y}, f_{\rm sec}(\mathbf{x}))\right],$$

where $P_{\mathbf{y},\mathbf{x}}(\mathbf{y},\mathbf{x}) = P_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x})P_{\mathbf{x}}(\mathbf{x})$. As discussed later, we will find it natural to use a logarithmic performance measure to quantify authentication failure. We use the error exponent

$$E_{\rm FR} = -\frac{1}{n}\log P_{\rm FR} \tag{1}$$

as this measure.

It must be assumed that an attacker makes many attempts to guess the desired secret. Therefore, measuring the probability that a single attack succeeds is not particularly meaningful. Instead, security should be assessed by measuring how many attempts an attack algorithm must make to have a reasonable probability of success. We formalize this notion by defining an attack as the creation of a list of candidate biometrics. If the true biometric is on the list, the attack is successful. The list size required to produce a successful attack with high probability translates into our measure of security.

Let $\mathcal{L} = \mathcal{A}_{R_{\rm sec}}(\cdot)$ be a list of $2^{nR_{\rm sec}}$ guesses for $\mathbf{x}$ produced by the attack algorithm $\mathcal{A}()$ that is parametrized by the rate $R_{\rm sec}$ of the attack and takes as inputs $p_{\mathbf{x}}(\cdot)$ $p_{\mathbf{y}|\mathbf{x}}(\cdot|\cdot)$, $f_{\rm sec}(\cdot)$, $f_{\rm hash}(\cdot)$, $g_{\rm dec}(\cdot,\cdot)$, $\mathbf{s}$, and $\mathbf{h}$. The attack algorithm does not have access to a probe generated from the enrollment $\mathbf{x}$ according to $p_{\mathbf{y}|\mathbf{x}}(\cdot|\cdot)$ because it does not have a measurement of the original biometric. From the quantities it does know, a good attack is to generate a list $\mathcal{L}$ of candidate biometrics that match the secure biometric $\mathbf{s}$ (candidate biometrics that do not match $\mathbf{s}$ can be eliminated out of hand). That is, for each candidate $\mathbf{x}_{\rm cand} \in \mathcal{L}$, $f_{\rm sec}(\mathbf{x}_{\rm cand}) = \mathbf{s}$. While the cryptographic hash $f_{\rm hash}(\cdot)$ is assumed to be non-invertible, we conservatively assume that the secure biometric encoding $f_{\rm sec}(\cdot)$ is known to the attacker, and furthermore assume that the attacker can invert the encoding, and hence the list $\mathcal{L}$ can be generated.

Once the list $\mathcal{L}$ is created, a natural attack is to test each $\mathbf{x}_{\rm cand} \in \mathcal{L}$ in turn to check whether $f_{\rm hash}(\mathbf{x}_{\rm cand}) = \mathbf{h}$. If the hashes match, the attack has succeeded. The system is secure against attacks if and only if the list of all possible candidate biometrics matching the secure biometric is so enormous that the attacker will only have computational resources to compute the hashes of a negligible fraction of candidate biometrics. Security thus results from dimensionality reduction: a high-dimensional $\mathbf{x}$ is mapped to a low-dimensional $\mathbf{s}$ by $f_{\rm sec}(\cdot)$. The size of the total number of candidate biometrics that map onto the secure biometric $\mathbf{s}$ is exponential in the difference in dimensionality.

[2]In a data encryption application an encryption key is generated from $\mathbf{x}$ and the matching decryption key from $\hat{\mathbf{x}}$. A cryptographic hash function $f_{\rm hash}(\cdot)$ is not required – if the reconstruction is not exact, then the generated key will not match the one used to encrypt and decryption will fail.

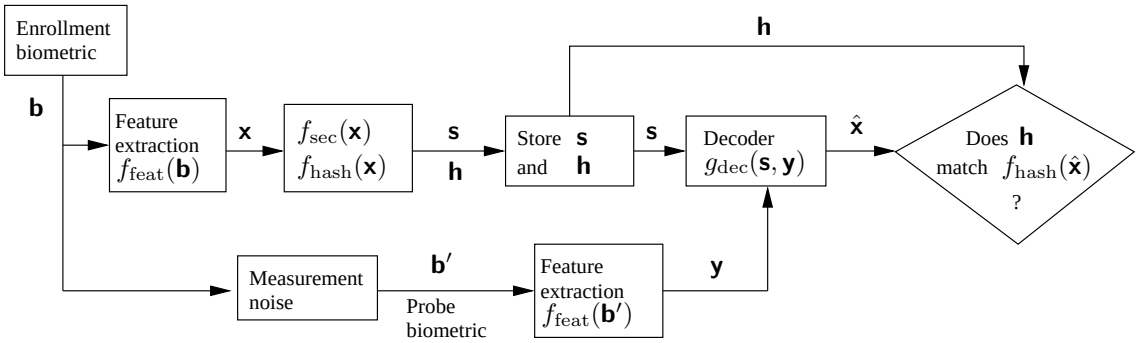


Fig. 1. Block diagram of Slepian-Wolf system for secure biometrics.

The probability that a rate-R_{sec} attack is successful equals the probability that the enrollment biometric is on the attacker's list, $P_{\text{SA}}(R_{\text{sec}}) =$

$$\Pr\left[\mathbf{x}\in\mathcal{A}_{R_{\text{sec}}}\left(p_{\mathbf{x}}(\cdot),p_{\mathbf{y}|\mathbf{x}}(\cdot|\cdot),f_{\text{sec}}(\cdot),f_{\text{hash}}(\cdot),g_{\text{dec}}(\cdot,\cdot),\mathbf{h},\mathbf{s}\right)\right].$$

The system is said to be "ϵ-secure" to rate-R_{sec} attacks if $P_{\text{SA}}(R_{\text{sec}}) < \epsilon$.

Equivalently, we refer to a scheme with $P_{\text{SA}}(R_{\text{sec}}) = \epsilon$ as having $n \cdot R_{\text{sec}}$ bits of security with confidence $1-\epsilon$. With probability $1-\epsilon$ an attacker must search a key space of $n \cdot R_{\text{sec}}$ bits to crack the system security. In other words the attacker must make $2^{nR_{\text{sec}}}$ guesses. The parameter R_{sec} is a logarithmic measure of security, quantifying the rate of the increase in security as a function of block length n. For instance, 128-bit security requires $nR_{\text{sec}} = 128$. It is because we quantify security with a logarithmic measure that we also use the logarithmic measure of error-exponents to quantify robustness in (1).

Our objective is to construct an encoder and decoder pair that obtains the best combination of robustness (as measured by P_{FR}) and security (as measured by $P_{\text{SA}}(R_{\text{sec}})$) as a function of R_{sec}. In general, improvement in one necessitates a decrease in the other. For example, if $P_{\text{SA}}(0.5) = \epsilon$ and $P_{\text{FR}} = 2^{-10}$ at one operating point, increasing the security to $0.75n$ might yield another operating point at $P_{\text{SA}}(0.75) = \epsilon$ and $P_{\text{FR}} = 2^{-8}$. With this sense of the fundamental trade offs involved, we now define the security-robustness region.

Definition 1: For any $\epsilon > 0$ and any $p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})$ the security-robustness region $\mathcal{R}_\epsilon$ is defined as the set of pairs (r,γ) for which an encoder-decoder pair $(f_{\text{sec}}(\cdot), g_{\text{dec}}(\cdot,\cdot))$ exists that achieves rate-r security with an authentication failure exponent of γ:

$$\mathcal{R}_\epsilon = \left\{(r,\gamma)\,\middle|\, P_{\text{SA}}(r) \le \epsilon, \gamma \ge -\frac{1}{n}\log P_{\text{FR}}\right\}.$$

D. Quantifying security

In this section, we quantify an achievable subset of the security-robustness region $\mathcal{R}_\epsilon$. This specifies the trade off between P_{FR} and $P_{\text{SA}}(\cdot)$ in an idealized setting. Our derivation assumes that $\mathbf{x}$ and $\mathbf{y}$ are jointly ergodic and take values in finite sets, $\mathbf{x}\in\mathcal{X}^n, \mathbf{y}\in\mathcal{Y}^n$. One can derive an outer bound to the security-robustness region by using upper bounds on the failure exponent (via the sphere-packing bound for Slepian-Wolf coding). Since our prime purpose in this section is to provide a solid framework for our approach, we don't further develop outer bounds here.

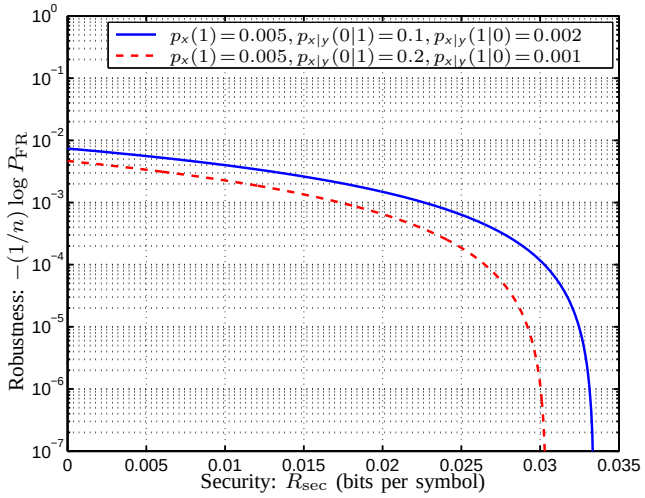


Fig. 2. Example security-robustness regions. The horizontal axis represents the maximum security rate R_{sec} such that $P_{\text{SA}}(R_{\text{sec}}) < \epsilon$, while the vertical axis represents robustness. The security-robustness region of the system corresponding to the solid curve (all points below the curve) dominates that of the dashed curve.

We use a rate-R_{SW} random "binning" function (a Slepian-Wolf code [1]) to encode $\mathbf{x}$ into the secured biometric $\mathbf{s}$. Specifically, we independently assign each possible sequence $\mathbf{x}\in\mathcal{X}^n$ an integer selected uniformly from $\{1,2,\ldots,2^{nR_{\text{SW}}}\}$. The secure biometric is this index $s = f_{\text{sec}}(\mathbf{x})$. Each possible index $s \in \{1,2,\ldots,2^{nR_{\text{SW}}}\}$ indexes a set or "bin" of enrollment biometrics, $\{\tilde{\mathbf{x}}\in\mathcal{X}^n | f_{\text{sec}}(\tilde{\mathbf{x}}) = s\}$. The secure biometric can be thought of either as a scalar index s, or as its binary expansion, a uniformly distributed bit sequence $\mathbf{s}$ of length nR_{SW}.

During authentication, a user provides a probe biometric $\mathbf{y}$ and claims to be a particular user. The decoder $g_{\text{dec}}(\mathbf{y},\mathbf{s})$ searches for the most likely vector $\hat{\mathbf{x}}\in\mathcal{X}^n$ given $\mathbf{y}$ according to the joint distribution $p_{\mathbf{x},\mathbf{y}}$ such that $\hat{\mathbf{x}}$ is in bin $\mathbf{s}$, i.e., $f_{\text{sec}}(\hat{\mathbf{x}}) = \mathbf{s}$. If a unique $\hat{\mathbf{x}}$ is found, then the decoder outputs this result. Otherwise, an authentication failure is declared and the decoder returns $\varnothing$.

According to the Slepian-Wolf Theorem [1], [18], the

decoder will succeed with probability approaching 1 as n increases provided that $R_{\rm SW} > (1/n)H(\mathbf{x}|\mathbf{y})$. Thus, $P_{\rm FR}$ approaches zero for long block lengths. The theory of error exponents for Slepian-Wolf coding [19] tells us that $-(1/n)\log P_{\rm FR} \geq E_{\rm SW}(R_{\rm SW})$, where $E_{\rm SW}(R_{\rm SW}) =$

$$\max_{0\leq\rho\leq 1}\left\{\rho R_{\rm SW} - \frac{1}{n}\log\sum_{\mathbf{y}} p_{\mathbf{y}}(\mathbf{y})\left[\sum_{\mathbf{x}} p_{\mathbf{x}|\mathbf{y}}(\mathbf{x}|\mathbf{y})^{\frac{1}{1+\rho}}\right]^{1+\rho}\right\}. \tag{2}$$

If $R_{\rm SW} < (1/n)H(\mathbf{x}|\mathbf{y})$ then $E_{\rm SW}(R_{\rm SW}) = 0$. For $R_{\rm SW} > (1/n)H(\mathbf{x}|\mathbf{y})$ the error exponent $E_{\rm SW}(R_{\rm SW})$ increases monotonically in $R_{\rm SW}$. Note that (2) holds for any joint distribution, not just independent identically distributed (i.i.d.) ones. However, if the source and channel are memoryless, the joint distribution is i.i.d., and $p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y}) = \prod_{i=1}^{n} p_{x,y}(x_i,y_i)$. As a result, the second term of (2) simplifies considerably to $-\log\sum_y p_y(y)\Big[\sum_x p_{x|y}(x|y)^{\frac{1}{1+\rho}}\Big]^{1+\rho}$.

Next, we consider the probability of successful attack, i.e., how well an attacker can estimate $\mathbf{x}$ given the secure biometric $\mathbf{s}$. According to the asymptotic equipartition property [20], under the fairly mild technical condition of ergodicity, it can be shown that conditioned on $\mathbf{s} = f_{\rm sec}(\mathbf{x})$, $\mathbf{x}$ is approximately uniformly distributed over the typical set of size $2^{H(\mathbf{x}|\mathbf{s})}$. Therefore, with high probability, it will take approximately this many guesses to identify $\mathbf{x}$. We compute $H(\mathbf{x}|\mathbf{s})$ as

$$H(\mathbf{x}|\mathbf{s}) = H(\mathbf{x},\mathbf{s}) - H(\mathbf{s}) \overset{(a)}{=} H(\mathbf{x}) - H(\mathbf{s}) \overset{(b)}{=} H(\mathbf{x}) - nR_{\rm SW}, \tag{3}$$

where (a) follows because $\mathbf{s} = f_{\rm sec}(\mathbf{x})$, i.e., $\mathbf{s}$ is a deterministic function of $\mathbf{x}$, and (b) follows from the method of generating the secure biometric, i.e., $\mathbf{s}$ is uniformly distributed over length-$nR_{\rm SW}$ binary sequences (in other words $\mathbf{s}$ is a length-$nR_{\rm SW}$ i.i.d. Bernoulli(0.5) sequence).

Using (2) and (3) we bound the security-robustness region in the following:

Theorem 1: For any $\epsilon > 0$ as $n \to \infty$, an inner bound to the security-robustness region $\mathcal{R}_\epsilon$ defined in Definition 1 is found by taking a union over all possible feature extraction functions $f_{\rm feat}(\cdot)$ and secure biometric encoding rates $R_{\rm SW}$

$$\mathcal{R}_\epsilon \supset \bigcup_{f_{\rm feat}(\cdot), R_{\rm SW}} \left\{ r,\gamma \Big| r < \frac{1}{n}H(\mathbf{x}) - R_{\rm SW}, \gamma < E_{\rm SW}(R_{\rm SW})\right\}$$

where $E_{\rm SW}(R_{\rm SW})$ is given by (2) for the $p_{\mathbf{x},\mathbf{y}}(\cdot,\cdot)$ induced by the chosen $f_{\rm feat}(\cdot)$.

Proof: The theorem is proved by the random-binning encoding and maximum-likelihood decoding construction specified above. The same approach holds for any jointly ergodic sources. The uniform distribution of the true biometric across the conditionally typical set of size $2^{H(\mathbf{x}|\mathbf{s})}$ provides security, cf. (3). As long as the rate of the attack $r < \frac{1}{n}H(\mathbf{x}) - R_{\rm SW}$, then $P_{\rm SA}(r) < \epsilon$ for any $\epsilon > 0$ as long as n is sufficiently large. Robustness is quantified by the error-exponent of Slepian-Wolf decoding given by (2). ∎

Fig. 2 plots an example of the security-robustness region for a memoryless insertion and deletion channel that shares some commonalities with the fingerprint channel that we discuss in Section V. The enrollment biometric $\mathbf{x}$ is an i.i.d. Bernoulli sequence with $p_x(1) = 0.05$. The true biometric is observed through the asymmetric binary channel with deletion probability $p_{y|x}(0|1)$ and insertion probability $p_{y|x}(1|0)$. We plot the resulting security-robustness regions for two choices of insertion and deletion probabilities.

We now contrast $P_{\rm SA}(\cdot)$, the measure of security considered in Theorem 1 and defined in Definition 1, with the probability of breaking into the system using the classic attack used to calculate the FAR. In the FAR attack, $\mathbf{y}$ is chosen independently of $\mathbf{x}$, i.e., $p_{\mathbf{y},\mathbf{x}}(\mathbf{y},\mathbf{x}) = p_{\mathbf{y}}(\mathbf{y})p_{\mathbf{x}}(\mathbf{x})$. This attack fails unless the $\mathbf{y}$ chosen is jointly typical with $\mathbf{x}$, i.e., unless the pair $\mathbf{y}$ and (the unobserved) $\mathbf{x}$ look likely according to $p_{\mathbf{y},\mathbf{x}}(\cdot,\cdot)$. Given that a $\mathbf{y}$ is selected that is jointly typical with the enrollment $\mathbf{x}$, the decoder will then successfully decode to $\mathbf{x}$ with high probability, the hash will match, and access will be granted. To find such a $\mathbf{y}$ when picking according to the marginal $p_{\mathbf{y}}(\mathbf{y})$ takes approximately $2^{I(\mathbf{y};\mathbf{x})} = 2^{H(\mathbf{x})-H(\mathbf{x}|\mathbf{y})}$ guesses. We must set $R_{\rm SW} > (1/n)H(\mathbf{x}|\mathbf{y})$, else as discussed above, (2) tells us that $P_{\rm FR}$ goes to one. This constraint means that (cf. eqn.(3)) $H(\mathbf{x}|\mathbf{s}) < H(\mathbf{x}) - H(\mathbf{x}|\mathbf{y})$. Thus, while a FAR-type attack required $2^{H(\mathbf{x})-H(\mathbf{x}|\mathbf{y})}$ guesses, the smarter attack considered in the theorem required $2^{H(\mathbf{x})-nR_{\rm SW}}$ and thus an FAR-type attack will almost always take many more guesses than an attack that makes its guesses conditioned on $\mathbf{s}$.

We again emphasize that an attack that identifies a biometric $\tilde{\mathbf{x}}$ such that $f_{\rm sec}(\tilde{\mathbf{x}}) = \mathbf{s}$ is not necessarily a successful attack. Indeed, our security analysis assumes that an attacker can easily find $\tilde{\mathbf{x}}$ that satisfies $f_{\rm sec}(\tilde{\mathbf{x}}) = \mathbf{s}$. However, if $\tilde{\mathbf{x}} \neq \mathbf{x}$, then $f_{\rm hash}(\tilde{\mathbf{x}}) \neq f_{\rm hash}(\mathbf{x}) = \mathbf{h}$ and access will not be granted. Thus, in the bounds on security provided by Theorem 1, it is assumed that the attacker is limited to guesses of $\tilde{\mathbf{x}}$ that satisfy $f_{\rm sec}(\tilde{\mathbf{x}}) = \mathbf{s}$.

E. Implementation using syndrome coding

In our work, the enrollment biometric $\mathbf{x}$ is binary and we use a linear code for the encoding function,

$$\mathbf{s} = f_{\rm sec}(\mathbf{x}) = \mathbf{H}\mathbf{x}, \tag{4}$$

where $\mathbf{H}$ is a $k \times n$ binary matrix and addition is mod-2, i.e., $a \oplus b = \text{XOR}(a,b)$. Using the language of algebra, the secure biometric $\mathbf{s}$ is the "syndrome" of the set of sequences $\tilde{\mathbf{x}} \in \{0,1\}^n$ satisfying $\mathbf{H}\tilde{\mathbf{x}} = \mathbf{s}$. This set is also referred to as the "coset" or "equivalence class" of sequences. Note that all cosets are of equal cardinality[3].

An attacker should limit his set of guesses $\mathcal{A}_{R_{\rm sec}}$ to be a subset of the coset corresponding to the stored $\mathbf{s}$. If all $\mathbf{x}$ sequences were equally likely (which is the case since cosets are of equal size and if $\mathbf{x}$ is an i.i.d.Bernoulli(0.5) sequence), then the attacker would need to check through nearly the entire list to find the true biometric with high probability. For this case and from (3), we calculate the logarithm of the list size to be $H(\mathbf{x}) - H(\mathbf{s}) = n - k$, where n and k are the dimensions of

[3]It can be shown that any $\tilde{\mathbf{x}}$ in the $\mathbf{s}'$ coset can be written as $\tilde{\mathbf{x}} = \mathbf{x} \oplus \mathbf{z}$ for some $\mathbf{x}$ in the $\mathbf{s}$ coset and where $\mathbf{z}$ is fixed. Thus, $\mathbf{H}\tilde{\mathbf{x}} = \mathbf{H}(\mathbf{x} \oplus \mathbf{z}) = \mathbf{s} + \mathbf{H}\mathbf{z} = \mathbf{s}'$. The $\mathbf{s}'$ coset corresponds to all elements of the $\mathbf{s}$ coset (defined by its syndrome $\mathbf{s}$) shifted by $\mathbf{z}$, and thus the cardinalities of the two cosets are equal.

the $\mathbf{x}$ and $\mathbf{s}$ vectors, respectively, and are also the dimensions of the $\mathbf{H}$ matrix in (4). This follows from the model: $H(\mathbf{x}) = n$ since $\mathbf{x}$ is i.i.d. Bernoulli(0.5) and $H(\mathbf{s}) = k$ since cosets are of equal size and $p_{\mathbf{x}}(\mathbf{x}) = 2^{-n}$ for all $\mathbf{x}$.

If the enrollment biometric $\mathbf{x}$ is not a uniformly-distributed i.i.d. sequence – which is going to be the case generally – the attacker need not check through the entire coset corresponding to $\mathbf{s}$. Instead the attacker should intersect the coset with the set of sequences in $\mathcal{X}^n$ that look like biometrics. These are the "typical" sequences [20] determined by the probability measure $p_{\mathbf{x}}(\cdot)$. This intersection is taken into account in (3).[4] If the rows of the $\mathbf{H}$ matrix in (4) are generated in an independent and identically distributed manner, then step (b) in (3) simplifies as follows:

$$H(\mathbf{x}|\mathbf{s}) = H(\mathbf{x}) - H(\mathbf{s}) = H(\mathbf{x}) - \sum_{i=1}^{k} H(\mathsf{s}_i) = H(\mathbf{x}) - kH(\mathsf{s}). \tag{5}$$

In an actual implementation, we generally do not generate the rows of $\mathbf{H}$ in an i.i.d. manner, but rather use a structured code such as a low-density parity-check (LDPC) code. In such situations, (3) is a *lower* bound on the security of the system since $H(\mathbf{s}) \leq \sum_{i=1}^{k} H(\mathsf{s}_i)$ using the chain rule for entropy and the fact that conditioning reduces entropy, and the third equality still holds as long as the rows of $\mathbf{H}$ are identically distributed (even if not independent). Furthermore, contrast (5) with (3). In the latter, $H(\mathbf{s}) = nR_{\mathrm{SW}}$ because of the random binning procedure. The assumptions of this procedure no longer hold when using linear codes to implement binning.

It is informative to consider estimating (5). The second term, $kH(\mathsf{s})$ is easy to estimate since it involves only the entropy of a marginal distribution. An estimation procedure would be to encode many biometrics using different codes, construct a marginal distribution for s, and calculate the entropy of the marginal. Particularly, if the code alphabet is small (say binary) little data is required for a good estimate. The first term $H(\mathbf{x})$ is harder to estimate. Generally, we would need to collect a very large number of biometrics (if n is large) to have sufficient data to make a reliable estimate of the entropy of the n-dimensional joint distribution. Thus, the absolute level of security is difficult to evaluate. However, the analysis provides a firm basis on which to evaluate the comparative security between two systems. The $H(\mathbf{x})$ term is common to both and cancels out in a calculation of relative security – the difference between the individual securities, which is $kH(\mathsf{s}) - k'H(\mathsf{s}')$.

IV. Iris System

This second describes a prototype implementation of a secure biometrics system for iris recognition based on syndrome coding techniques. Experimental results on the CASIA (Chinese Academy of Sciences Institute of Automation) database [21] are presented.

[4]We note that calculating the intersection may be difficult computationally. However, the security level quantified by Theorem 1 is conservative in the sense that it assumes that the attacker can calculate the intersection and produce the resulting list effortlessly.

A. Enrollment and Authentication

At enrollment the system performs the following steps. Starting with an image of a user's eye, the location of the iris is first detected, and the torus is then unwrapped into a rectangular region. Next, a bank of Gabor filters are applied to extract a bit sequence. The Matlab implementation from [22] could be used to perform these steps. Finally, the extracted feature vector $\mathbf{x}$ is produced by discarding bits at certain fixed positions that were determined to be unreliable[5]. The resulting $\mathbf{x} = f_{\mathrm{feat}}(\mathbf{b})$ consists of the most reliable bits; in our implementation 1806 bits are extracted. Finally, the bit string $\mathbf{x}$ is mapped into the secure biometric $\mathbf{s}$ by computing the syndrome of $\mathbf{x}$ with respect to a LDPC code. Specifically, a random parity check matrix $\mathbf{H}$ is selected from a good low rate degree distribution obtained via density evolution [23] and $\mathbf{s} = \mathbf{H} \cdot \mathbf{x}$ is computed.

To perform authentication, the decoder $g_{\mathrm{dec}}(\cdot,\cdot)$ repeats the detection, unwrapping, filtering, and least-reliable bit dropping processes. The resulting observation $\mathbf{y}$ is used as the input to a belief propagation decoder that attempts to find a sequence $\hat{\mathbf{s}}$ satisfying $\mathbf{H}\cdot\hat{\mathbf{s}} = \mathbf{s}$. If the belief propagation decoder succeeds, then the output $\hat{\mathbf{s}} = g_{\mathrm{dec}}(\mathbf{s}, \mathbf{y})$. Otherwise, an authentication failure (or false rejection) is declared and the output of $g_{\mathrm{dec}}(\mathbf{s}, \mathbf{y})$ is $\varnothing$.

Sample iris measurements from two different users are shown in Fig. 3. The bit correlation between different samples of the same user and differences between samples of different users are easily seen. It has also been observed that the bit sequences extracted from the irises contain 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, one would expect $p_{i,j} = 1/4$ for all $(i,j) \in \{0,1\}^2$. Instead, the following probabilities have been measured from the complete data set:

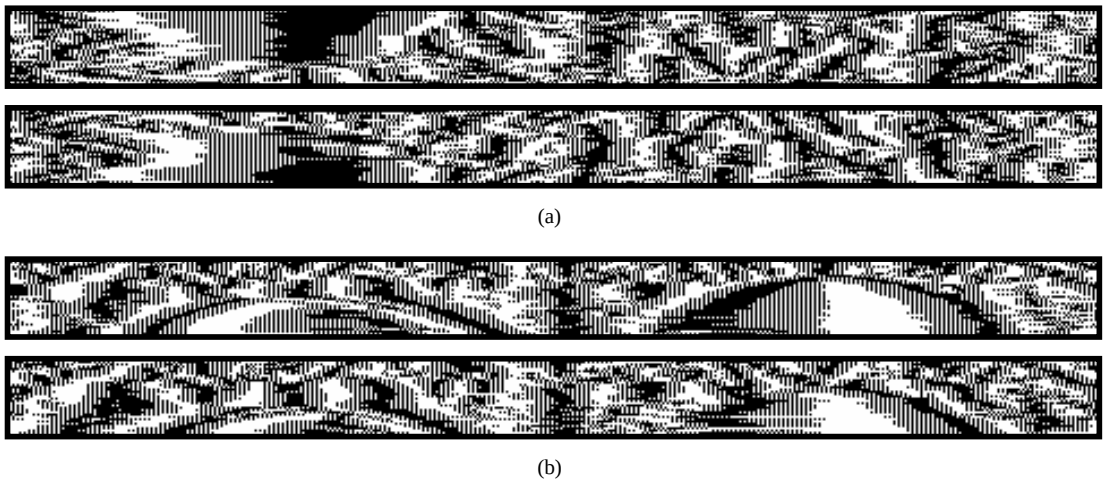
$$p_{0,0} = 0.319,\ p_{0,1} = 0.166,\ p_{1,0} = 0.166,\ p_{1,1} = 0.349.$$

Ignoring the inter-bit memory would result in degraded performance. Therefore, the belief propagation decoder is designed to exploit this source memory. Further details can be found in [24].

B. Experimental Results

The system is evaluated using the CASIA iris database [21]. The iris segmentation algorithm that was implemented was only able to correctly detect the iris in 624 out of 756 images [22, Chapter 2.4]. Since our emphasis is on the secure biometrics problem and not on iris segmentation, experiments were performed with the 624 iris that were segmented successfully. Furthermore, half of the iris images were used for training.

[5]Unreliable positions are those positions at which the bit values (0 or 1) are more likely to flip due to the noise contributed by eyelids and eyelashes, and due to a slight misalignment in the radial orientation of the photographed images. The bit positions corresponding to the outer periphery of the iris tend to be less reliable than those in the interior. These bit positions can be determined from the training data.



(a)

(b)

Fig. 3. Sample bit sequences extracted from iris data (a) Two sample measurements from one user (b) Two sample measurements from a second user.

Fig. 4 reports performance results for the 312 image test set from the CASIA iris database. The horizontal axis represents security while the vertical axis represents the probability of false rejection for a legitimate user. Better systems correspond to points in the lower right, but as Theorem 1 shows theoretically and the figure demonstrates, there is a trade-off between security and robustness. Specifically, if a rate R LDPC code is used, then $\mathbf{s}$ contains $n(1-R)$ bits. Under the idealized model where the iris data consists of i.i.d. Bernoulli(0.5) bits, our approach yields approximately $1806 \cdot R$ bits of security with confidence approaching 1. Increasing 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, there is a probability of false rejection equal to 0.0012 (*i.e.*, the leftmost point in the graph). Thus, it is shown that, relative to an insecure scheme, with essentially no change in the probability of authentication failure the syndrome-based scheme achieves almost 50 bits of security.

Higher levels of security can be achieved if larger authentication error rates are allowed. As discussed in Section III, the true level of security is more difficult to evaluate. Specifically, the original length of the bit sequence extracted from an iris in the 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. 4. If the original biometric is an i.i.d. sequence of Bernoulli(0.5) random bits, then the probability of guessing the true biometric from the syndrome would be about 2^{-t} (i.e., security of t bits). However, as discussed earlier in this section, 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 the syndrome vector 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 P_{FR}, 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.

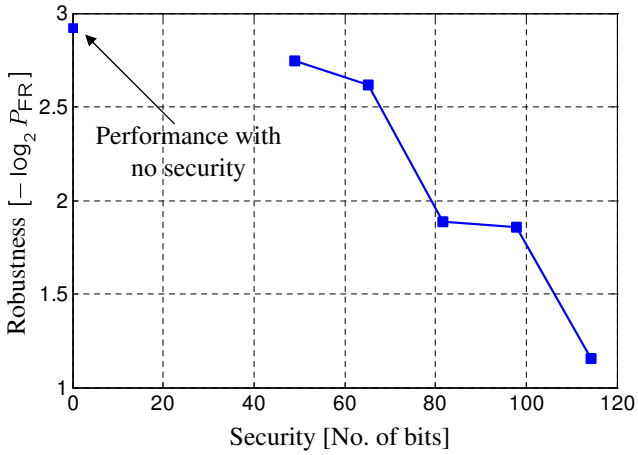


Fig. 4. Performance result of 312 iris images from CASIA database. Horizontal axis represents security, while vertical axis plots robustness in terms of the probability of false rejection. The original length of the bit sequence extracted from an iris is $n = 1806$, while the length of the syndrome is $1806-t$ bits, where t is plotted along the horizontal axis above. In fact, the actual number of bits of security is slightly smaller than t, since the syndrome bits are not Bernoulli(0.5). A detailed explanation appears at the end of this section.

That said, we are not aware of any computationally feasible methods of improving upon random guessing and believe that the estimated security provided here is still reasonable.

V. Fingerprint System: Modeling Approach

In the previous section we remarked on the difficulties caused by the correlations between bits in an iris biometric. These problems were dealt with by explicitly including the correlations in a belief propagation decoder. For fingerprint data, such problems are more severe. Models for fingerprint biometrics do not obviously map onto blocks of i.i.d. bits as would be ideal for a Slepian-Wolf LDPC code. We present two solutions to this problem. In this section, a "modeling" solution is discussed, in which the relationship between the enrollment biometric and the probe biometric is modeled as a noisy channel. The rest of this section describes a somewhat

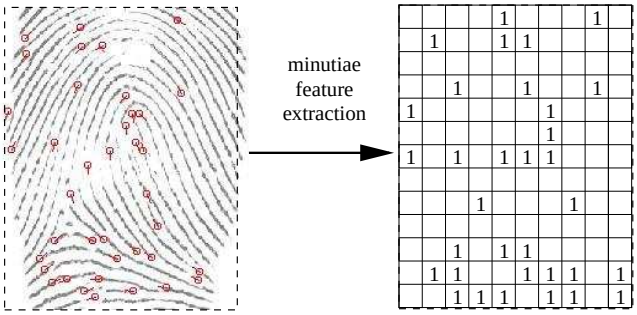


Fig. 5. Fingerprint and extracted feature vector.

complex statistical factor graph model for fingerprint data and corresponding graph-based inference decoding techniques.

In section VI, a second "transformation" approach is introduced, in which the fingerprint biometric is transformed, as well as possible, into a block of i.i.d. bits, and then a standard LDPC code and decoder are used. Although these two approaches are described in detail for fingerprint biometrics, other biometrics will have a similar dichotomy of possible approaches. For fingerprints, we have found that the transformation approach gives better results and makes it easier to quantify the security of the system, but both approaches are worth understanding.

A. Minutiae Representation of Fingerprints

A popular method for working with fingerprint data is to extract a set of "minutiae points" and to perform all subsequent operations on them [25]. Minutiae points have been observed to be stable over many years. Each minutiae is a discontinuity in the ridge map of a fingerprint, characterized by a triplet (x, y, θ) representing its spatial location in two dimensions and the angular orientation. In the minutiae map $\mathbf{M}$ of a fingerprint, $\mathbf{M}(x, y) = \theta$ if there is a minutia point at (x, y) and $\mathbf{M}(x, y) = \emptyset$ (empty set) otherwise. A minutiae map may be considered as a joint quantization and feature extraction function which operates on the fingerprint image, i.e., the output of the $f_{\text{feat}}(\cdot)$ box in Fig. 1. In Fig. 5, the minutiae map is visualized using a matrix as depicted in the right-hand plot, where a '1' simply indicates the presence of a minutiae at each quantized coordinate. In this figure, as well as in the model described throughout the rest of this section, the θ coordinate of the minutiae is ignored.

It is noted that different fingerprints usually have different numbers of minutiae. Furthermore, the number and location of minutiae could vary depending on the particular extraction algorithm that is used. For some applications, it could be important to account for such factors in addition to typical differences between fingerprint measurements, which will be discussed further in the next subsection. In the work described here, the enrollment feature vector $\mathbf{x}$ is modeled as a Bernoulli i.i.d.random vector.

B. Modeling the movement of fingerprint minutiae

In the following, a model for the statistical relationship $p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x})$ between the enrollment biometric and the probe biometric is described. There are three main effects that are captured by this model: (1) movement of enrollment minutiae when observed the second time in the probe, (2) deletions, i.e., minutiae observed at enrollment, but not during probe, and (3) insertions, i.e.,"spurious" minutiae observed in probe, but not during enrollment.

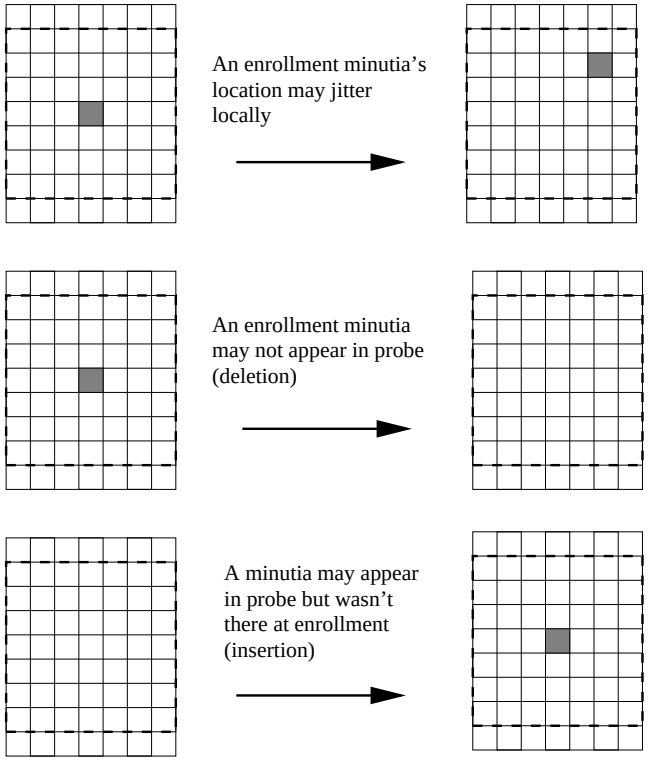


Fig. 6. Statistical model of fingerprints corresponding to local movement, deletion and insertion.

Fig. 6 depicts these three mechanisms in turn. First, minutiae observed at enrollment are allowed to jitter slightly around their locations in the enrollment vector when registered the second time in the probe. This movement is modeled within a local neighborhood, where up to three pixels in either the horizontal or vertical direction (or both) could be accounted for. The size of the local neighborhood depends on the resolution of the minutiae map and how coarsely it is quantized. Second, a minutia point may be registered in the enrollment reading, but not in the probe. Or, a minutia point may be displaced beyond the local neighborhood defined by the movement model. Both count as "deletions". Finally, minutia points that are not observed at enrollment, but may be in the probe vector are termed insertions.

The statistical model is formalized using a factor graph [26] as shown in Fig. 7. The presence of a minutiae point at position t, $t \in \{1, 2, \ldots, n\}$ in the enrollment grid is represented by the binary random variable x_t that takes on the value $x_t = 1$ only if a minutiae is present during enrollment.[6] For simplicity, the figure shows one-dimensional movement model. All

[6]Note that t indexes a position in the two-dimensional field of possible minutiae locations. The particular indexing used (e.g., raster-scan) is immaterial. The product of the number of rows and the number of columns equals n.

experimental results use a two-dimensional movement model.

The decoder observes two vectors: the probe biometric y_i for $i \in \{1, 2, \ldots, n\}$ and s_j for $j \in \{1, 2, \ldots, k\}$. The decoder's objective is to estimate the hidden x_t enrollment variables.

The factor graph breaks down into three pieces. At the bottom of Fig. 7 is the code graph representing the $\mathbf{H}$ matrix (cf. (4)) that maps $\mathbf{x}$ into $\mathbf{s}$. At the top of Fig. 7 is the observation $\mathbf{y}$. In between $\mathbf{x}$ and $\mathbf{y}$ is our model of movement, deletion, and insertion. Each circle in the figure represents a variable node either observed ($\mathbf{s}$ and $\mathbf{y}$) or unobserved ($\mathbf{x}$, $\mathbf{h}$, and $\mathbf{z}$) that need to be estimated. The vector $\mathbf{h}$ is a vector of binary variables each indicating the current belief (at a given point in the decoding process) whether an enrollment minutiae at position t is deleted. If a probe minutia is observed at position t (i.e., $y_t = 1$), then z_t indicates the current beliefs of what enrollment locations the minutiae might have come from and $\mathbf{z}_{\mathcal{N}(t)} = \{z_i | i \in \mathcal{N}(t)\}$ are the set of these variables in the neighborhood of enrollment position t.

The constraints between the variables and the priors that define the joint probability function of all system variables are represented by the polygon factor nodes. The constraints enforced by each are as follows. The prior on x_t is $p_{\square}(x_t)$. The prior on deletion is $p_{\blacksquare}(h_t)$. The prior on insertion is $p_{\triangledown}(z_t)$. The constraint that each enrollment minutia is paired with only a single probe minutia is enforced by the function node $\triangle$. In other word, $\triangle$ says that an enrollment minutiae can move to at most one position in the probe, or it can be deleted. Finally, in the reverse direction, $\Diamond$ constrains probe minutiae either to be paired with only a single enrollment minutiae or to be explained as an insertion. For a more detailed discussion of the statistical model see [27], [28]. The complete statistical model of the enrollment and probe biometrics is

$$p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y}) = p_{\mathbf{x}}(\mathbf{x}) p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x})$$
$$= \sum_{\{h_i\}} \sum_{\{z_i\}} \prod_t p_{\square}(x_t) p_{\blacksquare}(h_t) p_{\triangledown}(z_t) \triangle(x_t, h_t, \mathbf{z}_{\mathcal{N}(t)}) \Diamond(z_t, y_t).$$

The above statistical model of the biometrics is combined with the code graph. This yields the complete model used for decoding $p_{\mathbf{x},\mathbf{y},\mathbf{s}}(\mathbf{x}, \mathbf{y}, \mathbf{s}) = p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y}) \prod_j \oplus(s_j, \mathbf{x})$, where $\oplus(s_j, \mathbf{x})$ indicates that the mod-2 sum of s_j and the x_i connected to syndrome j by the edges of the LDPC code is constrained to equal zero. A number of computational optimizations must be made for inference to be tractable in this graph. See [27], [28] for details.

C. Experimental Evaluation of Security and Robustness

We use a proprietary Mitsubishi Electric (MELCO) database to evaluate our techniques. The database consists of a set of fingerprint measurements with roughly 15 measurements per finger. One measurement is selected as the enrollment, while decoding is attempted with the remaining 14 serving as probes. The locations of the minutiae points were quantized to reside in a 70×100 grid, resulting in a block-length $n = 7000$.

The mean and standard deviation of movement, deletions (p_{D}), and insertions (p_{I}) for the MELCO data set are plotted in Fig. V-C. The label $d = 1$ labels the probability an enrollment minutia moved a distance of one pixel in either the vertical or horizontal directions or both (i.e., the $\max$- or ∞-norm). These parameters are used to set parameter values in the factor graph.

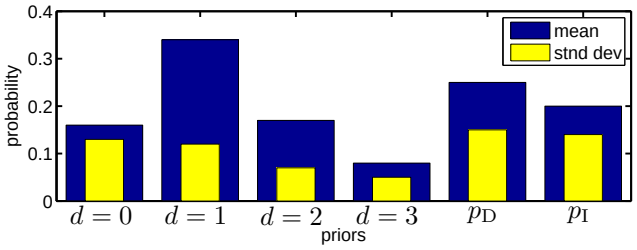


Fig. 8. Empirical movement statistics.

A summary test results are given in Table I. Results are categorized by the number of minutiae in the enrollment print. To first order, this is a measure of the randomness of the enrollment biometric. As an estimate of $H(\mathbf{x})$, we say that if a fingerprint has, e.g., 33 minutiae its entropy is $7000 \times H_B(33/7000) = 7000 \times 0.0432 = 302$. Each row in the table tabulates results for enrollment biometrics with the number of minutiae indicated in the first column. The second column indicates how many users had that number of minutiae in their enrollment biometric.

In the security-robustness trade-off developed in Section III-C, it was found that holding all other parameters constant (in particular the rate of the error-correcting code) security should increase and robustness decrease as the biometric entropy increases. To test this, we use LDPC codes of rate $R_{\mathrm{LDPC}} = 0.94$ and length-7000 for all syndrome calculations. The second and third groups of columns, labelled "False Negatives" and "False Positives" bear out the theoretic analysis. As the number of enrollment minutiae in a given fingerprint increase, the FRR goes up while the FAR drops. All non-enrollment probes of the given user are used to calculate FRR. Summing the "# tested" column under FRR gives 8111, which is roughly equal to the sum of the number of users (579) times the number of probes per user (roughly 14). To calculate the FRR we test the enrollment biometric uniformly against other users' biometrics. Note that for all results it is assumed that the fingerprints in the database are pre-aligned.[7]

The final group of columns in Table I is labelled "Security". Here, we quantify the information theoretic security for the prototype. From (5) and recalling that the length of the biometric is $n = 7000$, the number of bits of security is

$$H(\mathbf{x}|\mathbf{s}) = H(\mathbf{x}) - kH(s)$$
$$= 7000H(x) - 7000(1 - R_{\mathrm{LDPC}})H(s). \quad (6)$$

[7]We align fingerprints using a simple greedy minutiae-matching approach over a number of vertical and horizontal shifts (there was no rotational offset in the dataset). More generally, alignment would have to be done blindly prior to syndrome decoding. This is not as difficult as it may seem at first. For instance, many fingers have a "core point" and orientation in their pattern that can be used to define an inertial coordinate system in which to define minutiae locations. Doing this independently at enrollment and at verification would yield approximate pre-alignment. The movement part of the factor graph model is be able to compensate for small residual alignment errors.

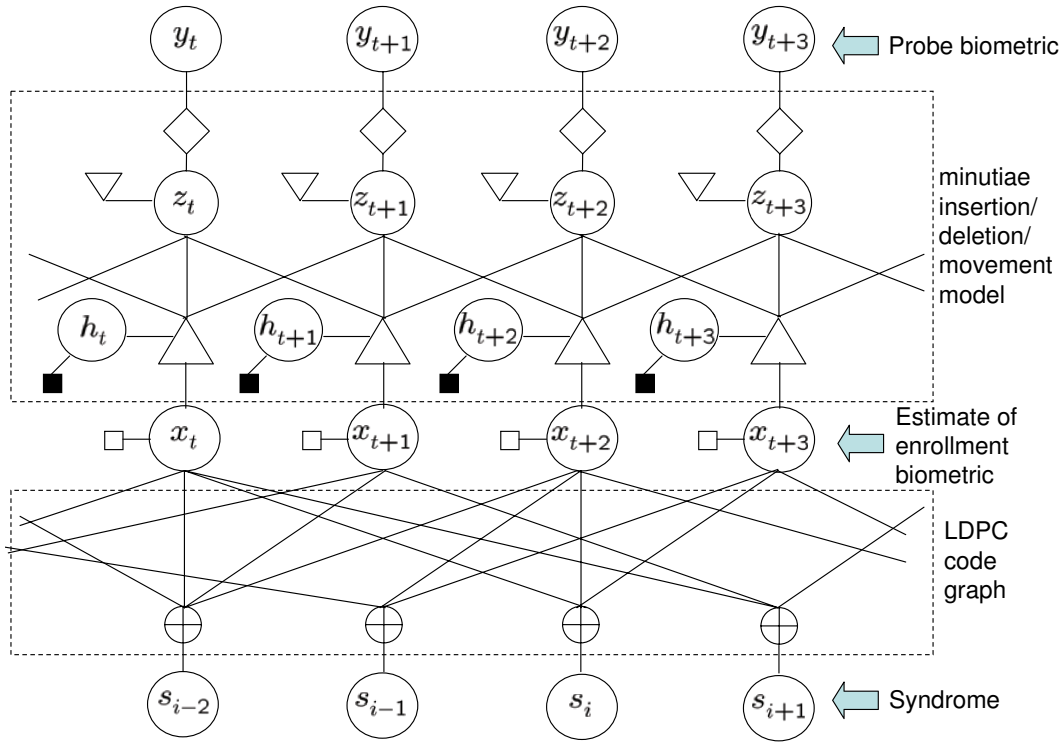


Fig. 7. Factor graph of minutiae movement model.

Enrollment		False Negatives		False Positives		Security		
# minutiae	# users	FRR	# tested	FAR	# tested	$H(x)$	$H(s)$	# bits
31	195	0.11	2736	0.0098	110000	0.0410	0.682	0.5
32	139	0.13	1944	0.0032	78000	0.0421	0.693	3.6
33	107	0.15	1506	0.0024	60000	0.0432	0.701	8.2
34	79	0.20	1101	0.0011	44000	0.0443	0.711	11.6
35	59	0.32	824	0.0003	33000	0.0454	0.716	17.2

TABLE I

TEST PARAMETERS, FRR AND FAR RESULTS FOR FULL MODEL DECODING WORKING ON MELCO DATA AT ENCODING RATE $R_{\mathrm{LDPC}} = 0.94$.

Equation (6) follows from our model that the underlying source is i.i.d. so $H(\mathbf{x}) = 7000H(x)$ and because we use syndrome codes via (4) the number of syndromes $k = 7000(1 - R_{\mathrm{LDPC}})$. Using $R_{\mathrm{LDPC}} = 0.94$ and substituting the values for $H(x)$ and $H(s)$ from the different rows of Table I into (6) gives the bits of security for this system, which are tabulated in the last column of the table.

D. Remarks on Modeling Approach

This section describes a secure fingerprint biometrics scheme in which an LDPC code graph was augmented with a second graph that described the "fingerprint channel" relating the enrollment to the probe biometric. A number of improvements are possible. For example, we implement an LDPC code designed for a binary symmetric channel (BSC). This design is not tuned to the fingerprint channel model. One possible improvement is to refine the design of the LDPC to match that channel. In general however, while the "fingerprint channel" is a reasonable model of the variations between the enrollment and probe fingerprints, the techniques developed are specific to the feature set and the resulting inference problem is complex and non-standard. In addition, higher levels of security are desired. For these reasons, we take a different approach in the next section that aims to redesign the feature extraction algorithm to yield biometric features that are well-matched to a standard problem of syndrome decoding.

VI. FINGERPRINT SYSTEM: TRANSFORMATION APPROACH

In this section we aim to revamp the feature extraction algorithm to produce biometric features with statistics well-matched to codes designed for the BSC. Since the construction of LDPC codes for the BSC is a deeply-explored and well-understood topic, we are immediately able to apply that body of knowledge to the secure biometrics problem. We believe this is a more promising approach, in part because the design insights we develop can be applied to building transforms for other biometric modalities. In contrast, the biometric channel model developed in Section V-B is specific to fingerprints and minutiae points. In addition, the system we describe for fingerprints in this section achieves a higher level of information-theoretic security.

The transformation-based secure fingerprint biometrics scheme is depicted in Fig. 9. In Section 5, the function $f_{\text{feat}}(\cdot)$ extracted minutiae maps from the enrollment and probe fingerprints. Here, in addition to minutiae extraction, the $f_{\text{feat}}(\cdot)$ box also encompasses a feature transformation algorithm that converts the 2-D minutiae maps to 1-D binary feature vectors. The central idea is to generate binary feature vectors that are i.i.d. Bernoulli(0.5), independent across different users but such that different measurements of the same user are related by a binary symmetric channel with crossover probability p (BSC-p), where p is much smaller than 0.5. This is one of the standard channel models for LDPC codes and therefore standard LDPC codes can be used for Slepian-Wolf coding of the feature vectors. We emphasize that the feature transformation we now present is made public and is *not* assumed to provide any security – in contrast to some of transformation-based techniques discussed in Section II.

A. Desired Statistical Properties of Feature Vectors

We aim to have a feature vector that possesses the following properties:

1) A bit in a feature vector representation is equally likely to be a 0 or a 1. Thus, $\Pr\{x_i = 0\} = \Pr\{x_i = 1\} = 1/2$ and $H(x_i) = 1$ bit for all $i \in \mathcal{I} = \{1, 2, \ldots, n\}$.
2) Different bits in a given feature vector are independent of each other, so that a given bit provides no information about any other bit. Thus, the pairwise entropy $H(x_i, x_j) = H(x_i) + H(x_j) = 2$ bits for all $i \neq j$ where $i, j \in \mathcal{I}$. This property, along with the first property, ensures that the feature vector can not be compressed further, i.e., it presents the maximum possible uncertainty for an attacker who has to guess a portion of a feature vector given some other portion.
3) Feature vectors $\mathbf{x}$ and $\mathbf{y}$ from different fingers are independent of each other, so that one person's feature vector provides no information about another person's feature vector. Thus, the pairwise entropy $H(x_i, y_j) = H(x_i) + H(y_j) = 2$ bits for all $i, j \in \mathcal{I}$.
4) Feature vectors $\mathbf{x}$ and $\mathbf{x}'$ obtained from different readings of the same finger are statistically related by a BSC-p. If p is small, it means that the feature vectors are robust to repeated noisy measurements with the same finger. Thus, $H(x_i'|x_i) = H(p)$ for all $i \in \mathcal{I}$.

The last property ensures that a Slepian-Wolf code with an appropriately chosen rate then makes it possible to estimate the enrollment biometric when provided with feature vectors from the enrollee. At the same time, the chosen coding rate makes it extremely difficult (practically impossible) to estimate the enrollment biometric when provided with feature vectors from an attacker or from a different user. To show that the resulting biometrics system is information theoretically secure, proceed just like in (3) to obtain

$$\begin{aligned} H(\mathbf{x}|\mathbf{s}) &= H(\mathbf{x}, \mathbf{s}) - H(\mathbf{s}) = H(\mathbf{x}) - H(\mathbf{s}) \\ &= H(\mathbf{x}) - nR_{\text{SW}} = n\left(H(x_i) - R_{\text{SW}}\right) \\ &= n(1 - R_{\text{SW}}) = nR_{\text{LDPC}} > 0 \end{aligned} \tag{7}$$

where the last two equalities follow from properties 1 and 2, and R_{LDPC} is the rate of the LDPC code used. Thus, the higher the LDPC code rate, the smaller is the probability of successful attack conditioned on an observation of $\mathbf{s}$. Moreover, $H(\mathbf{x}|\mathbf{s}) > 0$ and hence $nR_{\text{SW}} < H(\mathbf{x})$ implies that, if properties 1-4 are satisfied, the system has positive information-theoretic security for any LDPC code rate.

B. Feature Transformation Algorithm

To extract n bits from a minutiae map, it suffices to ask n "questions," each with a binary answer. A general framework to accomplish this is shown in Fig. 10. First, n operations are performed on the biometric to yield a non-binary feature representation that is then converted to binary by thresholding. As an example, one can project the minutiae map onto n orthogonal basis vectors and quantize the positive projections to 1s and negative projections to 0s.

In the implementation we now describe, the n operations count the number of minutiae points that fall in randomly chosen cuboids in $X - Y - \Theta$ space (x-position, y-position, θ-minutia-orientation), as shown in Fig. 10-(b). To choose a cuboid, an origin is selected uniformly at random in $X-Y-\Theta$ space, and the dimensions along the three axes are also chosen at random.

Next, define the threshold as the median of the number of minutiae points in the chosen cuboid, measured across the complete training set. A similar method is used for face recognition in [30]. The threshold value may differ for each cuboid based on its position and volume. If the number of minutiae points in a randomly generated cuboid exceeds the threshold, then a 1-bit is appended to the feature vector, otherwise a 0-bit is appended. We consider the combined operation of (a) generating a cuboid and (b) thresholding as equivalent to posing a question with a binary answer. With n such questions we get an n-bit feature vector.

The simplest way to generate feature vectors is to use the same questions for all users. In the sequel, we consider a more advanced approach in which the questions are user-specific. The rationale behind using user-specific questions is that some questions are more robust (reliable) than others. In particular, a question is robust if the number of minutiae points in a cuboid is much greater than or much less than the median calculated over the entire dataset. Thus, even if there is spurious insertion or deletion of minutiae points when a noisy measurement of the same fingerprint is provided at a later time, the answer to the question (0 or 1) is less likely to change. On the other hand, if the number of minutiae points is close to the median, the 0 or 1 answer to that question is less reliable. Thus, more reliable questions result in a BSC-p intra-user channel with low p. Different users have a different set of robust questions, and we propose to use these while constructing the feature vector. We emphasize that for the purposes of security analysis, the set of questions used in the system is assumed public. An attacker who steals a set of syndromes and poses falsely as a user will be given the set of questions appropriate to that user. Our security analysis is not based in any way on the obscurity of the questions, but rather on the information-theoretic difficulty of recovering the biometric given only the stolen syndromes.

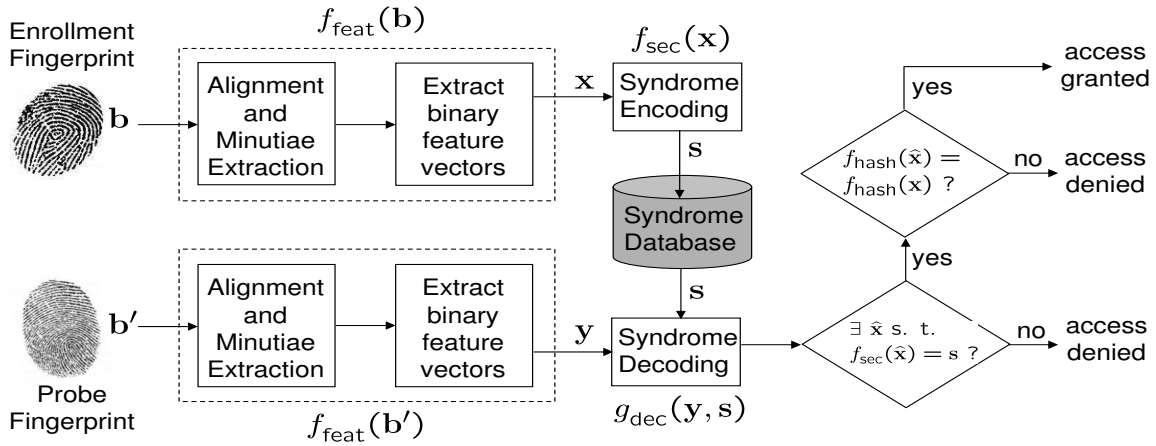


Fig. 9. Robust feature extraction is combined with syndrome coding to build a secure fingerprint biometrics system.

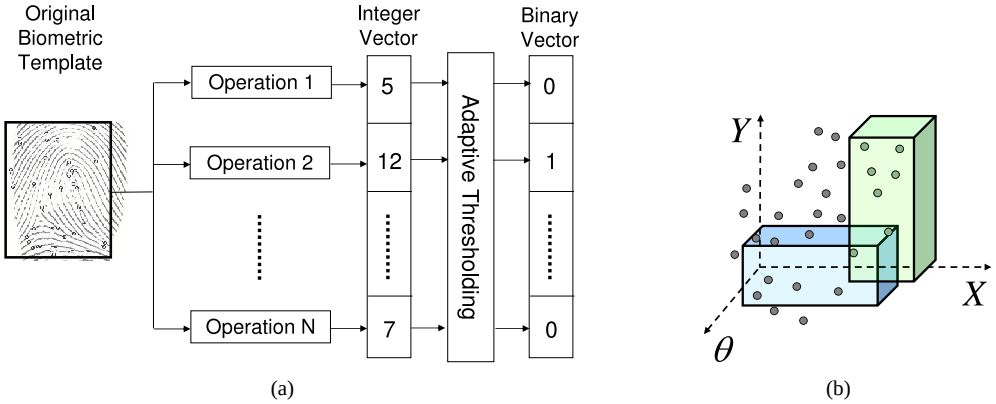


Fig. 10. (a) n questions can be asked by performing n operations on the biometric followed by thresholding. In our scheme, the operation involves counting the minutiae points in a randomly generated cuboid. (b) To obtain a binary feature vector, the number of minutiae points in a cuboid is thresholded with respect to the median number of minutiae points in that cuboid calculated over the entire dataset. Overlapping cuboid pairs will result in correlated bit pairs. For details about eliminating bit pairs with very high correlation, the reader is referred to [29].

For a given user i, the *average* number of minutiae points $\bar{m}_{i,j}$ in a given cuboid $\mathcal{C}_j$ is calculated over repeated noisy measurements of the same fingerprint. Let m_j and σ_j be the median and standard deviation of the number of minutiae points in $\mathcal{C}_j$ over the dataset of all users. Then, let $\Delta_{i,j} = (\bar{m}_{i,j} - m_j)/\sigma_j$. The magnitude, $|\Delta_{i,j}|$ is directly proportional to the robustness of the question posed by cuboid $\mathcal{C}_j$ for user i. The sign of $\Delta_{i,j}$ determines whether the cuboid $\mathcal{C}_j$ should be placed into $\mathcal{L}_{0,i}$, a list of questions with a 0 answer for user i, or into $\mathcal{L}_{1,i}$, a list of questions with a 1 answer for user i. Both these lists are sorted in the decreasing order of $|\Delta_{i,j}|$. Now, a fair coin is flipped to choose between $\mathcal{L}_{0,i}$ and $\mathcal{L}_{1,i}$ and the question at the top of the chosen list is stored on the device. After n coin flips, approximately $n/2$ of the most robust questions from each list will be stored on the device. This process is repeated for each enrolled user i.

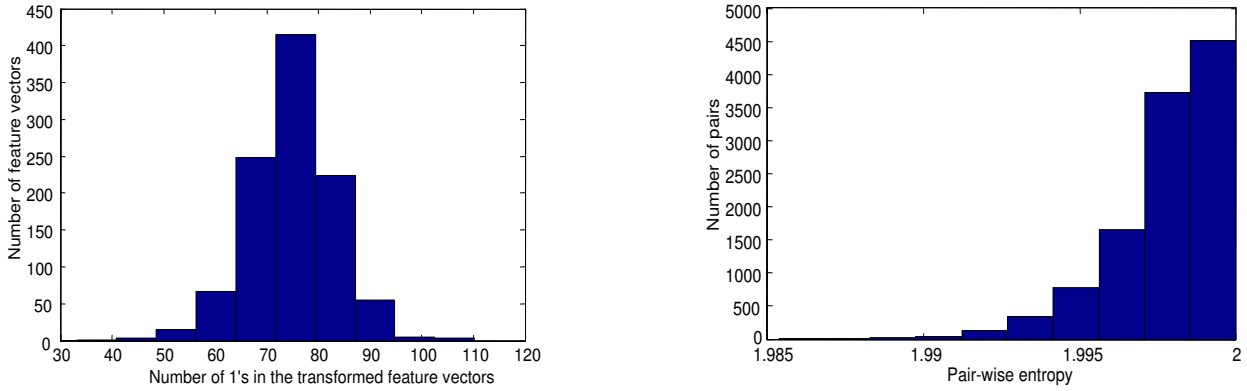
C. Experimental Evaluation of Security and Robustness

In the following experiments, we use the same Mitsubishi Electric fingerprint database as described in the previous section, which contains minutiae maps of 1035 fingers with 15 fingerprint samples taken from each finger. The average number of minutiae points in a single map is approximately 32. As before, all fingerprints are pre-aligned. To measure the extent to which the desired target statistical properties in Section VI-A are achieved, we examine the feature vectors obtained from the minutiae maps according to the method described in Section VI-B. The n most robust questions were selected to generate the feature vectors, with n ranging from 50 to 350. Fig. 11 shows the statistical properties of the feature vectors for n=150. As shown in Fig. 11(a), the histogram of the average number of 1-bits in the feature vectors is clustered around $n/2 = 75$. Fig. 11(b) shows that the pair-wise entropy measured between bits of different users is very close to 2 bits. Thus, bits are nearly pairwise independent and nearly uniformly distributed, approximating property 1.

In order to measure the similarity or dissimilarity of two feature vectors, the normalized Hamming distance (NHD) is used. The NHD between two feature vectors $\mathbf{x}$ and $\mathbf{y}$, each having length n, is calculated as follows:

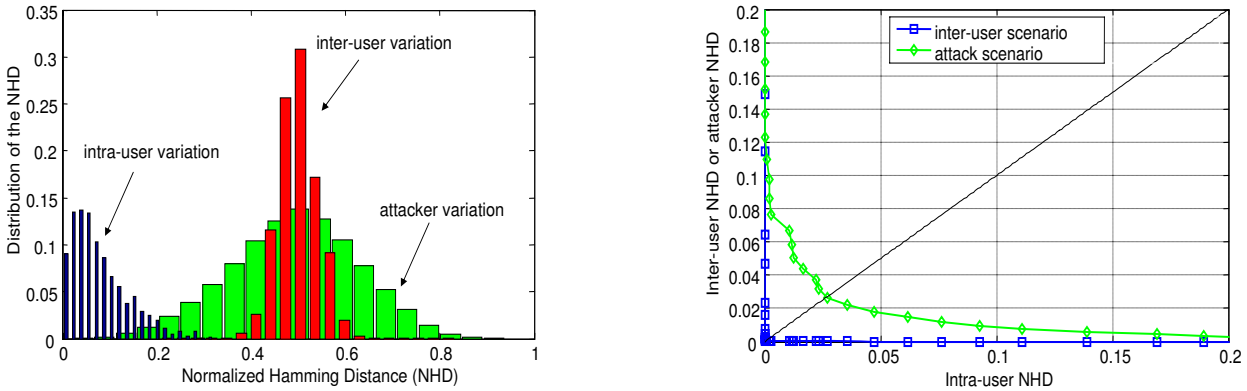
$$\text{NHD}(\mathbf{x}, \mathbf{y}) = \frac{1}{n} \sum_{i=1}^{n} (x_i \oplus y_i)$$

where $\oplus$ is summation modulo 2. The plot of Fig. 12(a) contains three histograms: (1) The intra-user variation is the distribution of the average NHD measured pairwise over 15



(a) (b)

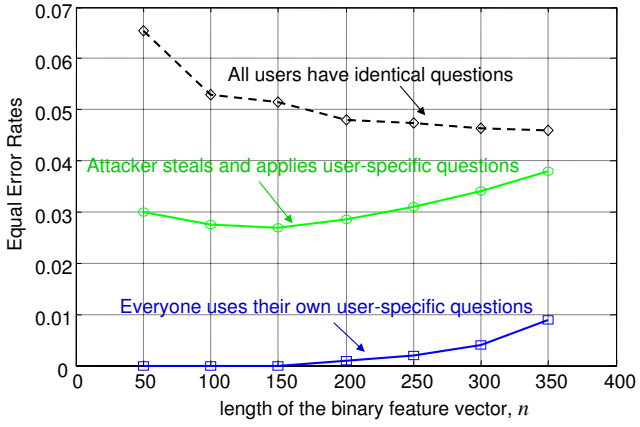
Fig. 11. (a) Histogram of the number of ones in the feature vectors for n=150 is clustered around $n/2 = 75$. (b) The pairwise entropy measured across all pairs and all users is very close to 2 bits.



(a) (b)

Fig. 12. (a) The Normalized Hamming Distance (NHD) between feature vectors shows clear separation within and across users. (b) The tradeoff between intra-user NHD and inter-user NHD is plotted by sweeping a threshold NHD across the histograms in Fig. 12(a). For n=150, equal error rate is 0.027 when the attacker has access to the victim's questions and is nearly zero when the attacker is impersonating a victim without knowing his specific questions.

samples of the same finger, (2) The inter-user variation is the distribution of the NHD averaged over all possible pairs of users, each with his own specific set of questions (3) The attacker variation is the NHD for the case in which an attacker attempts to identify himself as a given user i, while using a different fingerprint $j \neq i$, but while using the 150 robust questions of user i. As seen in the figure, there is a clean separation between the intra-user and inter-user NHD distributions, and a small overlap between the intra-user and attacker distributions. One way to ascertain the effectiveness of the feature vectors is to choose different threshold NHDs in Fig. 12(a) and plot the intra-user NHD against the inter-user NHD. This tradeoff between intra-user NHD and inter-user NHD is shown in Fig. 12(b) both for the case in which every user employs specific questions and for the case in which an attacker uses the questions stolen from the user being impersonated. A metric for evaluating plots such as Fig. 12(b) is the "equal error rate (EER)", which is defined as the point where intra-user NHD equals inter-user NHD. A lower EER indicates a superior tradeoff. Fig. 13 plots the EER for various values of n. Observe that user-specific questions provide a significantly lower EER than using the same questions for all users irrespective of the robustness of the questions. Even if



(a)

Fig. 13. User-specific questions result in lower EER than common questions, even if the user-specific questions are given to the attacker.

the attacker is provided with the user-specific questions, the resulting EER is lower than the case in which everybody has the same questions.

Based on the separation of intra-user and inter-user distributions, we expect that a syndrome code designed for a BSC-

n	BSC crossover probability, p	R_{LDPC}	FRR after syndrome coding	FAR after syndrome coding	No. of Bits of security
100	0.1	0.3	0.23	0.0001	30
150	0.13	0.2	0.11	0.0001	30
200	0.2	0.15	0.14	0.0014	30
250	0.2	0.125	0.15	0.0035	31.25

TABLE II

SYNDROME CODING WITH AN APPROPRIATE LDPC CODE GIVES AN INFORMATION-THEORETICALLY SECURE BIOMETRICS SYSTEM WITH LOW FRR AND EXTREMELY LOW FAR.

p, with appropriate $p < 0.5$ would authenticate almost all genuine users while rejecting almost all impostors. Table II shows the FRR and FAR[8] for overall syndrome coding with different values of n and p. These FAR and FRR values are measures of the security-robustness tradeoff of the distributed biometric coding system. The LDPC code rate is chosen so as to provide about 30 bits of security. This restriction on the LDPC code rate in turn places a restriction on how large p can be, especially for small n. Due to this restriction, the FRR is relatively large for $n = 100$. The lowest FRR is achieved for $n = 150$. As n increases, less robust questions need to be employed, so the statistical properties of the feature vectors diverge from those in Section VI-A. Thus, the FRR increases again when n becomes too large.

Compare the FRR, FAR and number of bits of security reported in Table II with those reported in Section V. We observe that the FRR and FAR are comparable, but the transformation approach described in this section provides a higher number of bits of security compared to the model-based approach of Section V (see final column of Table I). The reason for this improved security-robustness tradeoff is that the statistical properties of the transformed feature vectors are intentionally matched to the standard LDPC code for a binary symmetric channel.

VII. SUMMARY

This chapter demonstrates that the principles of distributed source coding can be successfully applied to the problem of secure storage of biometrics. A Slepian-Wolf framework is used to store a secure version of the biometric template data collected at enrollment and to recover the enrollment template at authentication. The trade-off between security and robustness in this framework is formally defined and discussed, and sample implementations based on iris and fingerprint data validate the theory.

While iris data tends to be relatively well behaved and exhibits easily modeled sample-to-sample variability (both between samples of the same user and across users) the same can not be said of fingerprints. It is shown that the fingerprint noise channel is far removed from the standard bit-flipping (e.g., BSC) channel model of communication systems. The design of a secure system for such biometric modalities therefore requires additional attention. Two approaches are discussed. The first design is based on using a sparse binary matrix representation of minutiae locations and developing a model of minutiae movement that can be combined with a graphical representation of a linear code. Although this approach does not yet yield satisfactory performance in terms of security and robustness, it does reveal various factors that affect performance and provides valuable insight that motivates the transform-based approach of Section VI.

In the latter approach, a transform is designed to convert the fingerprint feature set into a binary vector with desirable statistical properties, in the sense of being well-matched to well-understood channel coding problems. The resultant design yields very low false-acceptance and false-rejection rates. Further, it ensures operation well into the information-theoretically secure region. We believe this to be a powerful concept that will allow extension of this framework to other biometric data. It may also prove useful in resolving performance issues with other Slepian-Wolf inspired systems.

Besides further improving security and robustness, there are a number of additional open research issues. As one example, the designs presented in this chapter assumed that the biometric data is pre-aligned. In practice, this is not the case and biometric data must be aligned blindly, i.e., without access to other reference data. One research trajectory is the design of such algorithms. An alternative to blind alignment is the design of a translation- and rotation-invariant feature set. A second aspect of the secure biometrics that has not received much attention concern multi-biometric systems. In these systems multiple biometrics are collected at enrollment and verification – such as both iris and fingerprint. The measurements are fused to improve overall robustness and security. This particular combination and some encouraging results are presented by Nandakumar in [31]. However, the topic has yet to be studied in the context of a Slepian-Wolf coding system.

As the use of biometrics become more widespread, the incentive to attack biometric systems will grow. Assuming the technology for securing biometric data is sufficiently mature, it would be natural to standardize the template protection design. Such work is within the scope of ISO/IEC JTC1/SC37, which is an international standardization committee on biometrics. Open issues to be handled by this committee would range from quantifying the inherent entropy and security limits of biometric data to remote authentication scenarios.

As a final note, the biometric system described in this chapter is one example where a noisy version of an original signal is available at the decoder for the purpose of authentication. This type of setup is extended to the problem of image authentication following similar principles [32]. We believe

[8]While determining the FAR, if an input feature vector $\widehat{\mathbf{a}}$ satisfies the syndrome, it is counted as a false accept. This is a conservative FAR estimate since any $\widehat{\mathbf{a}}$ for which $f_{\text{hash}}(\widehat{\mathbf{a}}) \neq f_{\text{hash}}(\mathbf{a})$ is denied access.

that there are many such applications of this nature in which the principles of distributed source coding can be applied.

REFERENCES

[1] D. Slepian and J. K. Wolf, "Noiseless Coding of Correlated Information Sources," *IEEE Trans. Information Theory*, pp. 471–480, Jul 1973.

[2] N. Ratha, J. Connell, R. Bolle, and S. Chikkerur, "Cancelable Biometrics: A Case Study in Fingerprints," in *Intl. Conf. on Pattern Recognition*, 2006, pp. 370–373.

[3] N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle, "Generating Cancelable Fingerprint Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 561–572, 2007.

[4] K. Sakata, T. Maeda, M. Matsushita, K. Sasakawa, and H. Tamaki, "Fingerprint Authentication based on Matching Scores with Other Data," in *Lecture Notes in Computer Science*, ser. LNCS, vol. 3832, 2005, pp. 280–286.

[5] A. Teoh, A. Gho, and D. Ngo, "Random Multispace Quantization as an Analytic Mechanism for Biohashing of Biometric and Random Identity Inputs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1892–1901, 2006.

[6] R. Ahlswede and I. Csiszar, "Common Randomness in Information Theory and Cryptography I: Secret Sharing," *IEEE Trans. Information Theory*, vol. 39, no. 4, pp. 1121–1132, Jul 1993.

[7] 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*, May 1998, pp. 148–157.

[8] A. Juels and M. Wattenberg, "A Fuzzy Commitment Scheme," in *CCS '99: Proceedings of the 6th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 1999, pp. 28–36.

[9] F. Hao, R. Anderson, and J. Daugman, "Combining Cryptography with Biometrics Effectively," University of Cambridge, Tech. Rep. UCAM-CL-TR-640, July 2005.

[10] A. Juels and M. Sudan, "A Fuzzy Vault Scheme," in *Proc. International Symposium on Information Theory*, Lausanne, Switzerland, July 2002, p. 408.

[11] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure Smartcard-based Fingerprint Authentication," in *Proc ACM SIGMM workshop on biometrics methods and applications*, 2003.

[12] S. Yang and I. M. Verbauwhede, "Secure Fuzzy Vault-based Fingerprint Verification System," in *Asilomar Conference on Signals, Systems, and Computers*, vol. 1, November 2004, pp. 577–581.

[13] U. Uludag, S. Pankanti, and A. K. Jain, "Fuzzy Vault for Fingerprints," in *Audio- and Video-Based Biometric Person Authentication, 5th International Conference, AVBPA 2005, Hilton Rye Town, NY, USA, July 20-22, 2005, Proceedings*, ser. Lecture Notes in Computer Science, vol. 3546. Springer, 2005.

[14] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based Fuzzy Vault: Implementation and Performance," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 4, pp. 744–757, Dec 2007.

[15] D. Maio, D. Maltoni, J. Wayman, and A. K. Jain, "FVC2002: Second Fingerprint Verification Competition," in *International Conference on Pattern Recognition*, August 2002, pp. 811–814.

[16] U. Uludag, S. Pankanti, S. Prabhakar, and A. K. Jain, "Biometric Cryptosystems: Issues and Challenges," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 948–960, June 2004.

[17] 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.

[18] T. M. Cover, "A Proof of the Data Compression Theorem of Slepian and Wolf for Ergodic Sources," *IEEE Trans. Inform. Theory*, vol. 21, no. 2, pp. 226–228, Mar 1975.

[19] R. G. Gallager, "Source Coding with Side Information and Universal Coding," Massachusetts Institute of Tech., Tech. Rep. LIDS P-937, 1976.

[20] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[21] "CASIA Iris Image Database collected by Institute of Automation, Chinese Academy of Sciences." [Online]. Available: http://www.sinobiometrics.com

[22] L. Masek, "Recognition of Human Iris Patterns for Biometric Identification," Bachelors Thesis, University of Western Australia, 2003.

[23] 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, no. 2, pp. 619–637, February 2001.

[24] E. Martinian, S. Yekhanin, and J. S. Yedidia, "Secure Biometrics via Syndromes," in *Allerton Conf.*, Monticello, IL, Sep 2005, pp. 1500–1510.

[25] A. K. Jain, L. Hong, and R. Bolle, "On-line fingerprint verification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 302–314, April 1997.

[26] F. R. Kschischang, B. J. Frey, and H. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, February 2001.

[27] S. C. Draper, A. Khisti, E. Martinian, A. Vetro, and J. S. Yedidia, "Secure Storage of Fingerprint Biometrics using Slepian-Wolf Codes," in *Inform. Theory and Apps. Work.*, *UCSD*, San Diego, CA, Jan 2007.

[28] ——, "Using Distributed Source Coding to Secure Fingerprint Biometrics," in *Int. Conf. Acoutics Speech Signal Proc.*, Honolulu, HI, Apr 2007, pp. II–(129–132).

[29] Y. Sutcu, S. Rane, J. S. Yedidia, S. C. Draper, and A. Vetro, "Feature Transformation for a Slepian-Wolf Biometric System based on Error Correcting Codes," in *Computer Vision and Pattern Recognition (CVPR) Biometrics Workshop*, Anchorage, AL, Jun 2008, pp. 1–6.

[30] T. Kevenaar, G. Schrijen, M. V. der Veen, A. Akkermans, and F. Zuo, "Face Recognition with Renewable and Privacy Preserving Binary Templates," *Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, pp. 21–26, October 2005.

[31] K. Nandakumar, "Multibiometric Systems: Fusion Strategies and Template Security," *Ph.D. Thesis, Michigan State University*, 2008.

[32] Y. C. Lin, D. Varodayan, and B. Girod, "Image Authentication based on Distributed Source Coding," in *International Conference on Image Processing*, San Antonio, TX, Sep 2007, pp. III–(5–8).