USING DISTRIBUTED SOURCE CODING TO SECURE FINGERPRINT BIOMETRICS

Stark C. Draper, Ashish Khisti, Emin Martinian, Anthony Vetro, Jonathan S. Yedidia

Mitsubishi Electric Research Laboratories (MERL)
201 Broadway Ave, Cambridge MA, 02139

ABSTRACT

We describe a method to encode fingerprint biometrics securely for use, e.g., in encryption or access control. The system is secure because the stored data does not suffice to recreate the original fingerprint biometric. Therefore, a breach in database security does not lead to the loss of biometric data. At the same time the stored data suffices to validate a probe fingerprint. Our approach is based on the use of distributed source coding techniques implemented with graph-based codes. We present a statistical model of the relationship between the enrollment biometric and the (noisy) biometric measurement taking during authentication. We describe how to validate or reject a candidate biometric probe given the probe and the stored encoded data. We report the effectiveness of our method as tested on a database consisting of 579 data sets, each containing roughly 15 measurements of a single finger. We thereby demonstrate a working secure biometric system for fingerprints.

Index Terms— Biometric, fingerprint, distributed coding, Slepian-Wolf coding, belief propagation decoding

1. INTRODUCTION

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 and with the latter poorly-chosen or forgotten passwords.

Computer-verifiable biometrics provide a third approach. In these systems a sensor measures a biological feature of a person, for example, a fingerprint or an iris scan. It then compares the new sample, termed the probe, to a stored sample, termed the enrollment. If the samples match then, depending on the application, the person could be granted access or given a cryptographic key that is a function of the biometric. Advantages of biometrics include the fact that they cannot be forgotten, they can be hard to guess, and they can be difficult to forge.

Biometrics have certain characteristics that pose novel challenges and create new security holes. A central characteristic that differentiates biometrics from passwords is that each time a biometric is measured the observation differs. 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 authentication systems must be robust to such variations.

S. Draper, A. Vetro and J. Yedidia are with the Mitsubishi Electric Research Laboratories (MERL), {draper, avetro, yedidia}@merl.com. A. Khisti and E. Martinian were with MERL. A. Khisti is with the MIT department of Electrical Engineering and Computer Science, 77 Mass. Ave, Cambridge MA, 02139, khisti@mit.edu. E. Martinian is with Bain Capital, 111 Huntington Ave., Boston, MA, 02199, emin@alum.mit.edu.

Most biometric authentication systems deal with such variability by relying on pattern recognition. To perform recognition the enrollment biometric is stored on the device. This results in a serious security hole. If a malicious attacker gains access to the device, the attacker also gains access to the biometric. In contrast, passwords are not stored "in-the-clear". Instead, only the hash of a password is stored. When a user types in a password the computer compares the hash of the probe password to the stored hash Only if they match is access granted. Since the hash is effectively impossible to invert, security is not compromised even if an attacker learns the stored hash. Several researchers have attempted to develop "secure" biometric systems with similar characteristics.

Davida, Frankel, and Matt [1] consider the use of error correction coding as a solution to the secure biometrics problem. Juels and Sudan [2] introduce the idea of a fuzzy vault to formalize the use of error correcting codes for such applications. Some constructions for fingerprint biometrics exist, e.g., [3–5], but yield high false reject rates (FRRs). A main stumbling block is how to model and exploit the statistical relationship between enrollment and probe. From an information theoretic perspective the secure biometric problem is a problem of "common randomness" [6]. Different parties observe correlated random variables (the enrollment and the probe) and then attempt to agree on a shared secret key (the enrollment biometric). The basic tool used to extract the secret is a distributed source code [7].

Our formulation and proposed solution build on both sets of works. In our implementation we develop a statistical model of the "fingerprint channel" relating the enrollment to the probe, and use a graphical code to compress and scramble the enrollment probe. Iterative decoding using belief propagation (BP) is performed across *both* graphs. This successfully captures both the structure of the code and that of the measurement channel. Our initial work in this area considered iris biometrics [8].

The outline of the remainder of the paper is as follow. In Section 2 we describe the operation of the system, identify an appropriate biometric feature set, develop a fingerprint channel model, and describe a natural attack on secured biometric systems. In Section 3 we test our channel model on synthetically-generated data, demonstrate that simpler models do not lead to good decoding performance. We then evaluate our system on a database of roughly 8100 test fingerprints. We conclude and discuss future work in Section 4. In this paper we focus on the implementation and evaluation of our prototype system. More details on system security can be found in [9].

2. FEATURE SET AND STATISTICAL MODELING

In this section we describe the operation of our system and its required components. At enrollment we measure the original biometric $\mathbf{x}$ and compress it into a scrambled "syndrome" $\mathbf{s}$ which is our secured biometric. During authentication we measure $\mathbf{y}$, a noisy ver-

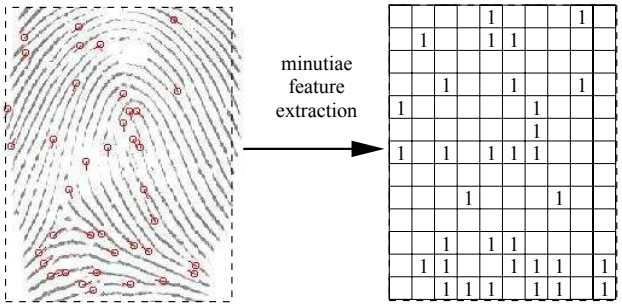

Fig. 1. Fingerprint and extracted feature vector.

sion of $\mathbf{x}$ and, from $\mathbf{s}$ and $\mathbf{y}$, estimate $\mathbf{x}$. Only if the estimate is perfect (verified by comparison of the cryptographic hash of the estimate with a hash of the original $\mathbf{x}$) is authentication given. The system stores the syndrome $\mathbf{s}$, the cryptographic hash of $\mathbf{x}$, and the joint distribution $p_{\mathbf{x},\mathbf{y},\mathbf{s}}(\mathbf{x},\mathbf{y},\mathbf{s})$, described below. In the rest of this section we specify an appropriate representation of fingerprints, develop the statistical model $p_{\mathbf{x},\mathbf{y},\mathbf{s}}(\mathbf{x},\mathbf{y},\mathbf{s})$, and conclude with a discussion of security.

2.1. Fingerprint representation

A popular method for working with fingerprint data is to extract a set of "minutiae points" and to perform all subsequent operations on them. Figure 1 gives an example of a fingerprint, the minutiae points, and the extracted feature vector that we work with. Each minutiae is a discontinuity in the ridge map of a fingerprint, indicated by the circles in the left-hand plot. These points are mapped to a list of triplets representing the spatial and angular coordinates of each minutiae point. We visualize the feature vector using a matrix as depicted in the right-hand plot. Each quantized coordinate corresponds to a particular location in the matrix. The presence of a minutiae is indicated by a '1'. More generally, instead of simply indicating the presence or lack of minutiae points, the entries could indicate the angles of enrolled minutiae points.

2.2. Modeling the movement of fingerprint minutiae

We create a statistical model for the fingerprint channel which captures three effects: (1) movement of enrollment minutiae when observed the second time in the probe, (2) deletions–minutiae observed at enrollment, but not during probe, and (3) insertions–"spurious" minutiae observed in probe, but not during enrollment.

Figure 2 depicts the factor graph [10] model we develop. The presence of a minutiae point at position t 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. For simplicity, the figure shows one-dimensional movement model. The results reported in this paper all use a two-dimensional movement model. We model the enrollment feature vector $\mathbf{x}$ as a Bernoulli-p_{p} independent identically distributed (i.i.d.) random vector. These prior probabilities are denoted by the white-square factor nodes (□).

For each position in the enrollment grid there is a corresponding position in the probe grid. The presence of a minutiae point at grid position t in the probe is represented by the binary random variable y_t taking on value $y_t = 1$.

Some minutiae observed during enrollment are not observed in the probe. The binary random variable h_t represents one such erasure. It takes on value $h_t = 1$ if x_t is erased. The black-square factor nodes (■) represent the prior probability of on h_t. We model $\mathbf{h}$ as an i.i.d. Bernoulli-p_{e} sequence.

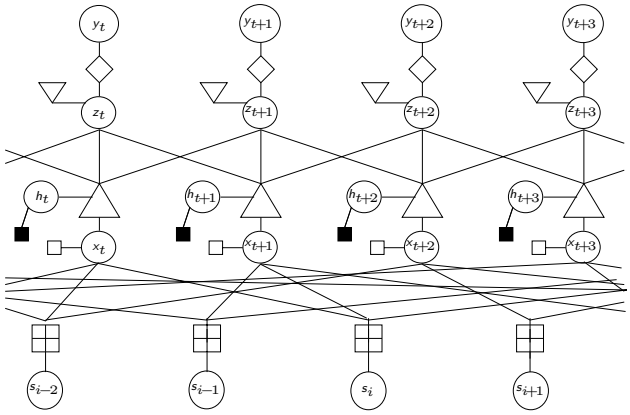

Fig. 2. Factor graph of minutiae movement model.

Our model captures the local elastic deformations in the skin that occur when a fingerprint is placed on a sensor. We assume that global translations and rotations of a fingerprint are corrected through a combination of pre-processing and a search over small (rigid) shifts. To model the elastic deformations with suitable accuracy, the model must capture the motion of the fingerprints about their enrollment positions.

For each enrollment position t the model specifies a neighborhood $\mathcal{N}(t)$ of positions to which the enrollment minutiae can move. The z_t variables in Fig. 2 capture the relative change in position of an enrollment minutiae, 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 upside-down triangle factor nodes (▽) represent the prior probability distribution both on minutiae movement and the event that a spurious minutiae is generated at this position. If a minutiae moves beyond its neighborhood, the model treats it as a deletion and an insertion.

The variables z_t take values in the set $z_t \in \{Ⓢ, *, \Delta\mathcal{N}(t)\}$. If $z_t = Ⓢ$, then a spurious minutiae unrelated to the enrollment was generated at position t in the probe. If $z_s = *$ there is no minutiae at position t in the probe (i.e., $y_t = 0$). The diamond factor nodes (◊) connecting each y_t to its corresponding z_t capture the notion that each probe minutiae y_t can only be non-zero if there is a corresponding $z_t \neq *$. Finally, $\Delta\mathcal{N}(t)$ is the set of relative shifts that define the possible movements, and the neighborhood $\mathcal{N}(t)$. For example, in the simple one-dimensional movement model of Fig. 2, $\Delta\mathcal{N}(t) = \{-1, 0, 1\}$. This definition of the z_t captures the continuity in the elastic deformation of the skin as only a single minutiae can move to each probe location.

Both the support of minutiae movement (the choice of the $\Delta\mathcal{N}(t)$) and the prior on the movement (the distribution on z_t) are design choices. While a larger neighborhood helps to capture the tails of minutiae movement, it also incurs greater computational complexity and adds loops to the graphical model. These extra loops can ultimately pose problems for the graph-based inference algorithm we used to decode. We use belief propagation (BP) as our decoding algorithm.

Each enrollment minutiae x_t is constrained to move only within its neighborhood $\mathcal{N}(t)$. Furthermore, it can move to only one point, and therefore can explain only a single minutiae point observed in the probe. The triangular factor nodes (△) in Fig. 2 capture these

movement constraints.

The complete model gives $p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y}) = p_{\mathbf{x}}(\mathbf{x})p_{\mathbf{y}|\mathbf{x}}(\mathbf{y}|\mathbf{x}) =$

$$\sum_{\{h_i\}}\sum_{\{z_i\}}\prod_t \square(x_t)\blacksquare(h_t)\nabla(z_t)\triangle(x_t,h_t,\mathbf{z}_{\mathcal{N}(t)})\Diamond(z_t,y_t).$$

In this paper the secure biometric $\mathbf{s}$ is a "syndrome" vector. Each syndromes s_j is the mod-2 sum of the enrollment variables x_t to which s_j is connected by a syndrome graph. The connections defining the syndrome graph are generated according to a low-density parity-check (LDPC) code. These are state-of-the-art channel codes. A main reason for using LDPCs is that they are well represented graphically. This makes it easy to merge their description into the graphical model movement to implement BP decoding. Each local constraint of the syndrome code $\boxplus(s_j,\mathbf{x})$ is an indicator function equaling one if the value of the syndrome s_j is compatible with $\mathbf{x}$ and zero otherwise. The complete model used for decoding $p_{\mathbf{x},\mathbf{y},\mathbf{s}}(\mathbf{x},\mathbf{y},\mathbf{s}) = p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})\prod_j \boxplus(s_j,\mathbf{x})$ is shown in Fig. 2.

Given the graphical model for $p_{\mathbf{x},\mathbf{y},\mathbf{s}}$, the raw message passing rules for use in belief propagation can be derived using standard techniques [10]. In order to make the computations tractable we introduce a number of computational optimizations. These optimizations exploit the particular structure of the messages, the graph, and the quantities being computed. Due to space constraints, we do not further discuss these optimizations here

2.3. Security

We now quantify the security of our system. Since the source $\mathbf{x}$ is a Bernoulli-$p_{\rm p}$ i.i.d. sequence, its entropy rate is $H_b(p_{\rm p})$. Say that we use a rate-$R_{\rm LDPC}$ code to encode the biometric. Then, an attacker that limits itself to guesses $\tilde{\mathbf{x}}$ that encode to the same secure biometric $\mathbf{s}$ will require $2^{n(R_{\rm LDPC}+H_b(p_{\rm p})-1)}$ guesses to identify the actual $\mathbf{x}$ with high probability. Coding rates $R_{\rm LDPC} > 1 - H(x) = 1 - H_b(p_{\rm p})$ give positive information theoretic security. The higher the rate the greater the secrecy, but also the more challenging the decoding problem. As long as $R_{\rm LDPC} < 1 - (1/n)H(\mathbf{x}|\mathbf{y})$ the probability of successful authentication converges to zero asymptotically in the block-length.

When $R_{\rm LDPC} < 1 - H_b(p_{\rm p})$ the system is not information-theoretically secure. However, recovering $\mathbf{x}$ from $\mathbf{s}$ can still be very difficult. This recovery is a "syndrome" decoding problem with $\mathbf{x}$ playing the role of the error sequence. Syndrome decoding requires storage of a look-up table of size $2^{n(1-R_{\rm LDPC})}$. In the results presented herein $n = 7000$ and we use a rate-0.94 LDPC. This means that the table size is larger than 2^{400}, so syndrome decoding is intractable. However, if $R_{\rm LDPC}$ is much smaller than $1 - H_b(p_{\rm p})$ other approaches can tractably recover $\mathbf{x}$.

We introduce the "zero-probe" attack to test this security. The attacker know $\mathbf{s}$, it knows the code structure, and it can use any attack it likes. The attacker guesses $\mathbf{y} = 0$ and uses BP to try to solve the syndrome decoding problem. If $R_{\rm LDPC}$ is small enough this BP-based attack will recover $\mathbf{x}$. However, when $R_{\rm LDPC}$ is below, but close to $1 - H_b(p_{\rm p})$ this attack fails. We report the efficacy of this attack, as well as that of the standard biometric attack of using some other fingerprint in conjunction with $\mathbf{s}$ to decode. The success rate of the latter attack is given by the false-acceptance rate (FAR). A more detailed analysis of system security is provided in [9].

3. EXPERIMENTAL RESULTS

We now report decoding performance on synthetic data as well as on a proprietary Mitsubishi Electric (MELCO) fingerprint database.

LDPC Rate	0-probe SAR	Simple FRR	Full Model, FRR		
			$p_{\rm e} = 0$	$p_{\rm s} = 0$	S & E
0.92	0.37	0.48			
0.93	0.23	0.76			
0.94	0.0	1.0	7.0e-3	9.4e-3	46e-3
0.95	0.0	1.0	4.2e-3	17e-3	79e-3

Table 1. Zero-probe SAR (successful attack rate), simple model FRR, and full-model FRR for synthetic test data. We do not simulate the full model for $R_{\rm LDPC} = 0.92, 0.93$ since the zero-probe attack is successful at those rates.

3.1. Synthetic data

The synthetic data model consists of a 70×100 grid of minutiae locations. The prior probability on the presence of a minutiae is $p_{\rm p} = 0.005$. The probability of an erasure is $p_{\rm e} = 0.2$. The probability of a spurious minutiae is $p_{\rm s} = 0.001$. We use a uniform prior on movement over a two-dimensional area with a maximum displacement of 2 in either the horizontal or vertical directions. Each (non-edge) minutiae has a neighborhood size $|\mathcal{N}(t)| = 25$.

The syndrome code is a randomly-generated LDPC code. For $R_{\rm LDPC} = 0.94$ all check nodes are of degree 50, 0.1% of the variable nodes are of degree 2, 99.8% are of degree 3, and 0.1% are of degree 4. For $R_{\rm LDPC} = 0.95$ all check nodes are of degree 80, 60% of the variable nodes are of degree 3, and 40% are of degree 8.

We test the effectiveness of the zero-probe attack by generating 10000 attacks on 10000 independently-generated syndromes at different code rates. The results in terms of SAR (successful attack rate) are reported in the second column of Table 1. We conclude that for $R_{\rm LDPC} \geq 0.94$ our system is safe from the zero-probe attack, although it is not information theoretically secure.

To test the necessity of the detailed minutiae movement model of Sec. 2 we evaluate a simpler statistical model. Erasures and spurious minutiae are modeled as in Sec. 2. But, instead of joint constraints on minutiae movement, we model minutiae movements as independent. If a minutiae is present in the probe at position t (i.e., $y_t = 1$), we model the likelihoods of any $x_{t'}$ such that $t \in \mathcal{N}(t')$ as equally likely. The resulting FRRs of BP decoding using this "simple model" are reported in Table 1. The performance in terms of error rates is comparable to (indeed, slightly worse than) that of the zero-probe attack (compare the FRR with one minus the zero-probe SAR).

Finally, to generate results for the full model, we sample 50 different codes at random from the degree distribution and test each on 1000 independent samples of $(\mathbf{x},\mathbf{y})$. FRR is averaged over all trials. To understand the relative impact of erasures and spurious minutiae, we also considered the case of movement and spurious only ($p_{\rm e} = 0$), and movement and erasures only ($p_{\rm s} = 0$). Erasures are seen to be more harmful than spurious insertions.

3.2. Tests on MELCO database

The MELCO database consists of measurements of 1000 fingers. Each of these 1000 data sets contains roughly 15 measurements of the corresponding finger. We select one of the measurements as the enrollment and try to decode using the remaining measurements as probes. All syndrome calculations use the rate $R_{\rm LDPC} = 0.94$ code described above. We simulated the performance of 1000 randomly-generated codes on synthetic data and used the 5 best in a round-robin manner to encode the MELCO data. This avoids particularly bad random codes. We now detail the experimental setup.

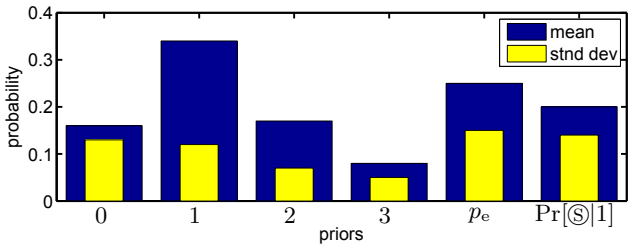

Fig. 3. Empirical movement statistics.

# enrolled minu. (ent)	Num. files	FRR		FAR	
		rate	probes	rate	probes
31 (0.0410)	195	11.6e-2	2736	0.98e-2	11e4
32 (0.0421)	139	13.3e-2	1944	0.32e-2	78e3
33 (0.0432)	107	14.9e-2	1506	0.24e-2	60e3
34 (0.0443)	79	20.2e-2	1101	0.11e-2	44e3
35 (0.0454)	59	32.3e-2	824	0.03e-2	33e3

Table 2. Test parameters, FRR and FAR results for full model decoding working on MELCO data at encoding rate $R_{\mathrm{LDPC}} = 0.94$.

1. The locations of the minutiae points are quantized to reside in a 70×100 grid giving block-length $n = 7000$.
2. Because we only use a rate $R_{\mathrm{LDPC}} = 0.94$ code for these initial experiments, we eliminate all data sets for which all measurements have 30 or fewer, or 36 or more minutiae points, leaving 648 data sets. Our choice of encoding rate makes the former susceptible to the zero-probe attack and the latter have an unacceptably high error rate. By matching the encoding rate to the biometric entropy, this step can be avoided.
3. To get a sense of the pairwise noise between measurements we compute a score between any pair of observations. We first make a greedy match between minutiae points where the infinity-norm distance $\max\{|x_a - x_b|, |y_a - y_b|\}$ between a matched pair is not allowed to exceed 3. Unmatched enrollment minutiae are classified as erasures and unmatched probe minutiae as insertions. Each matched pair is assigned a score equal to its squared Euclidean distance. Erasures are assigned a score of 20 and spurious minutiae a score of 10.
4. For each data set we select the enrollment measurement as the measurement with the largest number of probes with scores less than 300. If no measurement met this criterion we eliminate that data set. We do this to model the fact that multiple measurements can be made at enrollment to ensure a representative biometric reading. This left 579 data sets that we use to calculate FRR and FAR results, reported in Table. 2.
5. We use the matching algorithm to calculate movement statistics. The mean and standard deviation of movement, erasures, and insertions ($\Pr[z_t = Ⓢ | y_t = 1]$) are plotted in Fig. 3.
6. The parameters of the statistical model used for decoding are set according to the empirically measured statistics. The prior on a minutiae moving a given distance is uniformly divided among all positions at that distance. We note also that the fingerprints in the MELCO database are not aligned. Before attempting to decode we first align the probe with the enrollment data. Since the overlap in the area of enrollment and probe fingerprints is only partial, we adjust the decoding model appropriately (setting $p_e = 1$ outside the overlap).

The results of our tests are given in Table 2. The first and second columns indicate the number of enrollment minutiae (and the corresponding source entropies) and the number of data sets at each enrollment parameter. The final four columns contain FRR and FAR results and the number of probes used to calculate them. While the enrollment files are limited as discussed above, all 1000 fingerprints are used to calculate the FARs. As is predicted by theory, FRR increases with enrollment entropy while FAR decreases. The zero-probe attack failed on all enrollment. An examination of the enrollment entropies reveals that our codes are not yet strong enough to get into the information theoretically secure region. This is our current focus.

4. CONCLUSIONS

We present a prototype secure biometrics system for fingerprints. The design is based on a model of minutiae movement and graphical codes. Our current focus is the refined design of LDPC codes, better matched to the asymmetric (and not memoryless) nature of the fingerprint channel. The codes we currently use are optimized for AWGN channels. Better codes will get into the information theoretically secure region and improve the FRR/FAR trade off.

5. REFERENCES

[1] G. I. Davida, Y. Frankel, and B. J. Matt, "On enabling secure applications through off-line biometric identification," in *IEEE Symp. Secur. Priv.*, Oakland, CA, May 1998, pp. 148–157.

[2] A. Juels and M. Sudan, "A fuzzy vault scheme," in *IEEE Int. Symp. Inform. Theory*, Lausanne, Switzerland, Jul 2002, p. 408.

[3] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcard-based fingerprint authentication," in *ACM SIGMM Work. Biom. Meth. Apps.*, Berkeley, CA, 2003, pp. 45–52.

[4] S. Yang and I. M. Verbauwhede, "Secure fuzzy vault based fingerprint verification system," in *Asilomar Conf.*, Monterey, CA, Nov 2004, pp. 577–581.

[5] U. Uludag and A.K. Jain, "Fuzzy fingerprint vault," in *Workhop: Biometrics: Challenges Arising from Theory to Practice*, Cambridge, UK, Aug 2004, pp. 13–16.

[6] R. Ahlswede and I. Csiszar, "Common randomness in information theory and cryptography I: Secret sharing," *IEEE Trans. Inform. Theory*, pp. 1121–1132, Jul 1993.

[7] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, pp. 471–480, Jul 1973.

[8] E. Martinian, S. Yekhanin, and J. S. Yedidia, "Secure biometrics via syndromes," in *Allerton Conf.*, Monticello, IL, Sep 2005.

[9] 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.

[10] F. R. Kschischang, B. J. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, pp. 498–519, Feb 2001.