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

Leveraging Reliable Bits: ECC Design Considerations For Practical Secure Biometric Systems

Yige Wang, Shantanu Rane, Anthony Vetro

TR2009-074 December 2009

Abstract

It is well-known that a biometric fuzzy vault can be constructed by applying an error correcting code (ECC) to a biometric signal. This is attractive because authentication only requires the check bits of the ECC to be stored on the access control device, whereas the personal biometric traits need not be stored. For a given coding rate, the ECC attempts to correct the errors between an enrollment biometric and the provided probe, and authenticates if it is successful in doing so. Unfortunately, most implementations of biometric fuzzy vaults have very poor robustness to the inherent noisiness of biometric measurements. In this paper, we provide ECC design considerations for secure biometric systems, which provide both better robustness and greater security. In particular, for any feature extraction algorithm, we propose to reorder the feature bits according to their reliability, and associate the reliable bits with high-degree variable nodes in the graph of the ECC. Further, the reliability of a bit is measured at enrollment and used to initialize the ECC decoding. Experiments on an extensive database show considerable reduction in the false reject rate, while restricting the successful attack rate to a very low value.

IEEE Workshop on Information Forensics and Security

LEVERAGING RELIABLE BITS: ECC DESIGN CONSIDERATIONS FOR PRACTICAL SECURE BIOMETRIC SYSTEMS

Yige Wang, Shantanu Rane and Anthony Vetro

Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, USA.

ABSTRACT

It is well-known that a biometric fuzzy vault can be constructed by applying an error correcting code (ECC) to a biometric signal. This is attractive because authentication only requires the check bits of the ECC to be stored on the access control device, whereas the personal biometric traits need not be stored. For a given coding rate, the ECC attempts to correct the errors between an enrollment biometric and the provided probe, and authenticates if it is successful in doing so. Unfortunately, most implementations of biometric fuzzy vaults have very poor robustness to the inherent noisiness of biometric measurements. In this paper, we provide ECC design considerations for secure biometric systems, which provide both better robustness and greater security. In particular, for any feature extraction algorithm, we propose to reorder the feature bits according to their reliability, and associate the reliable bits with high-degree variable nodes in the graph of the ECC. Further, the reliability of a bit is measured at enrollment and used to initialize the ECC decoding. Experiments on an extensive database show considerable reduction in the false reject rate, while restricting the successful attack rate to a very low value.

Index Terms— Biometrics, fuzzy vault, distributed source coding, LDPC codes, error correction coding

1. INTRODUCTION

Biometric access control is becoming increasingly popular as an alternative to traditional password-based authentication. This is primarily because biometrics authentication is convenient (does not involve remembering a password) and because a biometric signal is difficult to replicate. However, biometric access control presents new challenges of its own. Biometric measurements are inherently noisy, and an authentication system must be robust to variations among the biometric samples of a given user. Conventionally, this problem is solved by storing a reference biometric sample, or a feature vector obtained from the biometric, on the device. Then, some robust pattern-matching algorithm compares the reference sample with a probe and confirms or denies access. This creates a security threat, because anyone who gains unauthorized access to the device can steal the biometric sample. Since a user cannot generate an unlimited number of new biometrics, this is a serious problem.

In principle, this problem can be solved by using a "fuzzy vault" scheme [1, 2][1]. The user and the system agree on a secret key, and if there is sufficient "common randomness" between the enrollment and the probe biometrics, then the user can extract the key and thereby gain access to the system. In practice, this involves using an error correction code (ECC) in a Slepian-Wolf coding framework [3]. The ECC can correct the slight variations among noisy but legitimate measurements. Further, the check bits of the ECC emulate the cryptographic hash in traditional password systems. Just as a hacker cannot invert the hash and steal the password, he cannot just use the check bits to recover and steal the biometric.

The advantages notwithstanding, implementations based on this principle [4, 5, 6] suffer from high false reject rates (FRR). The main reason for this is that it is difficult to model the noisy channel between multiple biometric measurements from a given user. Therefore, it is difficult to design an ECC for this noisy channel. This problem was partially remedied in [7] in the context of fingerprint biometrics. The method adopted in that work was to transform the fingerprint into a feature vector which possesses some desirable properties. In particular, after feature transformation, the complicated biometric channel is reduced to a binary symmetric channel (BSC), for which standard ECC designs are readily available. Using LDPC codes in a Slepian-Wolf coding framework, this system achieves FRR = 11% and FAR = 0.01% and provides 30 bits of security.

To make secure biometric systems practical, the FRR-FAR tradeoff must be improved further. While the scheme of [7] provides a simple framework for implementing a secure biometric system, it does not fully exploit the fact that bits extracted from a biometric have different reliabilities [8]. In particular, the feature transformation generated reliable bits, but the ECC was agnostic to the difference in reliabilities. The hypothesis of this work is that exploiting the unequal reliabilities of the feature bits in ECC decoding can significantly improve the security vs. robustness (FAR vs. FRR) tradeoff. We retain the idea from [7] of transforming a complicated channel between biometric signals to a simple channel between the corresponding feature vectors. However, after feature transformation, we manipulate the decoding decisions of the ECC based on the reliability of those features.

The remainder of this paper is organized as follows: In Section 2, we present a general framework for secure biometrics, in which the system is divided into a feature transformation and a syndrome code. The desirable characteristics of the feature transformation are enumerated. In Section 3, we discuss ECC design considerations that enable the system to achieve an efficient FAR-FRR tradeoff. The method presented is general and is applicable to any biometric modality. For concreteness, in Section 4, we take the example of feature vectors extracted from fingerprints that possess the properties of Section 2, and demonstrate the improvement in the security-robustness tradeoff.

[1] Another way to address this problem is via "cancelable" biometrics, in which biometric features stored on the device can be revoked and different features can be assigned in the case of suspected attack. However, it is difficult to provide security guarantees for such systems especially if the cancelable feature transformation algorithm is compromised.

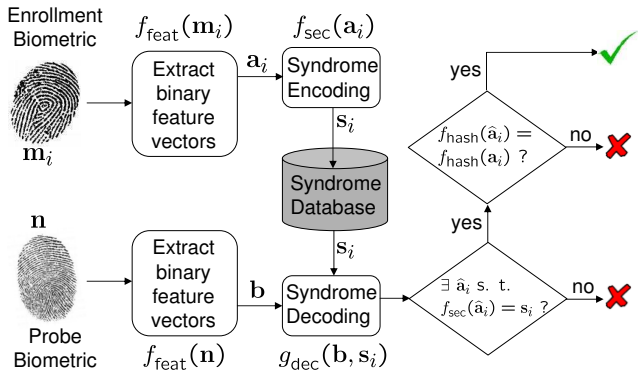

Fig. 1. A general framework for implementing and analyzing a secure biometric system.

2. SECURE BIOMETRICS ARCHITECTURE

A general framework for secure biometrics is shown in Fig. 1. This framework implements a biometric fuzzy vault in two stages. The first stage involves transforming the biometric into feature vectors, and the second stage involves Slepian-Wolf coding of the feature vectors. The central idea is to generate binary feature vectors which are i.i.d. Bernoulli(0.5), independent across different users but different measurements of the same user are related by a binary symmetric channel (BSC) with small crossover probability[2]. This is one of the standard channel models and therefore standard ECC designs such as LDPC codes can be used for Slepian-Wolf coding of the feature vectors. We emphasize that the feature transformation is made public and is *not* assumed to provide any security. Security is provided by the syndromes generated by the Slepian-Wolf coder.

2.1. Enrollment and Authentication Procedure

Suppose that there are M users. During enrollment, user $i \in \mathcal{I} = \{1, 2, ..., M\}$ provides a biometric $\mathbf{m}_i$. Then a feature transformation function $f_{\text{feat}}(\cdot)$ maps the biometric into a binary feature vector $\mathbf{a}_i = f_{\text{feat}}(\mathbf{m}_i)$ of fixed length N. Individual bits of $\mathbf{a}_i$ are denoted by $a_{i,j}$ with $j \in \mathcal{J} = \{1, 2, ..., N\}$. Next, a function $f_{\text{sec}}(\cdot)$ maps the binary vector into a secure biometric $\mathbf{s}_i = f_{\text{sec}}(\mathbf{a}_i)$. In the framework considered here, $f_{\text{sec}}(\cdot)$ does syndrome encoding using an error correcting code $\mathbb{C}$. The access control system stores $\mathbf{s}_i$, $\mathbb{C}$ and a cryptographic hash of the binary feature vector $f_{\text{hash}}(\mathbf{a}_i)$. It does not store either $\mathbf{m}_i$ or $\mathbf{a}_i$.

During authentication, the user i, or attacker impersonating user i, requests access by providing a probe $\mathbf{n}$. The access control system transforms $\mathbf{n}$ into a probe feature vector $\mathbf{b} = f_{\text{feat}}(\mathbf{n})$. Now, the ECC decoder assumes that the probe feature vector $\mathbf{b}$ is an error prone version of the enrollment feature vector $\mathbf{a}_i$. It combines the secure biometric $\mathbf{s}_i$ (syndrome) and the probe vector $\mathbf{b}$ and performs ECC decoding. In distributed source coding terminology, this is equivalent to Slepian-Wolf decoding of the syndrome $\mathbf{s}_i$ using $\mathbf{b}$ as side information. The result is either an estimate $\widehat{\mathbf{a}}_i$ of enrollment vector $\mathbf{a}_i$, or a special symbol $\emptyset$ indicating decoding failure. Now, it is possible that $\widehat{\mathbf{a}}_i \neq \mathbf{a}_i$, yet $\widehat{\mathbf{a}}_i$ satisfies the syndrome $\mathbf{s}_i$. To protect against this possibility, and more importantly to protect against an attacker using a stolen set of syndromes to construct his own estimate $\widehat{\mathbf{a}}_i$ which satisfies the syndromes but is not the true biometric, access is granted if and only if $f_{\text{hash}}(\widehat{\mathbf{a}}_i) = f_{\text{hash}}(\mathbf{a}_i)$.

[2]The statistical requirements on feature vectors in this section were first reported by some of the current authors in [7]. Here, the objective is to augment that framework so that the feature vectors can be properly matched to an error correcting code in the following section.

2.2. Desirable Statistical Properties of Feature Vectors

Based on the requirements mentioned at the beginning of this section, we propose a general secure biometric system in which the feature vectors possess the following properties:

1. A bit in a feature vector representation is equally likely to be 0 or 1. Thus, the entropy, $H(A_{i,j}) = 1$ bit for all $i \in \mathcal{I}$ and $j \in \mathcal{J}$. (Here $A_{i,j}$ denotes a random variable, and $a_{i,j}$ denotes the actual realization.)
2. Different bits in a given feature vector are independent, so that a given bit provides no information about any other bit. Thus, $H(A_{i,j}, A_{i,k}) = H(A_{i,j}) + H(A_{i,k}) = 2$ bits for all $j \neq k$ where $j, k \in \mathcal{J}$.
3. Feature vectors $\mathbf{A}_i$ and $\mathbf{A}_u$ from different users are independent, so that one user's feature vector provides no information about another user's feature vector. Thus, $H(A_{i,j}, A_{u,k}) = H(A_{i,j}) + H(A_{u,k}) = 2$ bits for all $i, u \in \mathcal{I}$, $i \neq u$ and all $j, k \in \mathcal{J}$.
4. Suppose feature vectors $\mathbf{A}_i$ and $\mathbf{A}'_i$ are obtained from different readings of the same biometric. Then bits $A_{i,j}$ and $A'_{i,j}$ are statistically related by a BSC whose crossover probability is denoted by $p_{i,j}$. Thus, $H(A'_{i,j}|A_{i,j}) = H(p_{i,j})$ for all $i \in \mathcal{I}$ and $j \in \mathcal{J}$. If $p_{i,j}$ is small, it means that the bit $A_{i,j}$ is robust to repeated noisy measurements.

It has been shown in [9] that this secure biometric framework has positive information theoretic security. In other words, given the syndrome $\mathbf{s}_i$, $H(\mathbf{A}_i|\mathbf{s}_i) > 0$. For this work, we are concerned with practical implementation using error correcting codes. For an ECC with rate $R, 0 < R < 1$, the syndrome stored on the device consists of $(1 - R)N$ bits. The coding rate R determines a performance tradeoff for the access control system, which is discussed in the following section.

3. ECC DESIGN CONSIDERATIONS

To optimize the security robustness tradeoff, we propose to exploit the reliability of feature vector bits and assign the feature bits to appropriate codeword bits of an ECC. For secure biometrics using the framework of Fig. 1, measures of security and robustness are defined in brief as follows:

1. The False Reject Rate (FRR) is the probability with which ECC decoding fails to recover an enrollment feature vector given the stored syndrome and a legitimate probe feature vector.
2. The False Accept Rate (FAR) is the probability with which ECC decoding recovers an enrollment feature vector given the stored syndrome and an illegitimate probe feature vector.
3. The Successful Attack Rate (SAR) is the probability with which ECC decoding recovers an enrollment feature vector given the stored syndrome and an illegitimate probe feature vector constructed by an attacker using some extra side information about the feature extraction process. For instance, the attacker may know which transforms applied to the biometric signal can produce reliable bits for the victim. This is a more realistic measure of security than FAR.

4. Number of Bits of Security (NBS) measures the secrecy offered by the secure biometric, i.e., the syndrome. It is defined as the number of bits that an attacker must guess correctly in order to extract a feature vector, given the syndrome and the ECC parameters. If the feature vector bits satisfy the properties of Section 2, then NBS $= N - (1 - R)N = RN$.

There is a natural tradeoff among these measures. For example, to make NBS large, one must perform ECC with a large coding rate, thereby generating a small number of syndrome bits. This will increase the likelihood that a noisy but legitimate biometric probe can not be decoded to the enrollment feature vector, thereby increasing the FRR. In any case, given a coding rate R, the best tradeoff between FRR and FAR (or SAR) will be achieved by a channel code that most closely approaches channel capacity. By design (Property 4, in Section 2), each bit of a legitimate probe feature vector is related to the corresponding bit of the enrollment feature vector by a BSC. The reliability of a feature vector bit can be measured in terms of the crossover probability of the corresponding BSC. Let the crossover probability be p and the reliability be $\mathcal{R}$, then

$$\mathcal{R} = \left| \log \left(\frac{1-p}{p} \right) \right|. \tag{1}$$

It is obvious that the larger $\mathcal{R}$ is, the more likely it is that the bit has not flipped between measurements. In the sequel, we apply low-density parity check (LDPC) codes to demonstrate how this reliability information can be combined with coding so as to optimize the security-robustness tradeoff.

3.1. Properties of LDPC codes

An LDPC code is often represented by a bipartite graph with two types of nodes, variable nodes that correspond to codeword bits, and check nodes that correspond to parity-check constraints. The number of check nodes that a variable node connects to is called the *degree* of that variable node.

In general, LDPC codes can be categorized into regular LDPC codes and irregular LDPC codes. Regular LDPC codes are those for which all nodes of the same type have the same degree, while irregular LDPC codes have non-constant degrees for variable and check nodes. In [10], it has been shown that irregular LDPC codes can approach Shannon capacity with iterative decoding. Therefore, we use an irregular LDPC code to generate the syndrome. Then, access control involves Slepian-Wolf decoding of the syndrome in the presence of the probe feature vector as side information. Decoding is performed iteratively using belief propagation (BP). When BP decoding is used for irregular LDPC codes, high-degree variable nodes obtain more information from their connected check nodes. Consequently, the bits in these nodes can be decoded more accurately [11]. We exploit this property in our code design, as detailed below.

3.2. Associating Reliable Bits with LDPC Codegraphs

There are numerous examples in the literature in which bits extracted from human biometric signals have different reliabilities. For instance, when bits are extracted from fingerprint minutiae, the reliability of the extracted bit depends on the location of the minutiae point. It stands to reason that, in order to make an accurate access control decision, it is necessary for the decoding algorithm to exploit the reliability information to the fullest extent possible. This applies to conventional biometric matching as well as to the proposed secure biometrics framework.

The probability $p_{i,j}$ in Property 4 of Section 2.2 can be estimated during the enrollment stage when multiple biometric samples, and hence multiple feature vectors, are extracted from each user. After $p_{i,j}$ is measured, we propose to re-order the bits in the feature vector $\mathbf{a}_i$ in the order of increasing $p_{i,j}$, i.e., in the order of decreasing reliability. By abuse of notation, we denote the re-ordered feature vector by the same symbol $\mathbf{a}_i$. After re-ordering, $j < k \Rightarrow p_{i,j} \leq p_{i,k}$ for all $j, k \in \mathcal{J}$. This reordering is performed for each enrolled user. The corresponding reliability $\mathcal{R}_{i,j}$ can be calculated for all $i \in \mathcal{I}$ and $j \in \mathcal{J}$ according to (1). The reordered reliabilities then have the property that $\mathcal{R}_{i,j} \geq \mathcal{R}_{i,k}$ for $j < k$.

Next, the reordered feature vector bits are associated with variable nodes of the chosen LDPC code graph such that highest-reliability bits are placed at the highest-degree variable nodes. The advantage of this is that reliable information can be spread out more quickly during the message-passing iterations of BP. After the mapping between feature vector bits and variable nodes is decided, we may permute variable nodes so that their indices agree with their corresponding feature vector bit indices. This permutation does not change either the code or its error performance.

3.3. Soft Initialization of LDPC Decoding

Following the above association of high-reliability bits with high-degree variable nodes, the coding performance can be improved further if the soft-decision decoder knows the reliability of each bit of side information. At each iteration of the BP algorithm, the messages exchanged between variable nodes and check nodes are often represented by Log Likelihood Ratio (LLR). Since different bits of the feature vector have different reliability, each variable node should have its own initial LLR at the start of the decoding process. Denote the initial LLR of the j^{th} variable node of user i by $L_{i,j}$ for $i \in \mathcal{I}$ and $j \in \mathcal{J}$. Now, the j^{th} bit in the probe feature vector of user i is the output of a BSC with crossover probability $p_{i,j}$ and reliability $\mathcal{R}_{i,j}$ which have both been estimated at the time of enrollment.

When a probe feature vector $\mathbf{b}$ is provided for authentication, the initial LLR for the j^{th} variable node can be obtained by

$$L_{i,j} = \begin{cases} \mathcal{R}_{i,j} & \text{if } b_j = 0 \\ -\mathcal{R}_{i,j} & \text{if } b_j = 1 \end{cases}$$

where b_j is the probe feature vector bit which is mapped to variable node j. This is repeated for all $j \in \mathcal{J}$.

For secure biometric authentication of M users with N-bit feature vectors, the above method would require the storage of MN reliabilities at the access control device, in addition to the M syndromes. As a more practical alternative, we propose to store only N reliabilities as follows: After reordering the feature vectors as in Section 3.2, the average crossover probability for each bit position across all M users is computed using $\bar{p}_j = \frac{1}{M} \sum_{i=1}^{M} p_{i,j}$ for all $j \in \mathcal{J}$. Then, the reliabilities $\bar{\mathcal{R}}_j$ corresponding to these average crossover probabilities are obtained by substituting $p = \bar{p}_j$ in (1). Finally, store these N reliabilities $\bar{\mathcal{R}}_j$, $j \in \mathcal{J}$ on the access control device. With these stored reliabilities, BP decoding for any probe feature vector will start by initializing the LLRs at the variable nodes using

$$\bar{L}_j = \begin{cases} \bar{\mathcal{R}}_j & \text{if } b_j = 0 \\ -\bar{\mathcal{R}}_j & \text{if } b_j = 1 \end{cases}$$

where b_j is the probe feature vector bit provided by the user, or attacker. Note that, the sign of the LLR is determined *at decoding time* and depends on the feature vector provided for authentication. In other words, storing $\bar{\mathcal{R}}_j$ on the access control device does not

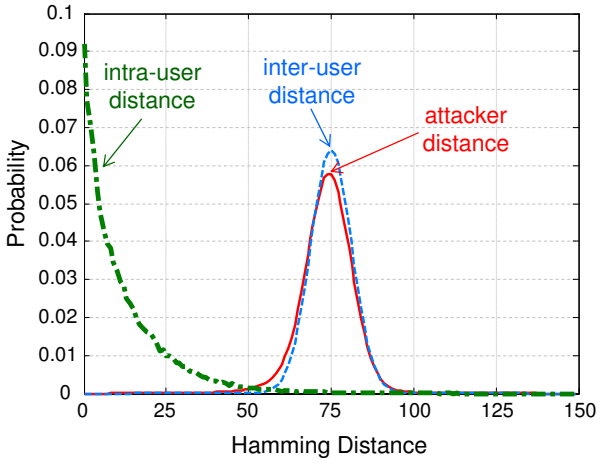

Fig. 2. Distibution of intra-user and inter-user pairwise distances. The attacker distribution is slightly shifted towards the left because the attacker knows some side information about the victim's feature vector.

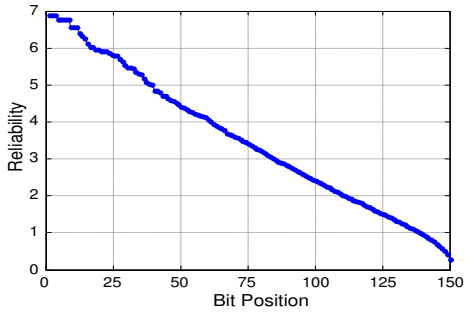

Fig. 3. The magnitude of the log likelihood ratio of the average crossover probabilities for each of the 150 bits is stored on the access control device and is used to initialize belief propagation decoding.

leak any information about whether a bit is more likely to be a 0 or 1. Even if the attacker knows the reliabilities, he still needs to guess enough positions correctly in order to successfully recover a victim's enrollment feature vector. Actually, the reliability information can improve security to some extent because, if the attacker provides wrong bits in the reliable positions, the decoding is more likely to fail.

3.4. Effect of Shuffled Belief Propagation

In standard BP, during each message passing iteration, all variable nodes or all check nodes are processed in parallel, while in shuffled BP [12, 13], they are processed serially. Therefore, nodes that are processed later can utilize the latest updated information from previously processed nodes. Compared with standard BP, the shuffled BP algorithm can reduce the number of iterations to achieve the same performance. Alternatively, by using the same number of iterations, the decoding performance is usually improved.

4. RESULTS

In this section, we present the results of experiments carried out to test the effectiveness of the proposed ECC design changes. We used an extensive (proprietary) database of 1035 users with 15 fingerprint samples per user. The fingerprint samples are converted into minutia maps; the average number of minutaie points in a sample is 32. There is a coarse alignment operation performed in the beginning in which a user's fingerprints are aligned with respect to one of the 15 available samples. This experimental setup and feature extraction algorithm are adoped from [7]. Since we are concerned with the effect of ECC design, we do not repeat the details of the feature extraction algorithm. In brief, the feature bits are extracted as follows: Random cuboids are generated in the minutiae space, and the minutiae falling inside each cuboid are counted. This number is compared with the median of the number of points in the cuboid measured for all users in the database, and a 0 or 1 bit is generated depending on whether the number is less than or greater than this median value. 400 cuboids are generated in all, and 150 of the most robust cuboids are preferred. Here, robustness refers to the fact that the number of minutiae points in the cuboid is far away from the median, resulting in a bit value that is preserved in repeated measurements. Clearly, different users have different robust cuboids.

With our ECC design considerations, we further reorder the cuboids using another reliability measure, namely the crossover probability $p_{i,j}, i \in \{1, 2, ..., 1035\}, j \in \{1, 2, ..., 150\}$. These cuboids are stored in their new order on the access control device. Fig. 2 plots the distribution of the Hamming distance between the feature vectors. It is observed that there is a very small overlap between the inter-user and intra-user distributions. These distributions assume that every user employs his own set of cuboids. Also plotted is an attacker distribution in which a user (attacker) decides to use the robust ordered cuboids of another user (victim). In this case, the distribution shifts slightly closer to the intra-user distribution, but the overlap between the two is still reasonably small.

As described in Section 3.3, the reliabilities corresponding to the average crossover probabilities are stored on the device in order from maximum to minimum. This is plotted in Fig. 3. For any user, these reliabilities are used to initialize BP decoding. The ECC used is an irregular LDPC code with 150 variable nodes. We test at various coding rates $R = 0.2, 0.25, 0.3, 0.35$ which will determine the number of syndrome bits. The corresponding code graphs have been obtained from [14]. To see the effect of ordering the bits according to reliability, soft initialization of LDPC decoding, shuffled belief propagation, access control is implemented for various scenarios, as shown in Table 1. The LDPC code rate used for all simulations in the table is $R = 0.2$, thus NBS$= 30$. It is observed that there is a slight reduction in the FRR when the reliabilities defined in Section 3.3 are used to initialize BP. There is a further dramatic reduction in the FRR when, in addition to the reliability initialization, the reliable feature bits are paired with the high-degree variable nodes. If the ECC is agnostic to the reliability ordering, the FRR is 11% whereas, with the proposed modifications, it drops to 3.3% with shuffled BP. The corresponding FAR is very small in all experiments. As expected, the SAR is greater than the FAR because the attacker additionally knows the new "reliability ordering" of the feature bits. Still, the SAR is less than 0.06% in all the simulations. Having obtained considerable reduction in FRR, at very low FAR, it is now possible to gradually trade off the increased robustness against the number of bits of security. If the system designer is dissatisfied with 30 bits of security, he can use larger coding rates to achieve greater secrecy. This tradeoff is plotted in Fig. 4 for the ECC decoding scheme with the best perfomance, i.e., shuffled BP decoding with soft initialization of LLRs and reliable bits in high-degree variable nodes. It is

Scheme	FRR	FAR	SAR
Unordered feature bits Equal initial LLR Standard BP decoding	0.11	1.19×10^{-4}	Not Applicable since equal initial LLRs
Unordered feature bits Unequal initial LLR Standard BP decoding	0.099	2.15×10^{-6}	4.37×10^{-4}
Unordered feature bits Unequal initial LLR Shuffled BP decoding	0.083	3.36×10^{-6}	5.00×10^{-4}
Reliability-ordered bits Unequal initial LLR Standard BP decoding	0.037	1.01×10^{-6}	4.30×10^{-4}
Reliability-ordered bits Unequal initial LLR Shuffled BP decoding	0.033	1.61×10^{-6}	5.41×10^{-4}

Table 1. The security-robustness tradeoff improves slightly when the initial LLRs for LDPC decoding are based on the reliabilities of the bits. It improves considerably when the reliable bits are associated with high-degree variable nodes. Using shuffled BP reduces FRR at the expense of a slight increase in FAR and SAR.

observed that by increasing the coding rate from 0.2 to 0.35, the number of bits of security increases from 30 to 53, while the FRR increases from 3.3% to 7%.

5. CONCLUSIONS

This paper proposed some ECC design considerations in secure biometrics. Specifically, the feature bits extracted from a human biometric are reordered in the order of reliability and then paired with appropriate variable nodes in the ECC code graph. Additionally, the ECC decoder is made aware of the average reliabilities in each variable node. It was shown using results on an extensive fingerprint database that the proposed changes reduce the FRR from 11% to 3.3% while maintaining a very low FAR. As a result of the proposed changes, the amount of data stored on the access control device increases very slightly. Further, the considerable reduction in FRR affords the opportunity to tradeoff the robustness for increased number of bits of security. It is possible that there are entirely new kinds of attacks whose effectiveness might not be captured by the error metrics analyzed herein. Our current work is directed at analyzing such attacks.

6. REFERENCES

[1] G. Davida, Y. Frankel, and B. Matt, "On Enabling Secure Applications through Off-line Biometric Identification," in *IEEE Symp. on Security and Privacy*, 1998, pp. 148–157.

[2] Ari Juels and Madhu Sudan, "A Fuzzy Vault Scheme," in *IEEE Intl. Symp. on Information Theory*, 2002.

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

[4] T. Charles Clancy, Negar Kiyavash, and Dennis J. Lin, "Secure Smartcard-based Fingerprint Authentication," in *ACM SIGMM workshop on biometrics methods and applications*, 2003.

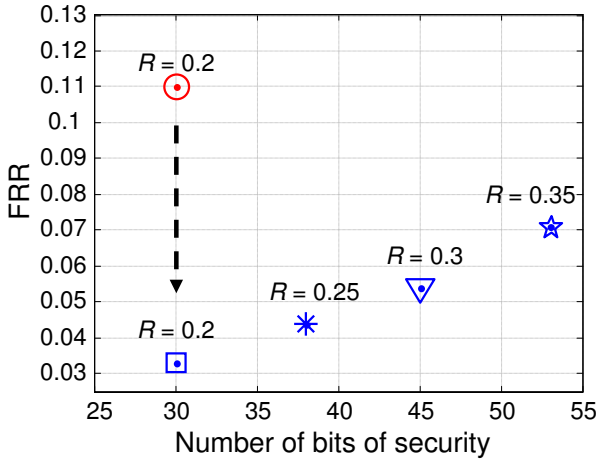

Fig. 4. With improved ECC design, the FRR is considerably lowered. The system designer can now consider increasing the rate of the channel code, trading off the FRR for more bits of security.

[5] Shenglin Yang and Ingrid Verbauwhede, "Automatic Secure Fingerprint Verification System based on Fuzzy Vault Scheme," in *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2005, pp. 609–612.

[6] U. Uludag and A.K. Jain, "Fuzzy Fingerprint Vault," in *Workshop on Biometrics: Challenges Arising from Theory to Practice*, Aug. 2004, pp. 13–16.

[7] J. S. Yedidia S. C. Draper Y. Sutcu, S. Rane and A. Vetro, "Feature extraction for a slepian-wolf biometric system using ldpc codes," in *IEEE Int. Symp. Inform. Theory*, July 2008, pp. 2297–2301.

[8] P. Tuyls, A.H.M. Akkermans, T.A.M. Kevenaar, G.J. Schrijen, A.M. Bazen, and R.N.J. Veldhuis, "Practical biometric authentication with template protection," in *International Conference on Audio and Video-based Biometric Personal Authenticaion*, jul.

[9] S. Rane A. Vetro, S. C. Draper and J. S. Yedidia, "Securing biometric data," *Distributed Source Coding*, vol. 47, pp. 293–324, Feb. 2009.

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

[11] M. A. Shokrollahi M. G. Luby, M. Mitzenmacher and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Information Theory*, vol. 47, no. 2, pp. 585–598, Feb. 2001.

[12] J. Zhang and M. Fossorier, "Shuffled iterative decoding," *IEEE Transactions on Communications*, vol. 53, no. 2, pp. 209–213, 2005.

[13] H. Kfir and I. Kanter, "Parallel versus sequential updating for belief propagation decoding," *Physica A: Statistical Mechanics and its Applications*, vol. 330, pp. 259–270, 2003.

[14] Information Processing Group, EPFL, Lausanne, Switzerland., "LdpcOpt Website," http://ipgdemos.epfl.ch/ldpcopt/.