On fuzzy syndrome hashing with LDPC coding

Marco Baldi, Marco Bianchi and Franco Chiaraluce
Università Politecnica delle Marche
Ancona, Italy
Email: {m.baldi, m.bianchi, f.chiaraluce}@univpm.it

Joachim Rosenthal and Davide Schipani
University of Zurich
Zurich, Switzerland
Email: {rosenthal, davide.schipani}@math.uzh.ch

***Abstract*—The last decades have seen a growing interest in hash functions that allow some sort of tolerance, e.g. for the purpose of biometric authentication. Among these, the syndrome fuzzy hashing construction allows to securely store biometric data and to perform user authentication without the need of sharing any secret key. This paper analyzes this model, showing that it offers a suitable protection against information leakage and several advantages with respect to similar solutions, such as the fuzzy commitment scheme. Furthermore, the design and characterization of LDPC codes to be used for this purpose is addressed.**

I. Introduction

The use of biometric passwords, such as fingerprints, irises, etc, has been an important issue in recent years, both because of the big advantages it may bring along and because of the clearly non negligible privacy concerns and implementation issues [1]. In fact, as far as privacy is concerned, the storage of biometric data in clear is not an acceptable solution, but, on the other hand, a secure storage cannot be easily implemented as for traditional passwords by simply introducing an hash function. This is due to the fact that the binary strings derived from different acquisitions of the same biometric feature can slightly change from each other, and the biometric feature can slightly change itself. Therefore a certain threshold of tolerance is needed to be able to identify legitimate from non legitimate users, but this prevents the standard use of collision resistant hash functions [2].

This problem has prompted researchers to devise other ways to still be able to store and use securely biometric passwords: while a selected survey of the literature can be found in [3] and the references therein, we limit ourselves to point out some of the most significant milestones. The idea behind most of the methods that have been proposed is a combined use of error correcting codes and hash functions, whose model is the fuzzy commitment scheme [4], which we revisit below. This has been later generalized to other types of metrics, such as the set difference metric [5] and the edit distance metric [6].

In particular, the fuzzy vault [5] uses polynomial interpolation in order to allow authentication based on the matching of a sufficient number of features, while the fuzzy extractor [6] is a further generalization which combines the previous constructions with particular objects called random extractors. These make the previous schemes stronger with respect to information leakage, though they cannot prevent it [7], [8].

Briefly, privacy and implementation issues are still a concern and our aim is to give a further contribution, by showing that a syndrome based fuzzy hashing is actually feasible and convenient as far as information leakage is concerned. First of all, we address the design of codes to be used within this context. We focus on LDPC codes and provide both theoretical and numerical results concerning their design. Secondly, we study the entropy of the fuzzy hashing output vectors and compare it with the corresponding quantity of its input vectors.

The paper is structured as follows: in Section II we briefly review the prototype of these methods, that is the fuzzy commitment scheme, and the main issues concerning it. Section III is devoted to the syndrome based construction which we denote by fuzzy hashing. Section IV addresses the design of codes to be used in this scheme, through both theoretical and numerical tools. Section V is aimed at analyzing the entropy of the output vectors, in order to assess the performance of fuzzy hashing in terms of privacy.

II. Prior art: the prototype

We briefly review here the prototype of most error correcting based methods, that is the fuzzy commitment scheme proposed by Juels and Wattenberg [4].

Suppose that we want to securely store a biometric vector $x \in \mathbb{F}_q^n$, where $\mathbb{F}_q$ is the Galois field of order q, having length n and let e be the maximum number of different symbols with respect to the reference vector x that we can tolerate in any other acquisition of the same biometric feature.

According to the fuzzy commitment scheme, we must choose a hash function H_a and a $[n, k]$-linear block code $C \subset \mathbb{F}_q^n$, able to correct e errors, and then store $(H_a(r_x), l)$, where r_x is a random codeword associated to x and $l = x - r_x$.

Now, given another biometric y, we compute vector $z = y - l$ and apply the decoding algorithm according to code C. If decoding succeeds, this results in a codeword $c_z \in C$, and we compute $H_a(c_z)$. If $H_a(c_z)$ equals $H_a(r_x)$, the value previously stored, we grant access, otherwise we deny it.

In fact, if the hashes are the same, then $c_z = r_x$ (apart from a negligible probability of a hash collision), hence $d(c_z, z) = d(r_x, z) \leq e$, where $d(\cdot)$ denotes the Hamming distance. Since $r_x = x - l$ and $z = y - l$, it results $d(r_x, z) = d(x, y) \leq e$.

Conversely, $d(x, y) \leq e$ implies $d(x - l, y - l) = d(r_x, z) \leq e$, so that decoding z results in $c_z = r_x$ and $H_a(c_z) = H_a(r_x)$.

In [2], we pointed out some of the main problems concerning the use of this scheme, namely implementation issues and security issues. In particular, privacy concerns may arise if the biometric templates are not uniformly distributed in the ambient space, that is their entropy is not maximal. In that case, it may be feasible to infer from l information about r_x and, therefore, endanger the security of the system.

III. Syndrome fuzzy hashing

Starting from the fuzzy commitment principle, an alternative scheme (here fuzzy hashing) can be devised, in which syndromes are used in the place of codewords. Under a coding theory viewpoint, the two schemes are equivalent. Despite this, the use of syndromes has several advantages in the considered context. The idea of storing the syndrome of x, instead of a shift vector from a codeword, already appeared in [6], where it is considered as an example of a sketch construction. We will show that the use of fuzzy hashing is advantageous with respect to the classical fuzzy commitment scheme, also by considering the characteristics of typical biometric data.

In the fuzzy hashing scheme, an $[n, k]$-linear block code $C \subset \mathbb{F}_q^n$, able to correct e errors, is selected, and it is described through its $r \times n$ parity-check matrix H, with $r = n - k$. Given a biometric vector x to be stored, the pair $(H_a(x), Hx)$ is used to represent x, were H_a is a given hash function. When another biometric y is acquired and is compared with x, the value $Hx - Hy = H(x - y) = Hv$ is computed, that coincides with the syndrome associated to the difference vector $v = x - y$.

Then, syndrome decoding is applied on Hv, according to the chosen code C. If y is taken from the same individual as x, then v has Hamming weight equal to $d(x, y) \leq e$ and it corresponds to a correctable error vector. So, syndrome decoding succeeds and correctly results in v. Then, starting from v and y, x can be computed, as well as $H_a(x)$. The latter coincides with the stored one, so access is granted. Otherwise, syndrome decoding fails or reports a vector $w \neq v$. In such case, $x' = w + y \neq x$ is computed and $H_a(x')$ obtained. This does not coincide with the stored value, so access is denied.

In the fuzzy commitment, the vector $l = x - r_x$ is stored. As some bits of the biometric x might be known with high probability, this reveals some information on the secret codeword r_x. The same may occur in fuzzy hashing, where the syndrome Hx is stored, but only under the condition $x = q + r_q$, where q is a correctable error vector and r_q is any codeword. In this case, syndrome decoding results in q; so, some bits of r_q can still be guessed, starting from the predictable bits of x. However, especially for very low rate codes, the probability that x is within the decoding radius of a codeword r_q is very low, so fuzzy hashing provides better security with respect to the classical fuzzy commitment.

IV. Codes for fuzzy hashing

In order to design suitable codes to be included in the fuzzy hashing scheme, we must consider the features of the biometric vectors we work with. If we refer to fingerprints or irises, a common acquisition will consist of a vector of several thousands of bits. However, it would be unpractical to apply fuzzy hashing directly on the plain acquisition, since a number of impairments could jeopardize the identification process. In fact, small changes in the acquisition conditions (as ambient light or small movements of the subject) could result in significant differences between two images of the same biometric feature. So, it is a common procedure to extract a set of representative features from the biometric data through algorithms aimed at making them invariant to some frequent acquisition impairments. An example of this kind of algorithms will be considered in Section V.

Based on these premises, the code must be designed to work well with the vectors produced as output by the feature extracting algorithm. Typically, such vectors have length on the order of 10k bits.

Another important aspect is the modeling of the errors affecting two vectors resulting from different biometric acquisitions from the same individual. In [9], the errors are modeled through a Binary Symmetric Channel (BSC) with transition probability p. We will adopt this approach in this paper.

In order to estimate the value of p that is best suited to model this application, we can assume that typical feature extraction algorithms are used in the process. As it will be shown in the next section, typical values of the percentage of different bits between the vectors representing two acquisitions of the same biometric range between 10% and 30%. So, codes able to correct such high fractions of errors are needed, that is, we need long codes having very low rate (R). Just to give an idea, a BCH code with ($n = 2047$, $k = 100$), that is, rate $R \approx 0.05$, is able to correct 379 errors, i.e. about 19%. A BCH code with ($n = 4095$, $k = 110$), that is, rate $R \approx 0.03$, is able to correct 767 errors, that is almost the same percentage. A similar value is reached by the BCH code having ($n = 8191$, $k = 170$), hence rate $R \approx 0.02$, able to correct 1533 errors.

This evidences that, for classical algebraic codes, in order to maintain a given relative error correcting capability, the code rate must be decreased as the code length increases. Furthermore, since very long codes are needed to match the length of biometric vectors, their decoding may also yield complexity issues, although recent algorithms can improve the decoding complexity [10]. A smarter choice is represented by modern iteratively decoded error correcting codes, characterized by capacity achieving performance. Within such class of codes, the choice of Low-Density Parity-Check (LDPC) codes [11] seems the natural one. Actually, the use of LDPC codes in this context has already been proposed in [2], [9], [12], but the code design was not addressed in those works. In summary, fuzzy hashing with LDPC codes brings the following advantages:

- Fuzzy hashing allows to reduce the amount of stored data, with respect to the fuzzy commitment, since $r < n$.
- Fuzzy hashing allows to reduce the predictability of the stored strings, as shown at the end of the previous section.
- LDPC codes have higher error correction capabilities with respect to classical algebraic codes. Moreover, the relative error correction performance of LDPC codes, for a fixed rate, remains almost constant as the code length increases.
- LDPC codes allow to reduce the size of the code representation, by exploiting the sparse nature of H.

A. Code Design

We are interested in almost regular codes, since they allow an easier implementation with respect to irregular ones; so, we fix the column weight of the matrix H to be equal to an integer d_v. The row weight, for the code rate values here of interest, cannot be constant as well. However, it will be minimally dispersed around its mean $\langle d_c \rangle = d_v/(1-R)$. If we suppose that (as it occurs for all the code examples we consider):

$$\frac{k}{n} = R < \frac{1}{d_v+1}, \tag{1}$$

the matrix H can have rows with only the following two values of weight: d_v and d_v+1. In this case, $r - k \cdot d_v$ rows have weight d_v and the remaining $k \cdot d_v$ rows have weight d_v+1.

We can describe the column and row weight distributions of the matrix H through the polynomials $\lambda(x)$ and $\rho(x)$, representing respectively the variable node and check node degree distributions of the associated Tanner graph [11]. Since we adopt the edge perspective, λ_i (ρ_i) denotes the fraction of ones in the parity-check matrix H which are in columns (rows) of weight i. Based on the hypotheses above, for the considered ensemble of codes it results:

$$\begin{aligned} \lambda(x) &= x^{d_v-1}, \\ \rho(x) &= [1 - R(1+d_v)]x^{d_v-1} + R(1+d_v)x^{d_v}. \end{aligned} \tag{2}$$

Starting from (2), we can estimate the asymptotic performance (that is, for $n \to \infty$) of LDPC codes in the considered ensemble, over the BSC, by applying the density evolution method [11].

In particular, as an example of LDPC decoding algorithm for the BSC channel, we can refer to Gallager's A algorithm [13] and estimate its convergence threshold through density evolution [14]. By doing so, and considering the variable and check node degree polynomials given by (2), we have obtained the results reported in Table I, where the threshold values computed for $d_v = 3, 4, 5$ are provided, for code rates ranging between 0.1 and 0.01.

TABLE I
THRESHOLD VALUES FOR THE CONSIDERED LDPC CODES ENSEMBLES UNDER GALLAGER'S A DECODING

R	$d_v = 3$	$d_v = 4$	$d_v = 5$
0.1	0.159	0.078	0.045
0.09	0.163	0.079	0.045
0.08	0.166	0.08	0.046
0.07	0.169	0.081	0.046
0.06	0.173	0.083	0.047
0.05	0.177	0.084	0.048
0.04	0.18	0.085	0.048
0.03	0.184	0.087	0.049
0.02	0.188	0.088	0.05
0.01	0.192	0.089	0.051

As we observe from the table, the choice of a small value of d_v (like 3) should be preferred. On the other hand, the asymptotic performance under Gallager's A decoding does not appear to be very good. For example, in order to reach an error correcting capability of 19%, for $d_v = 3$, a code rate as low as 0.02 seems to be needed, that is similar to what required by a BCH code with length 8191.

However, despite making reference to Gallager's A algorithm provides a first tool for designing the codes, we must consider that such decoding algorithm is not among the most effective ones. In fact, Gallager's A algorithm is a majority-based algorithm exploiting a fixed decision threshold b. For example, adopting a variable b (as in Gallager's B algorithm) gives a first performance improvement. Furthermore, several improved versions of these algorithms have been proposed in the literature, that are able to outperform Gallager's original algorithms [15], [16], [17]. Finally, the classical Sum Product Algorithm (SPA) [18], based on the Belief Propagation principle, can also be applied on the BSC, even though, in absence of soft-information from the channel, the initial likelihood associated to each bit can assume only two opposite values.

Despite this, the SPA is able to significantly improve the error correction performance with respect to that predicted in Table I, as we will show in the next section, by providing some examples of practical codes.

B. Examples

In this subsection we consider some examples of LDPC codes having parameters of interest in the fuzzy hashing context. For this purpose, we designed the codes, through the Progressive Edge Growth (PEG) algorithm [19], by imposing an almost constant column and row weight distribution of their parity-check matrices.

In detail, we first fixed the column weight d_v. Then, we imposed the lower triangular form for the parity-check matrices, in such a way as to facilitate encoding, especially for very long codes. This introduces a last column having weight 1, and some columns having weight $< d_v$. However, their incidence with respect to the total number of columns is very small. Then, the PEG algorithm has been used to optimize the length of the local cycles within the Tanner graph associated to each code, while keeping the row weight distribution as much concentrated as possible.

So, the characteristics of the codes we have designed are well overlaid with those fixed in the previous subsection. The codes mentioned above have been used to perform Montecarlo simulations over the BSC, based on SPA decoding.

A first set of results is reported in Fig. 1, where LDPC codes having $n = 9600$ and $k = 1000$ (hence, rate ≈ 0.1) have been considered. We have designed two codes with different column weight: $d_v = 3$ and $d_v = 5$. As we observe from the figure, the simulation confirms that the code with $d_v = 3$ has better performance, in the waterfall region, with respect to the code having $d_v = 5$. This was expected on the basis of the results of density evolution. However, we also observe that the code with $d_v = 5$ has a better performance in the error floor region, so its Bit Error Rate (BER) and Frame Error Rate (FER) curves tend to intersect with those of the first code. So, the choice of $d_v = 3$ is suitable if a failure rate on the order of 10^{-4} or

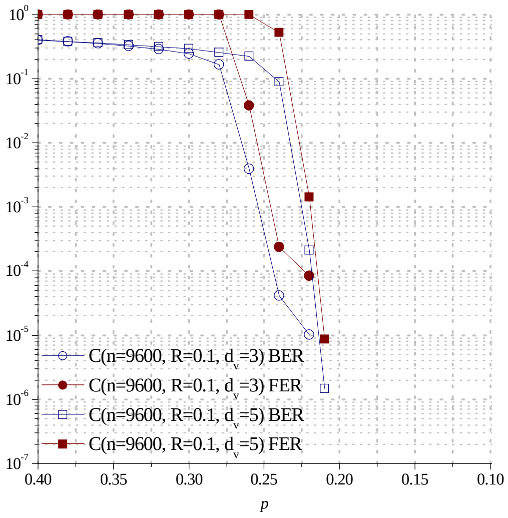

Fig. 1. Performance of rate 0.1 LDPC codes with $d_v = 3$ and $d_v = 5$.

more is acceptable; otherwise, the choice of $d_v = 5$ should be preferred.

The performance improvement due to the SPA is evident: both codes are able to achieve a rather low error rate for a percentage of bit errors around 20%, or even more.

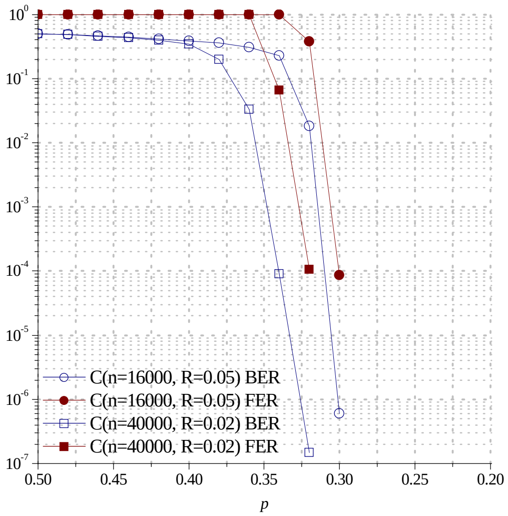

Fig. 2. Performance of LDPC codes with $d_v = 3$ and rate 0.05, 0.02.

In order to further increase the error correcting capability of these codes, it is necessary to reduce their rate. To provide some examples in this sense, we have considered $k = 800$ and designed two further LDPC codes, having $d_v = 3$ and rate 0.05 and 0.02 (that is, $n = 16000$ and $n = 40000$), respectively.

As we observe from their simulated performance, reported in Fig. 2, by using the SPA these codes are able to reach very low error rates for a percentage of bit errors around 30% and even more. Also in this case, the performance improvement due to the SPA with respect to the theoretical performance referred to Gallager's A algorithm is evident.

These results confirm that LDPC codes are well suited for the application in the considered context, in which a high correction capability is needed. Furthermore, we can observe that, in this study, we have limited ourselves to consider almost regular codes, in order to keep their implementation complexity low. However the adoption of irregular LDPC codes can result in a further performance improvement.

V. Entropy analysis

In this section, we discuss the use of fuzzy hashing for iris recognition and we study how the adoption of syndromes affects some statistical properties of the biometric data. As a feature extractor, we use the algorithm described in [20] and available in [21], together with its associated matching algorithm. In our simulations, we refer to the iris pattern database known as CASIA V.1, provided by the Institute of Automation of the Chinese Academy of Sciences.

According to [22], [23], we evaluate the discrimination entropy over both the sets of iris templates and of their fuzzy hashes. For this purpose, we first compute the distribution of the normalized Hamming distances between all the couples of patterns within the set (of images of the same iris or of images of different irises). Then, we compute the mean μ and standard deviation σ of the normalized Hamming distance distribution. Finally, the discrimination entropy (also known as "Degrees of Freedom" or DOF) is obtained as $\text{DOF} = \mu(1-\mu)/\sigma^2$.

Applying fuzzy hashing to an iris recognition framework is not straightforward, due to the high variability in the irises acquisition phase. In fact, we must try to avoid all the differences given not only by the measure variability (i.e scale and rotation), but also by the eye variability, that can significantly change the amount of visible iris and its shape.

The standard way to take these issues into account is to compute a mask describing which bits in the iris template are free from such occlusions. The masks, in general, have a different number of set bits for each iris reading, resulting into information patterns having different lengths, both in the case they describe different irises and different readings of the same iris. This is not a problem in the case of the standard matching algorithms, since we can take, as inputs for the matching phase, the templates and masks of both the stored iris and the one we want to check. Then, we can just compute the intersection of the two masks and obtain the number of different bits within the two templates among the ones chosen by the masks intersection.

Instead, when we use syndromes, we cannot access the reference template in clear; so, we must cope with different lengths of the information patterns. One way is to treat the matching channel as an error-and-erasure channel [12], where erasures are given by the masks. Note that, in [12], the authors use a different algorithm from that we consider here. In our case, the large number of bits erased by the masks make this approach unusable. In order to obtain a fixed length of the information patterns, we compute, for each template bit position i, the probability $m(i)$ that such bit is not erased by a mask. Then, we compute a pseudomask selecting the bit positions corresponding to a value of $m(i)$ lower than a threshold: $m(i) \leq m_{th}$. In our case, we fix $m_{th} = 2.4\%$.

We are aware that this approach makes us neglect some bits that were not erased by their associated masks, but we have verified that this has a very limited effect for the considered algorithm. In fact, using all the selected bits in each template, we obtain, between two different readings of the same iris, an average Hamming distance of 28.24%, a standard deviation $\sigma = 0.0435$ and a discriminant entropy equal to 107 bits. Instead, using only the bits selected by the pseudomask, we obtain an average Hamming distance of 26.2%, a standard deviation $\sigma = 0.0486$ and a discriminant entropy equal to 81 bit. The explanation for this moderate variation, in terms of discriminant entropy performance, is that the feature extraction algorithm does not compute the masks in the best possible way (for example the two eyelids are approximated with straight lines and not with curves); so, when we take into account all the bits in each template, we commit some errors that afflict the result, that is, we take into account some bits that we should actually erase.

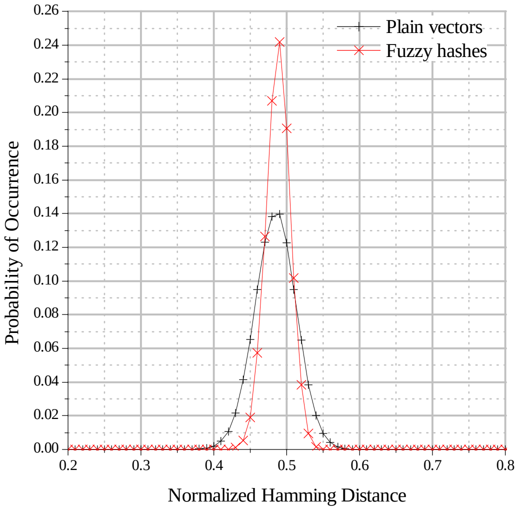

Fig. 3. Inter-class analysis of the Hamming distance for the considered set of iris templates with and without fuzzy hashing.

In order to show that the use of fuzzy hashing is able to increase the discriminant entropy, that is to provide a better protection against information leakage, we estimate the DOF on the set of plain templates, before and after the application of fuzzy hashing. The latter is performed through the LDPC code having $n = 9600$, $R = 0.1$ and $d_v = 5$, described in the previous section. We compute the normalized Hamming distance between each pair of templates created from different irises and then estimate its probability density function. The results, so obtained, are reported in Fig. 3.

The set of plain template vectors has $\mu = 0.4897$, $\sigma = 0.0281$, hence DOF $= 316.5$. After performing fuzzy hashing, the values become $\mu = 0.4932$, $\sigma = 0.0166$, DOF $= 907.1$. This confirms the positive effect of fuzzy hashing.

ACKNOWLEDGMENT

The Research was supported in part by the Swiss National Science Foundation under grant No. 132256.

REFERENCES

[1] 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, 2004.

[2] D. Schipani and J. Rosenthal. Coding solutions for the secure biometric storage problem. *Proceedings of IEEE Information Theory Workshop 2010*, Dublin 2010.

[3] P. Tuyls, B. Skoric and T. Kevenaar (Eds.). *Security with noisy data*. Springer, London 2007.

[4] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In editor G. Tsudik, editor, *Sixth ACM Conference on Computer and Communications Security*, pp. 28–36. ACM Press, 1999.

[5] A. Juels and M. Sudan. A fuzzy vault scheme. *Des. Codes Cryptogr.*, vol. 38, no. 2, pp. 237–257, 2006.

[6] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. *SIAM J. Comp.*, vol. 38, no. 1, pp. 97–139, 2008.

[7] I. Buhan, J. Doumen and P. Hartel, Controlling leakage of biometric information using dithering. In *Proceedings of 16th European Signal Processing Conference*, 2008.

[8] Y. Dodis and A. Smith, Correcting errors without leaking partial information. in *Proceedings of the Annual ACM 37th Symposium on Theory of Computing (STOC)*, pp. 654–663, 2005.

[9] Y. Sutcu, S. Rane, J.S. Yedidia, S.C. Draper and A. Vetro, Feature extraction for a Slepian-Wolf biometric system using LDPC codes In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2008)*, pp. 2297–2301, Toronto, ON, Jul. 2008.

[10] D. Schipani, M. Elia and J. Rosenthal, On the Decoding Complexity of Cyclic Codes Up to the BCH Bound. *Proceedings of IEEE Int. Symp. on Information Theory (ISIT 2011)*, Saint-Petersburg, Russia, Jul. 2011.

[11] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, 2001.

[12] J. Bringer, H. Chabanne, G. Cohen, B. Kindarji, and G. Zemor, Optimal Iris Fuzzy Sketches In *Proceedings of the First IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS 2007)*, Washington DC, Sep. 2007.

[13] R. G. Gallager, *Low-density parity-check codes*. M.I.T. Press, 1963.

[14] L. Bazzi, T. J. Richardson and R. L. Urbanke, "Exact Thresholds and Optimal Codes for the Binary-Symmetric Channel and Gallagers Decoding Algorithm A," *IEEE Trans. Inform. Theory*, vol. 50, no. 9, pp. 2010–2021, 2004.

[15] Z. Liu and D. A. Pados, "A decoding algorithm for finite-geometry LDPC codes," *IEEE Trans. Commun.*, vol. 53, pp. 415–421, 2005.

[16] N. Miladinovic and M. P. C. Fossorier, "Improved bit-flipping decoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 51, no. 4, pp. 1594–1606, 2005.

[17] M. Shan, C. M. Zhao, and M. Jiang, "Improved weighted bit-flipping algorithm for decoding LDPC codes," *IEE Proc.-Commun.*, vol. 152, pp. 919–922, 2005.

[18] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, 1996.

[19] X. Y. Hu and E. Eleftheriou, "Progressive edge-growth Tanner graphs," in *Proc. IEEE Global Telecommunications Conference (GLOBE-COM'01)*, San Antonio, Texas, Nov. 2001, pp. 995–1001.

[20] L. Masek, "Recognition of Human Iris Patterns for Biometric Identification", School of Computer and Software Engineering, The University of Western Australia, 2003.

[21] L. Masek, P. Kovesi, MATLAB Source Code for a Biometric Identification System Based on Iris Patterns, The School of Computer Science and Software Engineering, The University of Western Australia, 2003. [Online]. Available: http://www.csse.uwa.edu.au/∼pk/studentprojects/libor/sourcecode.html

[22] T. Cover, J. Thomas, "Elements of Information Theory", Wiley: New York, 1991.

[23] J. Daugmann. "How iris recognition works", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, n. 1, pp. 21–30, 2004.