Journal of Advanced Computer Science & Technology, 4 (1) (2015) 6-11
www.sciencepubco.com/index.php/JACST
©Science Publishing Corporation
doi: 10.14419/jacst.v4i1.3958
Research Paper

Coset decomposition method for storing and decoding fingerprint data

Mohamed Sayed

Faculty of Computer Studies, Arab Open University on Leave from Alexandria University, Faculty of Engineering
E-mail: msayed@aou.edu.kw

Abstract

Biometrics such as fingerprints, irises, faces, voice, gait and hands are often used for access control, authentication and encryption instead of PIN and passwords. In this paper a syndrome decoding technique is proposed to provide a secure means of storing and matching various biometrics data. We apply an algebraic coding technique called coset decomposition to the model of fingerprint biometrics. The algorithm which reveals the matching between registered and probe fingerprints is modeled and implemented using MATLAB.

Keywords: *Biometrics; Coset Decomposition; Fingerprints Matching; Syndrome Decoding.*

1. Introduction

With the increased integrations of computers and internet into individual's lives, it is substantial to protect sensitive and personal data. Now, biometric technologies are turning on the ground of an overall array of highly secure identification and personal verification solutions, see for instance [1] and [3]. Biometrics is largely automatic method of differentiating a person based on a physiological or behavioral characteristic. Examples of physiological characteristics include finger images, hand, facial characteristics and iris recognition. Behavioral characteristics are traits which are learned or acquired such as dynamic signature verification, speaker verification, gait, and keystroke dynamics. Biometrics method of identification shows various advantages over traditional methods such as ID cards (tokens) or PIN numbers (passwords) [8] for several reasons: The user to be sympathized has to present physically, often mandatory, at the point of identification and identification based on biometric techniques avoids the need to carry token or to remember any passwords. There are many types of biometrics currently in use, and many more new types are to come such as DNA and holograms. Different significant affairs have to be taken into consideration in order to design a practical biometric system. For instance, a user must be enrolled in the system so that his biometric template (or reference) can be captured. This template is securely stored in a central database or a smart card issued to the user. The template is used for matching when a user needs to be identified or authorized to login a system. It should be noting that a biometric system can operate either in verification (authentication) or an identification mode.

Fingerprints which are the patterns of friction ridges and valleys on an individual's fingertips are unique for each finger of any person even for identical twins [4]. For decades, the implementation has been determining and classifying identity by matching certain points of ridge endings and bifurcations. For example, fingerprint recognition devices for laptop, desktop and cell phones access, at a low cost, are now excessively obtainable from various vendors. Users no longer require to type passwords, instead, the users fingerprints provides immediate entry.

On the other hand, the codes provide a systematic way to send messages, with some additional information (check digits) in such a manner that an error occurring in the original messages will not simply be noticed (detected) by the receiver but, in many instances, may be adjusted. In parliamentary procedure for error correction to be effective, the decoding problem must be efficiently solvable. For security and matching reasons, it is desirable that the biometric information is stored in encrypted arrangement rather than in plain text. Similar to the problem of receiving and verifying a message through a noisy channel, when a user wants to access the system, the access device should allow access as long as biometrics does not vary by more than a definite number of binary digits. Many techniques which employ coding theory to tackle 'the secure biometric storage problem' are proposed. For instance, Martinian et al. [5] suggested an information theoretic solution which is based on the Slepian-Wolf theorem.

The purpose of this paper is to develop an algorithmic method (called coset decomposition method) that uses syndrome bits as secure storage and decoding fingerprints data. In general, the syndrome bits should contain a sufficient data such as the preserved fingerprint (or template) data to infer the user's information. Thus, the syndrome bits should have enough attributes such that the fingerprint is securely stored and then matched in a later phase. The implemented algorithm is used to overcome the disadvantage that a person who has access to the system may not match any of fingerprint information. Therefore, the algorithm should compute a bit string which will furnish access to the system even though the bit string is not shut to any fingerprint information.

For additional information about the syndrome decoding of biometrics information the reader is referred to [2], [7], and [15].

2. Fingerprints model

A fingerprint, as the name indicates, is the typography or the impression caused by the finger because of the style formed on the skin of the palms and fingers. It is completely formed at about seven months of fetus development and finger ridge configurations remain without changing throughout the whole life. With age, these marks get prominent but the pattern and the structures present in those fine lines do not undergo any change. Moreover, each of an individual ten fingerprints is different from one another as well as from those of each different person. For the fingerprints persistence and uniqueness, they have been used for not only in identification but also in the field of security as criminal and forensic investigation [17] for a long time.

In general, every fingerprint comprises of ridges and furrows where the ridges are thick lines while the furrows are space between two ridges. Therefore, the biological principles of fingerprints count on the individual epidermal ridges and furrows which have various characteristics for different fingers. It should be is noted that the configurations and types only differ within limits that authorize for orderly assortment. Here we utilize a minutiae-based representation, which might contain more global attributes such as position and orientation of figure, fingerprint class, etc. As we will see below, a prevalent technique for working with fingerprints data is to extract a set of minutiae points and to perform some operations on them. Each biometric identifier has its own distinguishable features that can be exploited for identification purposes. In the case of fingerprints, the most important features are the ridge configurations: the way the ridge lines and the valleys between them are arranged. The configuration of the ridge lines can be analyzed at three different levels: global, local and micro levels. At the global level, alertness is driven to regions at which ridge lines take shapes of high-curvature. These are called singularities or singular regions and can be distributing into three essential types: loop, delta and whorl. The patterns on the ten fingertips should be all different, but they might also have some similar features. We can find loops, whorls and arches on our fingertips. Some fingertips have only one singularity, but some have two types of singularity [11] on one fingertip.

At the local level, attention is paid to the ridge lines individually. A ridge line can be discontinuous in various ways. For example, it can turn up to an end suddenly, or it can divide into two ridges. The aim is to identify the point where a ridge line is discontinuous. These points are regarded as minutiae. Many types of minutiae can be identified from fingerprints, but the most common ones are termination, bifurcation, lake, independent ridge, island or point, spur and crossover.

At present, fingerprints are preserved digitally by scanning the user fingertip. The scanning process is simple and rapid. Fingerprint sensors which work in an analogous approach are particularly designed to capture details of the fingertip. The fingerprint sensors are ordinary taken on a two-dimensional array. They are covered by a pellucid coat of glass or plastic [10]. The most common sensor types are optical and solid state. Optical sensors work by shining light on to the fingertip which is placed on the transparent sensing surface of the sensor. They reveal the light that is inverted back on to the light-sensitive sensors. The ridges, which are in contact with the sensitive surface are, either scatter or absorb the light and consequently appear dark. In contrast, valleys, which are the gaps between ridges, appear lighter because they are at a distance from the surface and so allow the light to be reflected to the light-sensitive sensors. On the other hand, solid-state sensors were primarily designed to reduce the physical size as well as the expense of the sensors. The concept was to structure an all in one silicon chip with a 2-D sensory array placed directly on the chip. To provide fingerprint images, users only touch the sensing surface of the chip directly. The idea of solid-state sensors is transform thermal, capacitive, piezoelectric or electric field information to electrical signals. Because of their simplicity and low cost, the capacitive sensors are most common type used [6].

Fingerprint identification is the oldest method that has been successfully utilized in various computer systems. Fingerprint matching is a process of evaluating the degree of similarity (or difference) of two given fingerprints. One difficulty faced in the matching process is that some fingerprints from different fingers can be similar. The differences between fingerprints from different fingers are known as interclass differences, so problems occur whenever there are small inter-class differences. Another difficulty is that some fingerprints from the same finger can be different, known as intra-class differences, so problems occur whenever there are large intra-class differences. The intra-class variations are particularly problematic, as they are much more likely to happen. There are several reasons for intra-class variations: Displacement (different parts of the fingertip are presented to the sensor); rotation (the fingertip is presented to the sensor at a different angle); pressure of the impression (the finger is pressed on the sensor with a different force); skin condition (on different occasions the fingertip may be dry, wet, scratched or dirty); condition of the sensor surface (on different occasions the surface may be clean, dirty or greasy); feature extraction accuracy.

In general the matching procedures for fingerprints are categorized into: minutiae based matching, correlation based matching and ridge feature based matching. For instance, correlation-based matching works by superimposing one image over another image and changing their alignments until the correlation between the corresponding pixels of the two images is maximized. This is an intuitive method of matching fingerprints but the time and resources required to match the images pixel by pixel are huge. Sometimes, when the qualities of the fingerprint images are not good, minutiae extraction is difficult.

The outcome of the matching process could be a similar value, or it could be a decision of either match or no match. Either way, an algorithm is needed to evaluate the overall difference between the two fingerprints [9]. When the outcome of the matching is required to be a decision (match or no match), a threshold is required. The degree of similarity between two fingerprints has to be higher than the threshold for the system to consider them as a match. The threshold is usually set according to the required security level: the higher the threshold, the more difficult it is for two fingerprints to be considered as a match; the lower the threshold the easier it is for them to be considered a match. The threshold and the acceptable difference level are crucial in determining whether two prints are a match, and their values need to be considered carefully in all situations where fingerprints are used for identification or authentication. As identification is a process of identifying an individual from a population of individuals, if the population is large, it may take a very long time to search through the database. The sensitivity of a fingerprint recognition system is determined by thresholds. The thresholds used in biometric recognition systems set the balance point between security and convenience. For example, when a threshold is set too low, different biometric data can appear to match when they are not the same. This is known as a false match. Conversely, when a threshold is set too high, biometric data from the same person can appear not to match because of slight variations. This is known as false non-match.

To enhance the system performance, see [14], a common strategy is to divide the database into many bins. Each bin contains only fingerprints of the same class. When a fingerprint is to be identified, it is compared only with those in the bin of the same class. One simple and intuitive method is to classify the fingerprints using singularities. However, dividing the database into only five bins does not help much in improving the performance. Many real systems make use of other ridge information, such as ridge count between two distinctive features, to further divide the database into more bins. Other systems tag fingerprints with a number of attributes and classify them according to the tags.

To conclude this section, we have to raise dome important issues. False match refers to incorrectly believing that two given sets of biometric data are matched. The consequence of the former error is that imposters could gain access to resources they are not allowed to access. False non-match refers to incorrectly believing that two given sets of biometric data are not matched. The consequence of the latter error is that legitimate users could be refused access to resources they are entitled to access. In practice, these two types of error are unavoidable with current technologies but, ideally, both types should be kept to a minimum.

3. Modeling approach

As mentioned above, the fingerprint templates acquired from the same person are most probably different, and needs error-correction. Therefore, for the same person, we attempt to match the preserved (enrolled) fingerprint and the inquest (verified) fingerprint which is modeled as a noisy channel. Once an icon of a user's fingerprint is scanned, the position of the minutiae is initially detected, and the torus is then let out into a cuboids region. Next, a stack of "Gabor" filters is used to evoke a bit gradation. "Gabor" filter act as a directed smoothing process which removes residual random noise. A MATLAB implementation could be practiced to do this most critical step. Then, the extracted feature vector w is produced by giving out bits at certain specified positions that were ground to be unreliable. Lastly, the bit string w is represented (encoded) into the secure biometrics by computing the syndrome of w with respect to a low density parity check (LDPC) code. In fact error-correcting codes can provide a tight technique to overcome the variations in biometric data. The same schemes that have been offered in the context of fingerprint data can also handle iris, face, signatures and voice information. Some outlines that make function of multi-biometrics are also starting to come out, see [12], [16], [19], [20].

A prevalent technique for dealing with fingerprint data is to distill a group of "minutiae points" and to carry out subsequent operations on these minutiae. A minutia is a discontinuity in the ridge map of a fingerprint which is depicted by its locative in two dimensions (x, y) and the angular orientation θ, see [18]. We defined the minutiae map of a fingerprint as $\mathrm{Min}(x, y) = \theta$ if there exists a minutia point at (x, y). A minutiae map is considered as a feature extraction function. The minutiae map which acts on the fingerprint image is pictured using a binary matrix, where a 1-bit simply indicates the presence of minutiae at each concrete coordinate and 0-bit otherwise. It is commented that contrastive fingerprints normally have different numbers of minutiae. Furthermore, the number as well as the location of minutiae could slightly vary depending on the extraction algorithm that is practiced.

In addition to minutiae extraction, a feature transformation procedure that changes the two dimensional minutiae maps to binary feature vectors is utilized. The estimate is to generate binary feature vectors independent across different users such that different measurements of the same user are concerned by a binary symmetric channel. This is one of the principle channel models for low density parity check codes and therefore these standard codes can be used for Slepian-Wolf coding of the feature vectors. Following, we measure the number of minutiae points in a selected relatively small region across a training set of fingerprints. Then the threshold is defined as the median of the number of minutiae points in the chosen region. The threshold value may diverge for each area based on its location and intensity. If the number of minutiae points in any region overreach the threshold, then a '1' is added to the feature vector, otherwise a '0' is added. Eventually, we get an n-bit feature vector.

In summation and to conclude this section, we should consider these significant two questions: What type of error-correcting codes should be practiced in biometrics problem and what takes place if templates of biometric data come with redundancy. These two questions will lead us to the look for error correcting codes with low-rate and large-minimum. These codes which have the planned length should also come with efficient decoding algorithm.

4. Coding solution for fingerprint matching problem

Consider the problems of securely storing and matching fingerprints with the help of linear coding theory since, as motioned above, biometric data is stored in binary form. The author syndrome decoding coset decomposition algorithm in [13] will be revisited to give a reliable and secure storing and authentication of fingerprint data.

4.1. Preliminaries

In this part of the article we briefly recall a few classic notions needed in the constructions of the decoding algorithm which will be utilized in fingerprint matching problem. Let $F = \mathbf{Z}_2 = \{0,1\}$ be the group of two elements and let $n, k \in N$. As mentioned above, the biometric data is given in form of words (vectors) of length n as members of F^n, the direct product of n copies of F . This direct product is an "Abelian" group under the addition operation. The weight of $\mathbf{v} \in F^n$ is defined to be the number of nonzero entries in the vector $\mathbf{v}$. The distance, $dist(\mathbf{v},\mathbf{w})$ of $\mathbf{v},\mathbf{w} \in F^n$ is defined to be the weight of the difference $\mathbf{v}-\mathbf{w}$. Here $\mathbf{v}-\mathbf{w} = \mathbf{v}+\mathbf{w}$, as we are working in a product of copies of F in which every element is its own additive inverse.

We define a coding function $f : F^k \to F^n, k \le n$, and instead of storing a word $\mathbf{w}$, we store the word $f(\mathbf{w})$. There is a visible constraint on the selected coding function f : f is injective; otherwise there would be two distinct words of length k that would be received as the same word of length n . We say that (n,k) -code is a linear code over F if the images of f form a subgroup of F^n and the elements of such a code are called codeword. For $d \in N$ an (n,k,d) -code is an (n,k) -code for which d is the minimum distance between two different codewords. One preference of linear codes is that the minimum distance between codewords is comparatively easily found. We consider that there is an effective algorithm that is capable in decoding up to t errors, where $d = 2t+1$.

Let C be an (n,k) -code for some n and k . A generator matrix for the code C is a matrix $G \in F^{(n,k)}$ whose rows are an F -basis of C . The matrix G which generates the code C should have rank k . A vector $\mathbf{w} \in F^k$ is encoded as the vector $\mathbf{z} = \mathbf{w}G$. It is possible that during the identification some bits of $\mathbf{z}$ are changed. The system receives the incorrect message $\mathbf{y}$. The system solves the decoding problem, that is, it calculates $\mathbf{x} \in C$ such that $dist(\mathbf{x},\mathbf{y})$ is minimized. If $dist(\mathbf{z},\mathbf{y}) < \frac{d}{2}$, where d is the minimum distance of any two distinct codewords, then $\mathbf{x}$ is equal to the original vector $\mathbf{z}$.

In general any syndrome decoding technique, which is used to correct t errors in a codeword of length n , consists of main table including every binary n -tuples and the codeword into which it is to be detected. The rule for constructing this table is to decode an n -tuple into the nearest codeword. However, the table lookup decoding (coset decoding table) is feasible only for rather small codes. Therefore, one should persist looking for algorithmic decoding techniques which are considerably faster and request extremely less storage.

4.2. Syndrome decoding and coset decomposition

In this work, the preserved biometric $\mathbf{w}$ is binary and we use a linear code for the encoding function. Given the $k \times n$ binary generator matrix $G = \begin{pmatrix} I_k & A \end{pmatrix}$, we define the corresponding parity-check binary matrix H to be the $n \times (n-k)$ matrix $\begin{pmatrix} A \\ I_{n-k} \end{pmatrix}$. The syndrome $\mathbf{w} \in F^n$ is defined to be the H matrix product $\mathbf{w}H$ in F^{n-k} . This product is also referred to as the “coset” or “equivalence class” of $\mathbf{w}$. It should be noted that any codewords produced from the system generator matrix G should satisfy the condition $\mathbf{w}H = 0$. It is appropriate to put up two-column decoding table, one which contains just the column of coset leaders and the column of syndromes. Given a word $\mathbf{w}$ to decode, compute its syndrome, add to (subtract from, indeed) $\mathbf{w}$ the coset leader $\mathbf{u}$ which has the same syndrome - the word $\mathbf{w}+\mathbf{u}$ will then be corrected version of $\mathbf{w}$ - finally read off the first k digits to reconstruct the original word.

Let $\mathbf{v}$ stand for the received word when t-error correcting codeword $\mathbf{w}$ is transmitted over a channel corrupted by additive noise. Now $\mathbf{v} = \mathbf{w} + \mathbf{e}$, where $\mathbf{e}$ is a linear combination of some elements from the set $\{\mathbf{e}_i : 0 \le i \le n\}, \mathbf{e}_i \in F^n$ is the word of length n which has all digits 0 except the i^{th} digit which is 1. To find the codeword $\mathbf{w}$, the syndrome of the corrupted word $\mathbf{v}$ should be calculated, then this syndrome is expressed as combinations of the known syndromes, the error $\mathbf{e}$ is obtained as the same combinations of the corresponding coset leaders and in the end the corrected codeword is secured as $\mathbf{w} = \mathbf{v} + \mathbf{e}$.

The fingerprint matching is onerous for codeword lengths of several thousand bits and tens of errors per word. Consequently, we need, as mentioned above, an algorithmic decoder which demands less storage. Suppose that we have a coding function $f : F^k \to F^n$ for which the correlated code is linear code. Assume that the rate of errors is relatively high so that more parity check bits are added. Therefore, the number $n-k$ becomes large leading a longer decoding table. In [13] we showed that the number of coset leaders in this table is reduced from 2^{n-k} to $n+1$ which presents advantage over other coding algorithms. In gain to low demand for storage, lies in two facts: high capability of correcting random errors and notable simplicity of doing calculations. The operations which are combinations of codewords are all XOR operations, thus, it could be easily programmed into hardware to evolve a fast decoder.

4.3. Fingerprint matching algorithm

The major problem to get over is the fact that the schema demands the preserved fingerprint to be compared (or matched) with inquest fingerprint; and the hardness (or practically impossible) in matching takes place when the preserved fingerprint has been provided with feature vectors from a different user or an attacker. There are also some other aspects that has to be fixed; for instance the possibility of rubbing out an unordered collection of fingerprint features.

In our fingerprint matching algorithm, the generator and parity check matrices that define an error correcting code are clearly defined. Next, a fingerprint is scanned, and the minutiae are extracted and mapped to a binary feature vector. The redundantly encoded vector is obtained by applying the coset decomposition encoder. Finally, this encrypted vector of the fingerprint is retained (registered) in a secure storage medium for subsequent rapprochement during the phase of authentication.

The verification (or authentication) procedure is similar to the enrollment phase. A fresh taken fingerprint sample is captured (and processed) during the admittance. In order to compare this fingerprint sample against a previously stored sample, if it exists, the feature extraction is conducted. The feature vector result from this process is usually different from the corresponding encoded vector (secret) which is calculated in the enrolment steps. Here we have a problem that is identical to the problem of transmitting an encoded secret through a noisy channel. To fix this error (or to match two vectors), we use the coset decomposition decoder. Therefore, the system attempts to identify the individual from a stored fingerprint database samples.

A set of fingerprints measurements (acquired by Biometrika HiScan PIV Optical Fingerprint Scanner) has been used to evaluate the MATLAB implementation of the algorithm. All measurements have been successfully enrolled; measurements (authorized and unauthorized) have been attempted to serve as probes. The percentage of successful identification for unauthorized users is 100%, while the percentage of successful identification for authorized users exceeded 96%.

5. Conclusion

We developed an algorithm which is fit for matching fingerprints when several minutiae are missing or when some fictional minutia is detected. The algorithm could also be able to take care of translations, rotations and any further affine transformations. A number of modality biometrics matching algorithms comparable to the introduced algorithm have been proposed in the literature. However, the model for the secure biometrics problem based on the coset decomposition algebraic technique is, to the best of my knowledge, entirely new.

My next aim is to apply the same algorithm to different biometrics systems, for instance, the iris biometrics which seem very promising. In particular, the iris matching could be easier in terms of high true match rates because of the large amount of information that might be extracted from an iris.

References

[1] D. Blackburn, C. Miles and B. Wing, “National Science and Technology Council (NSTC), Subcommittee on Biometrics”, 2006.

[2] S.C. Draper, A. Khisti, E. Martinian, A. Vetro and J.S. Yedidia, “Secure Storage of Fingerprint Biometrics using Slepian-Wolf Codes”, in *Information Theory and Applications Work, UCSD*, San Diego, CA, 2007.

[3] A. Jain and S. Aggarwal, “Multimodal Biometric System: A Survey”, *International Journal of Applied Science and Advance Technology*, vol. 1, no. 1, pp. 58–63, 2012.

[4] D. Maltoni, A.K. Jain and S. Prabhakar, “Handbook of Fingerprint Recognition”, Springer: New York, 2005.

[5] E. Martinian, S. Yekhanin and J.S. Yedidia, “Secure Biometrics via Syndromes”, *Technical Report 112, Mitsubishi Electric Research Laboratories*, 2005.

[6] F.L. Podio and J.S. Dunn, “Biometric Authentication Technology: From the Movies to Your Desktop”, *NIST, 100 Bureau Drive, Stop 1070, Gaithersburg, MD20899-1070 [US Department of Commerce, 1401 Constitution Avenue*, NW, Washington DC 20230], 2005.

[7] V.K. Raina, “Integration of Biometric Authentication Procedure in Customer Oriented Payment System in Trusted Mobile Devices”, *International Journal of Information Technology Convergence and Services*, vol. 1, no. 6, pp. 15–25, 2011. http://dx.doi.org/10.5121/ijitcs.2011.1602.

[8] V.K. Raina and U.S. Pandey, “Biometric and ID Based User Authentication Mechanism Using Smart Cards for Multi-server Environment”, *INDIACom-2011, 5th National Conference on 'Computing for Nation Development' on 10-11 March, 2011 at BVICAM*, New Delhi, 2011.

[9] N.K. Ratha and R.M. Bolle, “Automatic Fingerprint Recognition Systems”, Springer: New York, 2004. http://dx.doi.org/10.1007/b97425.

[10] N.K. Ratha, J.H. Connell, R.M. Bolle and S. Chikkerur, “Cancelable Biometrics: A Case Study in Fingerprints”, in *International Conference on Pattern Recognition*, pp. 370–373, 2006.

[11] 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. http://dx.doi.org/10.1109/TPAMI.2007.1004.

[12] 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, pp. 280–286, 2005.

[13] M. Sayed, “Coset Decomposition Method for Decoding Linear Codes”, *Int. J. Algebra*, vol. 5, no. 28, pp. 1395–1404, 2011.

[14] M. Sayed. and F. Jradi, “Biometrics: Effectiveness and Applications within the Blended Learning Environment”, *Journal of Computer Engineering and Intelligent Systems (CEIS)*, vol. 5, no. 5, pp. 1–8, 2014.

[15] 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*, pp. 1–6, 2008.

[16] 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. http://dx.doi.org/10.1109/TPAMI.2006.250.

[17] J.R. Vacca, “Biometric Technologies and Verification Systems”, Elsevier Science & Technology, 2007.

[18] A. Vetro, S.C. Draper, S. Rane and J.S. Yedidia, “Securing Biometric Data”, in *Distributed Source Coding, P.L. Dragotti and M. Gastpar Eds.*, Academic Press, 2009. http://dx.doi.org/10.1016/B978-0-12-374485-2.00016-0.

[19] J.L. Wayman, A.K. Jain, D. Maltoni and D. Maio, “Biometric Systems Technology, Design and Performance Evaluation”, London: Springer, 2005.

[20] J. Yang, “Biometric Verification Techniques Combing with Digital Signature for Multimodal Biometrics Payment Systems”, *IEEE International Conference on Management of e-Commerce and e-Government*, pp. 405–410, 2010.