Application-Specific Key Release Scheme from Biometrics

Ong Thian Song, Andrew Teoh Beng Jin, and David Chek Ling Ngo
(Corresponding author: Ong Thian Song)

Faculty of Information Science and Technology, Multimedia University
Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia (Email: tsong@mmu.edu.my)

(Received Jan. 27, 2006; revised and accepted July 31, 2006)

Abstract

This paper outlines a novel biometric key release scheme to bind application-specific key from biometric data such that the key can be retrieved effectively by using Discrete-Hashing and Reed Solomon Block Coding (RSB). We use fingerprint as a subject of study and our experiment shows that the proposed method could retrieve an error free key reliably from a legitimate fingerprint up to 99.8% success rate with zero false acceptance and 0.12% of false rejection. In addition, our analysis suggests that the scheme is feasible to use in practice.

Keywords: Application-specific key, discrete-hashing, reed-solomon block coding

1 Introduction

Biometrics is the science of using unique human tangible parameters both in biological and behavioral for personal authentication. Biometrics identifiers such as face, fingerprint, iris, signature etc could be used to address non-repudiation problem in a typical cryptosystem, i.e. to confirm whether the person is who he/she claims to be.

Research on the incorporation of biometrics into cryptosystem to reap the benefit of both has been explored lately for more reliable and convenient authentication services to computer applications [9]. One of the approaches is known as biometric key release scheme where the key will be released for appropriate application upon a successful biometrics authentication. This scheme provides benefit as the key and biometric data are completely independent to each other. It is secure as the key can be easily modified or updated at the later time if it is ever compromised. Soutar [7] first proposed the key-release algorithm to securely bind a cryptographic key with user's biometric image. It was applied in an optical correlation-based fingerprint-matching system to form a secure block of data which known as Bioscrypt. Bioscrypt comprises a stored filter function, produced by a correlation-based image processing algorithm as well as other information that is required to first retrieve and then verify the validity of the key. However, it did not undergo any rigorous testing to prove the security viability of the algorithm and thus success rate is unknown. Furthermore, the assumption of no misalignment problem in all type of image acquisition system is too ideal due to the limited capacity of fingerprint scanner such as small sensing area or too little overlap between acquisitions of the same finger. Clancy et al. [2] implemented the technique of fuzzy vault as proposed by Juels-Sudan [5]. In Clancy's work, a group of minutia points were extracted from input fingerprint to bind in a locking set using polynomial-based secret sharing scheme. Subsequently, a non-related chaff point were added intentionally to obscure the key in order to maximize the unlocking computational complexity, where the secret key could only be recovered if there is a substantial overlap between the input and testing fingerprint. The method has been theoretically proven secure in protecting the secrecy of fingerprint. Nevertheless, it is way beyond the level of practical use due to the high false acceptance rate at 20-30%.

In this paper, we outlines a novel biometric key release scheme to bind application-specific key from biometric data such that the key can be retrieved effectively by using Discrete-Hashing [8] and Reed Solomon Block Coding (RSB). Our scheme enable: 1) to rectify the uncertainty of biometrics feature by using Discrete-Hashing and error correction technique so that it is stable enough to bind with application key and stored into a secure device like chip card. 2) the key release process is performed offline on secure device, which able to preserve the user privacy on every occasion of new key linking-releasing process for a specific application. 3) the key can be reproduced with high accuracy rate (very low FRR and zero FAR).

The outline of the paper is as follow: Section 2 gives the detail description of our scheme. Subsequently, we examine the viability of the scheme through different types of experiments and security analysis. The paper ends with a conclusion section.

2 Application Specific Key Release Scheme

Figure 1 illustrates the schematic framework of our proposed biometrics based application key release scheme. We distribute key linking and retrieving into two separate processing units, i.e., terminal unit and secure device. In our scheme, the fingerprint is first discretized into FingerHash based on the iterative inner-product between user's fingerprint and tokenized random number, r which can be produced from a seed in the secure device, i.e chip card or USB token. Subsequently, RSB is applied to correct the error (bit differences) within the reference and test FingerHash. Specifically, we formulate the process of linking and retrieving an error free application-specific key with the following step:

$$\begin{aligned} \textit{Biocode Generation}: &\quad b_c \bigoplus k = \beta \qquad (1) \\ \textit{Application Key Retrieval}: &\quad \beta \bigoplus t_c = k' \qquad (2) \end{aligned}$$

As shown in Equation (1) and (2), we notice that the core process of securely linking a key with its reference Fingerhash, b_c counterpart is known as Biokey generation, while the process of retrieving this key from test Fingerhash, t_c is known as key retrieval; whereby k refers to external application key and $\bigoplus$ denotes bitwise XOR operation.

In Equation (1), the external key, k is linked to b_c to derive Biocode, β and b_c are then discarded for the security purpose. During the key retrieval process, t_c which derived from a genuine user's fingerprint is used to unbind the Biocode through XOR process (Equation (2)) to generate k' , whereby the retrieval process is considered success if $H(k) = H(k')$ where H is the hash function.

Note that the scheme provides a good key protection characteristics - it allows multiple keys to be bound to FingerHash for different applications. This is to enable a random key with arbitrarily size (as determined by the dimensionality of FingerHash), which is unique to a specific application to be released. The key release is considered secure as it is performed in the secure device. Figure 1 illustrates how the secure device and terminal unit work cooperatively to make authorization decision, which cannot be made if one of the systems absent. Since both entities participate actively in the process, this mechanism ensures overall security of the proposed scheme.

2.1 Discrete-hashing Overview

Discrete-Hashing [8] is described in terms of successive simplifications as follow:

1) Feature Extraction. Filter bank-based feature extractor which was proposed in [4] is used in this paper to extract fingerprint feature and is represented in a vector format, $w \epsilon \Re^n$, with n denoting the feature length of w.

2) Use a token to generate m orthonormal pseudo random vectors,$\{r_{\perp i} \epsilon \Re_n | i = 1, \ldots, m\}$ and $m \leq n$.

3) Compute $\{< \Gamma | r_{\perp i} > | i = 1, \ldots, m\}$ where $< \cdot | \cdot >$ indicates the inner product operation.

4) Compute a m bit FingerHash template, $b = \{b_i | i = 1, \ldots, m\}$ from $b_i = \begin{cases} 0 \; if \langle \Gamma | r_{\perp i} \rangle \leq 0 \\ 1 \; if \langle \Gamma | r_{\perp i} \rangle > 0. \end{cases}$

Repetition of the above procedure renders the issue of independent multiple bits b_i of all others, so that legitimate (and unavoidable) variations in Ψ that invert b_i would not necessarily have the same effect on b_{i+1}, which is the necessary prerequisite to link and retrieve an application key.

2.2 RS Block Coding

Reed-Solomon (RS) code is an important subclass of BCH codes. It was introduced by Reed and Solomon [6]. Generally, RS code is a linear cyclic block code family and it allows multiple-error correction. It is designated as (n, k) block codes, whereby k is the number of data symbols input per block, and n is the number of symbols per block that encoder outputs. Given $z = n - k$ parity symbols, RS code can correct up to $z = 2t$ symbol errors in known positions (erasures), or detect and correct up to $z = t/2$ symbol errors in unknown positions. By the Berlekamp-Messey decoding algorithm [1], we can find the symbol error locations and correct up to $t = (n - k)/2$ error detected FingerHash.

3 Experiments and Discussion

To analyze the viability of the proposed scheme, we first examine the vigorousness of FingerHash in discriminating genuine and imposter fingerprint distribution. It followed with evaluation of the effect of RS error correcting to rectify the bit-differences in genuine samples of a class (person). Finally, we present our analysis by computing FAR-FAR to determine the recoverable of external key, k from BioCode.

The experiments are performed by using fingerprint images obtained from FVC2002 with 3 different set of databases, namely DB1, DB2 and DB3 [3]. All databases contain 8 impressions of 100 different fingers, hence yields 2400 in total. Preprocessing is done to detect the core point of every finger image and a 128 x 128 square region centered in the core point of the fingerprint images is extracted [10]. The experiments are conducted separately for DB1, DB2 and DB3 due to their fingerprint images are acquired by using different type of sensor.

To effectively measure the proposed RSB scheme, it is necessary to estimate how much variance between reference set, $\mathbf{b}$ and testing set, $\mathbf{t}$. For this purpose, the hamming distance is used to compare the number of corresponding bit positions that differ between both samples. To clarify this, let d be the Hamming distance, the closeness of each bit of two feature code set, $\mathbf{b_r}$ and $\mathbf{b_t}$ are

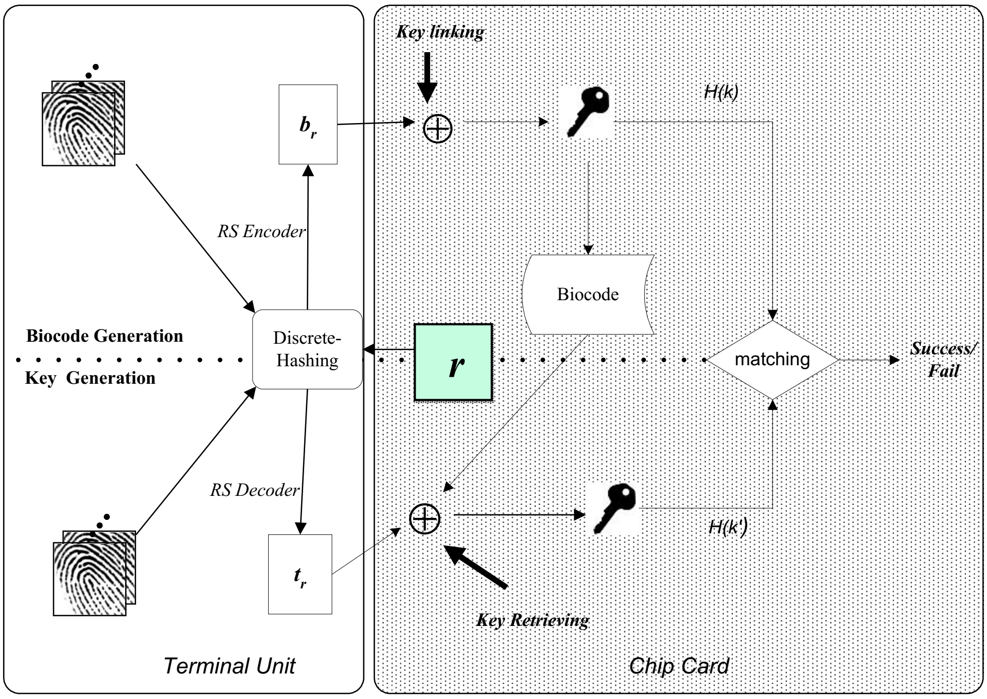


Figure 1: Application-specific key release scheme

then compared as:

$$d_b = \frac{\Sigma|b_r - b_t|}{w} \text{ for Fingerhash viability testing} \quad (3)$$

$$d_p = \frac{\Sigma|b_r^c - b_t^c|}{w} \text{ for RS error correction method testing,}$$

where b_r^c is the error corrected reference Fingerhash, m refers to the number of bit length and b_t^c refers to the error corrected test Fingerhash.

3.1 Performance Evaluation on Different FingerHash Setting

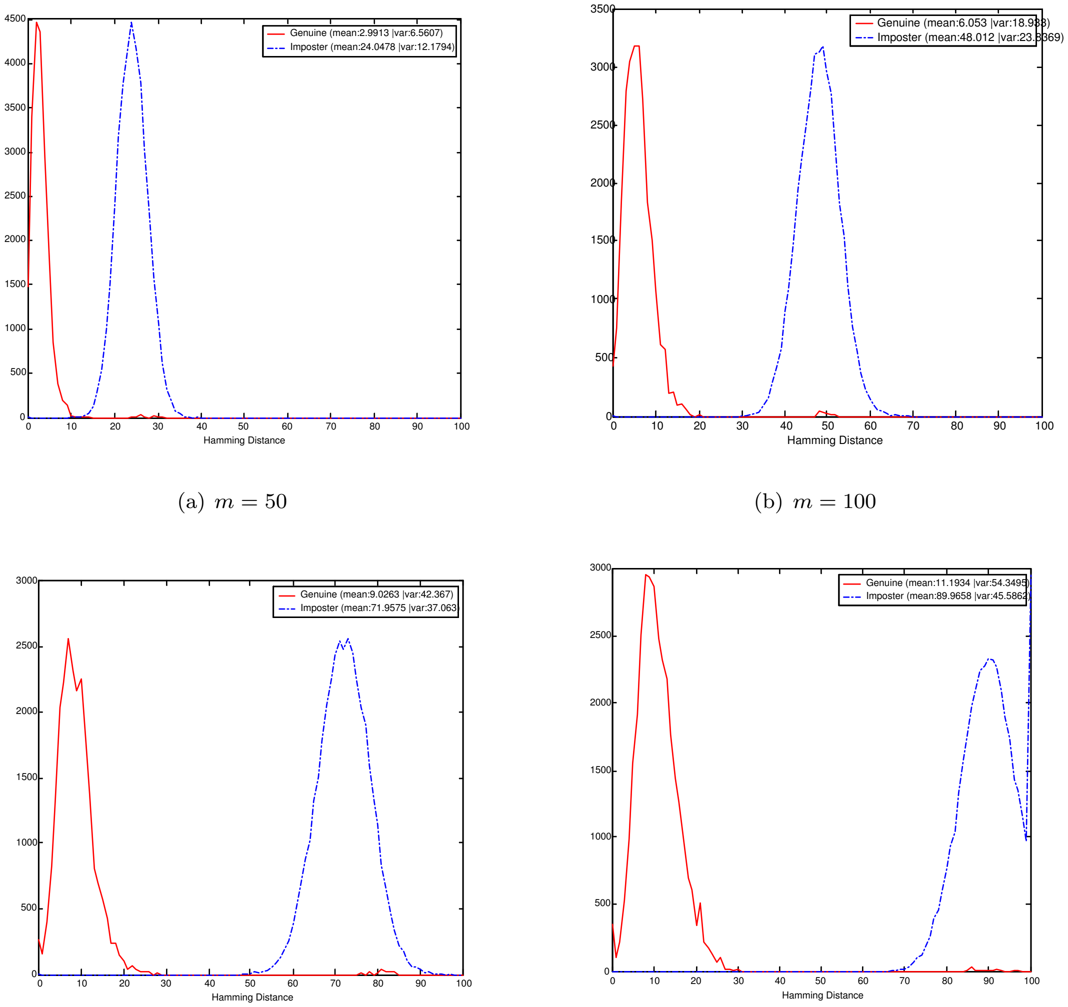
The objective of such a testing is to prove the viability of FingerHash in enhancing the discrimination power of RS error correction method. The evaluation of genuine and imposter population distribution (the accumulated scores of d_b) would be a good indicator for this purpose. Generally, the genuine distribution shows the results when different images of the same class are compared; but when images from different classes are compared, the imposter distribution is the outcome. The experimental data is tested for $m = 50, 100, 150$ and 180 based on the hamming distance similarity matching as described in the Equation (3). Figure 2(a) - 2(d) illustrate the performance in all the above cases respectively. The clean separation of genuine and imposter distribution prevent RSB coding to over correct the bit-differences of FingerHash - our error correction method only correct the bit disparity between the sample of same user (class) but not the uncorrelated bit differences from imposter users.

3.2 RS Block Coding Testing

In this experiment, the first FingerHash of each class (reference set) is used for encoding and the resulting parity check bit will be applied to the 7 others FingerHash (testing set) for correcting purpose. The genuine population (the accumulated scores of d_p) is generated by matching the FingerHash samples of each class against other 6 samples of the same class, whereas imposter population is obtained when the first samples of each class in the testing set is matched against the first samples of all the other different classes and the same matching process is repeated for others 6 samples in that classes.

To observe the effect of RS Block coding on the FingerHash, experiments are carried out to compare the mean of the genuine-impostor distribution before and after performing the RSB code, whereby hereafter we denoted them as non-RSB and RSB respectively. Table 1 depicts the value of mean and standard deviation based on RSB decoder (7, 3) scheme for pair-wise FingerHash comparison of genuine and imposter distribution. It manifests that the proposed RSB scheme for w=3 (no of bits per symbol block) has substantially reduced the mean of genuine samples for all three databases at average of 94-99% respectively. This is contrast to imposter means where the bit different between every sample image of different fingerprint remained unchanged. This analysis confirms our previous judgment that error of the bit code from different sampling of the same fingerprint can be significantly diminished without debasing the originality of imposter distribution.

To measure the performance for a decoder, we com-



(a) $m = 50$

(b) $m = 100$

(c) $m = 150$

(d) $m = 180$

Figure 2: Genuine and imposter distribution for $m = 50, 100, 150$ and 180

pute the ratio of corrected bits of reference sample, b_{ref} to total number of decoded (corrected) bits of the other seven testing set, t as bit error rate, e as:

$$e = \Sigma_{j=1}^{7}(b_{ref} - t_j)^2.$$

Typically, for a code to effectively combat the disparity of bit string, the data rate has to represent a relatively percentage of the codeword. A low code rate has a high detection probability, but a high redundancy. Nevertheless, this is not the case of codes operating in a fixed bit string length of FingerHash, it is interesting to observe the effect as shown in Table 2, which records the RSB error correcting capabilities based on different code rate from the decoder (15, k) settings. Since the length of m has been set to 180, we decided to choose $n = 15$ (with $w = 4$ bits per symbol) in order to test for the optimal parity bits that can best in improving the bit-error performance; from the result in the Table 2, it seems that RSB decoder with $k = 9$ demonstrate the optimum error correcting rate with bit error rate $= 0.0272$, of the genuine samples. We stop at $k = 7$ based on the rationality that the code would be over-corrected if too little data (k) use for the evaluation purpose.

3.3 Key Generation FAR-FRR Analysis

This section examines the accuracy of key retrievable from Biocode. The evaluation can be done by using False Acceptance Rate (FAR), False Rejection Rate (FRR) and Equal Error Rate (EER) as defined in below. In this context, we define FAR as the probability of accepting the imposter users to retrieve the key, while the FRR is the probability of rejecting a genuine user to retrieve the key.

Table 1: Mean and standard deviation of genuine and imposter distribution

	Non-RSB	**Non-RSB**	**RSB**	**RSB**	% difference
Database	*Genuine*	*Genuine*	*Genuine*	*Genuine*	
	M	σ	M	σ	
DB1	11.88	2.79	0.19	1.92	98.4
DB2	13.89	3.36	0.53	3.20	96.2
DB3	12.85	3.56	0.68	3.95	94.7
	Imposter	*Imposter*	*Imposter*	*Imposter*	
DB1	95.99	2.64	95.97	6.90	0
DB2	95.96	2.66	95.98	6.93	0
DB3	96.13	2.73	96.16	7.03	0

M - Mean σ - Standard Deviation

Table 2: Performance comparison of RSB decoder (15, k)

DB1				**Genuine**
n	k	$Coderate$	t	$Biterrorrate(e)$
15	13	0.86	1	0.0433
15	11	0.73	2	0.0337
15	9	0.6	3	0.0272
15	7	0.46	4	0.0278

Table 3: TFAR-FRR performance analysis on DB1,2 and 3 based on decoder (15,9)

Database	***FAR***(%)	***FRR***(%)
DB1	0	0.12
DB2	0	1.01
DB3	0	1.35

Therefore, the evaluation can be done by

$$\begin{aligned} \text{FAR} &= \frac{\text{Namber of accepted imposter claims}}{\text{Total Namber of imposter accesses}} \times 100\% \\ \text{FRR} &= \frac{\text{Number of reject genuine claims}}{\text{Total Namber of genuine accesses}} \times 100\% \\ \text{EER} &= \frac{\text{FAR} + \text{FRR}}{2}. \end{aligned}$$

From the analysis in the previous section, we found that the RSB decoder (15,9) is the optimal code setting to correct the bit disparity in the FingerHash. Thus, it would be used to perform FAR-FRR analysis on the three database set, DB1, 2 and 3 respectively and the empirical results are tabulated in Table 3. The table confirms that the proposed scheme could reproduce error free key reliably from the genuine samples up to a 99.88% success rate, and completely deterring illegitimate users from extracting the application key, or equivalently FAR = 0% and FRR = 0.12%.

4 Feasibility Analyzes

We perform the following analysis to study the viability of the proposed scheme.

4.1 Key Security Analysis

In this case we assume that an attacker has zero knowledge of our chip card and key release algorithm but attempt to discover key secret by using a brute force attack, which requires 2^{180} number of attempts to try on all the combination of external key. It is infeasible to perform such a huge amount of computational practically. The security of an application key could be further enhanced by increasing the dimensionality of m; there is no a-priori restriction on the value of m, as long as $d > m$, where d denotes the maximum length of features, f extracted from fingerprint data. In our case, since $d = 208$, we set $m = 180$ for application key, k binding and release purpose. Note that f can be increased easily by tessellate the region of interest into more sectors.

4.2 Key Release Analysis

Consider the worst scenario where a forger has accessed to our tokenized pseudo random number, r. Figure 3 illustrates the error equal rate (EER) for the following two cases that derived from the above scenario:

Case 1: $\{<r_A, \Gamma_A>, k)\}\varnothing\{<r_A, \Gamma_F>, k\}\}$
In this case, we assume that the tokenized r_A that belonging to genuine user A was stolen and a forger F have tried to release an application key by combining r_A with his/her own Γ_F.

Case 2: $\{<r_A, \Gamma_A>, k_i)\}\varnothing\{<r_A, \Gamma'_A>, k_i)\}$
This is the case when a genuine user, A wishes to recover the key by using his genuine FingerHash (with valid r).

From the Figure 3, it is important to observe that the EER of the Case 1 is poorer than normal Case 2, i.e., the performance of the prior case degrades substantially (EER$\sim>$ 50%) if compared to the Case 2. Notice that

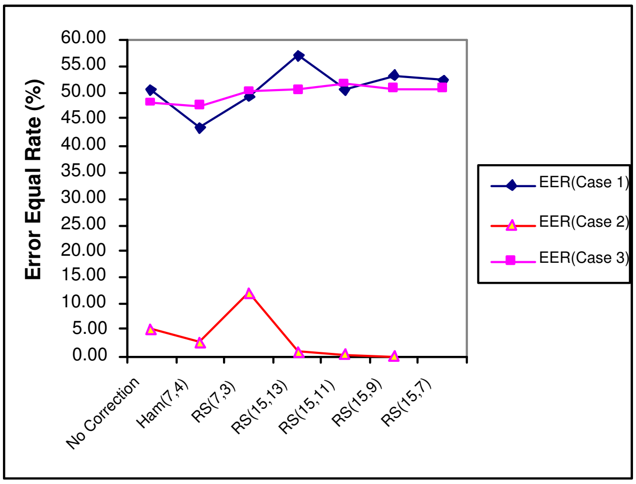


Figure 3: Equal error rate for Cases 1, 2 and 3

Note: Ham-hamming code

the Case 1 demonstrates the unfavorable EER results if a forger attempt to release a key by mixing his illegimate fingerprint with other user specific random number r_A.

4.3 Application Specific Key Authentication

As described in Section 2, our scheme enables reliable authentication of applications by using multiple keys scenario. For example, a user could have separate keys for accessing multiple applications such as bank account, workplace computer access control and other types personal account. The key diversity provide more practical mechanism as any key could be securely revoked without affecting the others. The following case analysis illustrates the uniqueness of our application specific key scheme:

Case 3: $\{\beta_i = (< r_A, \Gamma_A >, k_i)\}\varnothing\{(< r_A, \Gamma'_A >, \beta_j) = k_j)\}$
This is the case when a user attempts to reconstruct other non-correlated *jth* application key, k_j by using his *ith* Biocode, β_i.

Figure 3 above demonstrates the viability of our proposed method in preserving application specific key environment, as it is only allow each of the unique Biocode to be used for an application key reproduction - apparently the high EER ($\sim>$ 50%) in the Case 3 reveals that the accuracy of key retrieval would be seriously deteriorated if a forger attempt to use his sole Biocode for unlocking many other different applications.

5 Concluding Remarks

A scheme of application-specific key release from fingerprint has been presented for information assurance and personal authentication. The approach take advantages of discrete-hashing, RS error correction method and key release process to provide a novel way to link and retrieve multiple key for uniquely authenticate different application environments. The experimental results verify the proposed scheme demonstrates the rigorous of FingerHash and the potency of RSB to enable a high accuracy in term of the key reproduction with good key size, and the uniqueness of an application key. In addition, our feasibility analysis confirms the viability of the scheme.

References

[1] E. R. Berlekamp, R. E. Peile, and S. P Pope, "The application of error control to communications," *IEEE Communications Magazine*, vol. 25, no. 4, pp. 44-57, 1987.

[2] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure chip card-based fingerprint authentication," in *ACM SIGMM 2993 Multimedia, Biometrics Methods & Applications Workshop*, pp. 45-52, 2003.

[3] FVC2002. (http://bias.csr.unibo.it/fvc2002/)

[4] A. K. Jain, P. Salil, and H.Lin, "Filterbank-based fingerprint matching," *IEEE Transactions On Image Processing*, vol. 9, no. 5, pp. 846-859, 2000.

[5] A. Juel and M. Sudan, "A fuzzy vault scheme," in *Proceedings of IEEE Internation Symposium on Information Theory*, pp. 408, 2002.

[6] I. S. Reed, and G. Solomon, "RS codes over certain finite fields," *Journal of SIAM*, vol. 8, no. 2, pp. 300-304, 1960.

[7] C. Soutar, D. Roberge, A. Stoianov, and B. V. K. V. Kumar, "Biometric encryption using image processing," in *proceeding of SPIE*, pp. 178-188, 1998.

[8] A. B. J Teoh, D. C. L Ngo, and A. Goh, "Finger-Hashing: two factor authentication featuring fingerprint data and tokenized random number," *Pattern Recognition*, vol. 37, pp. 2245-2255, 2004.

[9] U. Uludag, S. Pankanti, S. Prabhakar and A. K. Jain, "Biometric cryptosystems: Issues and challenges," *Proceeding of The IEEE*, vol. 92, pp. 948-960, 2004.

[10] B. J. Andrew Teoh, T. S. Ong and N. C. L. David, "Automatic Fingerprint Center Point Determination," in *Lecture Notes of Artificial Intelligent (LNAI)*, vol. 2903, Springer-Verlag, pp. 633-640, 2003.

Ong Thian Song works in Faculty of Information Sciences and Technology (FIST), Multimedia University as a lecturer. His research interests include image processing, statistical pattern recognition and Biometrics security. He holds MSc degree from University of Sunderland, UK and is currently pursuing his PhD in the fingerprint security area.

Andrew Teoh Beng Jin obtained his BEng (Electronic) in 1999 and Ph.D degree in 2003 from National University of Malaysia. He is currently a senior lecturer and Associate Dean of Faculty of Information Science and Technology, Multimedia University Malaysia. He held the post of co-chair (Biometrics Division) in Center of Excellent in Biometrics and Bioinformatics in the same university. He also serves as a research consultant for Corentix Technologies in the research of Biometrics system development and deployment. His research interest is in multimodal biometrics, pattern recognition, multimedia signal processing and Internet security. He has published over 80 international journals and conference papers.

David Chek Ling Ngo was awarded a BAI in Microelectronics & Electrical Engineering and PhD in Computer Science in 1990 and 1995 respectively, both from Trinity College Dublin. Ngo's research interests lie in the area of Automatic Screen Design, Aesthetic Systems, Biometrics Encryption, and Knowledge Management. He is author and co-author of over 20 invited and refereed papers. He is a member of the IEEE.