A Secure Face Verification System Based on Robust Hashing and Cryptography

Eman A. Abdel-Ghaffar *, Mahmoud E. Allam †, Hala A. K. Mansour *, and M. A. Abo-Alsoud ‡
*Communications and Electronics Dept. - Benha University
Email: emaneng1@yahoo.com - hala.mansour@gmail.com
†School of Communication & Information Technology - Nile University
Email: allam@ieee.org
‡Electronics and communications Dept. - Mansura University
Email:mohyldin@ieee.org

***Abstract*— Face verification has been widely studied during the past two decades. One of the challenges is the rising concern about the security and privacy of the template database. In this paper, we propose a secure face verification system which employs a user dependent one way transformation based on a two stage hashing algorithm. We first hash the face image using a two stages robust image hashing technique, then the result hash vector is encrypted using Advanced Encryption Standard (AES). Both the hashing and the encryption/decryption keys are generated from the user claimed ID, using a modified password-based key driven algorithm. The proposed system is tested on the ORL (AT&T) face database.**

keywords: Face verification, Image Hashing, Encryption, Key Generation.

I. Introduction

The Uniqueness of biometrics makes them favorable in many applications requiring a high level of security [1], [2], [3]. Face verification became an intensive field of research since the early nineties, together with other biometrics (fingerprint, iris, retina, hand geometry,.....etc.). While fingerprint and iris scan can provide high accuracy rates, they still require complex and specialized scanners. On the contrary, face recognition can be performed with as simple a device as a web-cam, guarantying both a non-intrusive feeling from the scanned person, and a wide range of everyday applications. Various techniques have been used for face recognition [4], [5].

Despite the qualities of biometrics, they have a common shortcoming; most of the biometrics-based authentication systems need a template database, in which a biometric samples, and all users important information are saved. Recently, biometric template protection became one of the important issues in deploying a practical biometric system, a number of algorithms have been reported [6], [7]. In this work, higher template security is attempted by incorporating biometrics hashing [8], [9] and cryptography [10], [11].

This paper is organized as follows: The proposed face verification system is illustrate in section II. Section III to V are devoted to explain the underlaying techniques used in our proposed system. Modified password-based key derivation algorithm is investigated in Section III. In Section IV the used robust perceptual image hashing techniques and achieved results are illustrated. Advanced Encryption Standard (AES) technique is explained in Section V. Section VI is devoted to conclusion and future work.

II. The Proposed Face Verification System

In this section, we explain our proposed face verification system. The system uses robust perceptual image hashing followed by AES encryption to ensure reliable template protection. The hashing and encryption keys are generated from the user claimed ID using modified password-based key derivation algorithm. There is no restriction on the type of the claimed ID (could be a password, ID number or a user name).

Our algorithm was tested on the famous ORL (AT&T) face database which was constructed by AT&T Laboratories at Cambridge [12]. ORL consists of 400 92X112 grayscale images equally contributed by 40 subjects, 5 images for each individual were used for training and 5 for testing. There are variations in facial expression (open/closed eyes, smiling/non-smiling), and facial details (glasses/no glasses). All images were taken against a dark homogeneous background with the subjects in an up-right, frontal position, with tolerance for some tilting and rotation of up to about 20 degrees. A subset of ORL (AT & T) database is shown in Fig. 1.

During the enrollment stage Fig. 2, each user offers his claimed ID (there is no constrain on its length or character type) and a face image.

First, Modified Password-based Key Derivation algorithm (See Section III) is used to derive two different keys. A 16-bit key used to determine the location of the overlapping rectangles during the image hashing process. The second key is a 128-bit used as a cryptographic key.

The face image is hashed using two stages robust perceptual image hashing technique as illustrated in Section IV. The hash vector result from this step is encrypted using AES (See Section V) and the encryption key is the second key driven from the user claimed ID. The result encrypted hash vectors of all users is stored in the system database each in a separated record.

During the verification stage Fig. 3, the user offers his claimed ID and face image. The corresponding user record stored in the database is retrieved. Then, as in the enrolment

Fig. 1. Example of Face Images of 10 subjects from ORL Database.

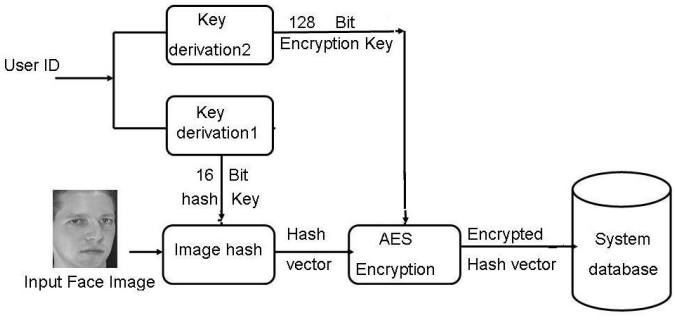

Fig. 2. Enrollment Stage.

stage user claimed ID based keys are generated, one is used during the hashing process and the other is used to decrypt the retrieved user encrypted hash vector. The Euclidean distance between the two hash vectors (the decrypted one and the generated one) is computed if it is less than threshold the user is who he claims to be. In this work, two types of thresholds were examined. The first, is a system threshold (user independent) used for all subjects in the database. The second, is a user dependent threshold determined for each user separately during the training stage. The use of user dependent threshold gave higher recognition rates as shown in table II.

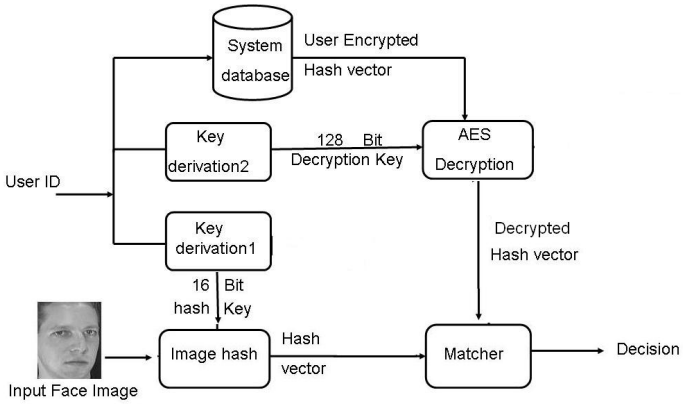

Fig. 3. Verification Stage.

III. Modified Password-Based Key Derivation Algorithm.

In this work, we need two keys, a 16-bit key for image hashing and a 128-bit encryption/decryption key. For strong hashing and cryptography, keys need to be with no particular pattern that can be recognized throughout the key sequence and with the highest possible entropy. Each user offers his claimed ID which is either a user name, a password or an ID number. There is no restriction on the type of the offered ID except that, it's the same each time he tries to access the system.

A number of approaches have been developed for strong key derivation such as: functional based, biometric based [11], [13] and voice based [14]. The functional based key derivation approach is often used to derive one or more keys from a common secret value (password); therefore, it may be referred to as a password-based key derivation.

A modified version of the Password-Based Key Derivation Algorithm in [15] based on the Key Based Random Permutation (KBRP) [16] method is used here. According to Shakir et al. [15], it generates a strong key regardless of the password elements and length.

KBRP is a method that generates one permutation of size N out of $N!$ permutations. This permutation is generated from a certain password by considering all characters of the password in the generation process. The permutation is stored in one-dimensional array, P of size N. The KBRP method consists of three steps:

- *init()*: This step initializes an array of size N with characters from the given password (password of length M characters). It stores the ASCII code of each character in the password in the array consecutively. If the password is less than N characters, the unfilled elements are derived by adding two consecutive ASCII codes. Finally, mod $N+1$ is calculated for each element, so that all values are set within the range 1 to N.
- *eliminate()*: This step eliminates the similar values by replacing them with zero and keep only one value out of these similar values.
- *fill()*: This step replaces all zero values with values in the range 1 to N that does not exist in the array. The resulted array is now representing the permutation P.

In [15] Shakir et al. added two more steps:

- *derive()*: Mod 2 operation is performed for each element in P. This ensures that each element in P will have either 0 or 1 and the number of 0's and 1's are even, so only one bit is complemented.
- *certify()*: This step, checks the run length r for the two binary digits 0 and 1.

In the proposed system, to add more security to the generated keys in the *derive()* stage we choose to complement element number M, and in the *certify()* stage we took $r = 2 + M/2$ (if M is odd we use add 1). By doing this, we ensure that the derived keys is completely user dependent (no single parameter is the same for the entire database). Table I shows some examples of different users claimed ID's and their corresponding generated keys.

Claimed ID	16 bit key (in hex.)	128 bit key (in hex.)
Karim Ahmed	451B	6F5AD8121378C4CD AD4D35545A92D6CA
znt287df4	5D9C	4C0D18630FC6AAED AD6956BAA35B552A
2768741	FB84	DE8C79B55095D253 AB5C5574EA90A6D8

TABLE I

USER CLAIMED ID AND IT'S CORRESPONDING GENERATED KEYS. IN FIRST ROW THE USER USED HIS NAME, IN THE SECOND ROW HE USED A PASSWORD, AND IN THE LAST ROW HE USED HIS ID NUMBER.

IV. ROBUST PERCEPTUAL IMAGE HASHING

Most of the biometrics- based authentication systems have a common weakest link, which is the need for a template database. If the database is attacked, there is no way to assign a different template, therefore, storing biometric templates should be avoided. However, the variability of the face images (for example, changing hair style, facial furniture) and the imperfect image acquisition conditions (different lighting conditions, pose, scale, camera gain,..etc.) prevents the use of secure cryptographic hashing algorithms for securing the biometric data. Secure cryptographic hashing algorithms such as MD-5 and SHA-1 give completely different outputs even if the inputs are very close to each other.

In recent years, researchers have proposed many different solutions. One of the most successful solutions is to use cancelable biometrics. The main idea is to use a noninvertible transform to map biometric data to other space and store the mapped template instead of the original one. A detailed survey of various cancelable biometrics techniques can be found in [11].

A robust image hash function has two inputs, an image I and a secret key κ and produces a short binary vector $\overrightarrow{h} = H\kappa(I)$ from a set $\{0,1\}^h$ (i.e., h bits long). The hash function should possess perceptual properties: Hash values for all "approximately-the-same" images are desired to be equal, such a hash function is a many-to-one mapping. For face image hashing we use Robust Perceptual Image Hashing Algorithm [17] described as follows:

Step 1 Let the $n \times n$ input image be $I \in R^{n \times n}$.

Step 2 From I, pseudo-randomly form p possibly overlapping rectangles (each of them of size $m \times m$ and their location is determined using the secret key): $A_i \in R^{m \times m}$, $1 \leq i \leq p$.

Step 3 Generate a feature vector $\overrightarrow{g_i}$ from each rectangle A_i via the transformation $\overrightarrow{g_i} = T_1(A_i)$.

Step 4 Construct a secondary image j by using a pseudo-random (PR) combination of intermediate feature vectors $\{\overrightarrow{g_1}, ..., \overrightarrow{g_p}\}$.

Step 5 From j, pseudo-randomly form r possibly overlapping rectangles (each of them of size $d \times d$ and their location is determined using the secret key): $B_i \in R^{d \times d}$,$1 \leq i \leq r$.

Step 6 Generate a final feature vector $\overrightarrow{f_i}$ from each rectangle B_i via the transformation $\overrightarrow{f_i} = T_2(B_i)$.

Step 7 Combine $\{\overrightarrow{f_1}, ..., \overrightarrow{f_r}\}$ to form the final hash vector (See Fig. 4).

The choice of the transformations $T_1$and T_2 is crucial for the system performance, in this work, we examined Singular value Decomposition (SVD) and Wavelet Transforms (WT) with different arrangements. Table II below shows the various arrangement and the achieved Genuine Accept Rate (GAR) and False Reject Rate (FRR). All the arrangements was tested with keeping the False Accept Rate (FAR) equal to zero, which makes our system more reliable.

As shown in table II, working with user dependent threshold gave better results than using a single system threshold. As, the result hash vectors will be encrypted during the enrollment stage, and the user encrypted hash vector will be decrypted during the verification stage. We choose to work with T_1 and T_2 as a Wavelet transform, as it gave best GAR with the shortest hash vector.

V. ADVANCED ENCRYPTION STANDARD (AES)

The final stage in the proposed system is to encrypt the hash vector before storing it in the system database. Rijndael block

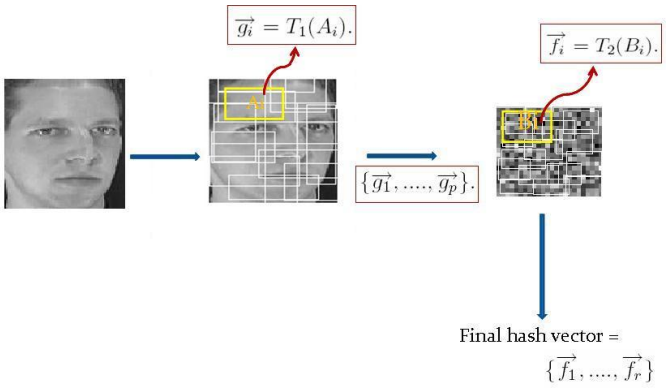

Fig. 4. Robust Perceptual Image Hashing.

T1 – T2 Transforms	User dependent threshold		System threshold		Hash vector Length
	GAR	FRR	GAR	FRR	
SVD - SVD	95.4%	4.6%	95.1%	4.9%	600
SVD - WT	95.9%	4.1%	95.1%	4.8%	135
WT - SVD	96.9%	3.1%	95.5%	4.5%	90
WT - WT	98.7%	1.3%	97.3%	2.7%	15

TABLE II
PERCEPTUAL IMAGE HASHING WITH DIFFERENT TRANSFORM T_1 AND T_2, AND THE CORRESPONDING GAR, FRR, AND HASH VECTOR LENGTH

cipher, is used here it was developed by Joen Daemen and Vincent Rijmen [18]. The algorithm is flexible in supporting any combination of data and key size of 128, 192, and 256 bits. It was selected to be the advanced encryption standard (AES) in 2000, replacing the 56-bit Data Encryption Standard (DES) cipher. AES allows a 128-bit data length that can be divided into four basic operation blocks. These blocks operate on an array of bytes organized as 4 x 4 matrix called the state. For full encryption, the data is passed through Nr rounds (in this work, $Nr = 11$ for a 128-bit block and key length). These rounds are governed by the following transformations [18]:

1) *AddroundKey transformation*: is a simple XOR between the working state and the roundkey (the key output from the key-scheduling operation).
2) *Subbyte transformation*: is a non-linear byte substitution, using an s-box, which is constructed by multiplicative inverse and affine transformation.
3) *Shiftrows transformation*: is a simple byte transformation where the bytes in the last three rows of the state are cyclically shifted; the offset of the left shift varies from zero to three bytes according to the row number.
4) *Mixcolumns transformation*: is equivalent to matrix multiplication. The output from the shiftrow operation is multiplied by a fixed matrix, It should be noted that the bytes are treated as polynomials rather than numbers.

The key is generated for each round using a key expansion algorithm [18]. The decryption structure has exactly the same sequence of transformation as in encryption structure, by using Inv-subbyte, Inv-mixcolumn, Inv-shiftrow, and AddroundKey allow the form of key scheduling to be identical for encryption and decryption. The Electronic Code Block (ECB) mode is used in which, every single data block is encrypted and decrypted independently from other blocks.

VI. Conclusion and Future Work

In this paper, we proposed a secure face verification system which employs a user dependent one way transformation based on a two stage hashing algorithm. Using cancelable biometrics will not only eliminate the need for storing biometric template in the database but also, provide flexibility to change the transform from one application to the other to ensure the security and the privacy of the biometric data. Using a two stage hashing technique increased, robustness, reliability and security. Both the hashing and cryptography keys are user dependent, derived from his own claimed ID (no restrictions on the offered ID as long as it's the same every time he tries to access the system), which offers more security. The keys are generated using modified password-based key derivation algorithm and is not stored in the system database. Furthermore, Adding encryption as a final stage, increases the security and overcomes attacks on the communication channels. Our system was tested on ORL face database using different transforms in the hashing technique and two types of thresholds.

Our future work will focus on two areas, (1) Testing our approach on a larger database, and different types of biometrics. (2) As no single biometric can be considered optimal, we are currently working on using multiple biometrics, and investigating the best fusion approaches.

References

[1] A. K. Jain, "An Introduction to Biometric Recognition", IEEE Trans. on Ciruits and Systems for Video Technology ,Vol. 14, No. 1, 2004.
[2] R. M. Bolle, J. H. Connell, S. Pankati, N. K. Ratha, and A. W. Senior, "Guide to Biometrics", Springer, New York, USA , 2004.
[3] A. K. Jain, A. Ross and S. Pankanti, "Biometrics: A Tool for Information Security", IEEE Trans. on Information Forensics and Security, Vol. 1, PP. 125 143, June 2006.
[4] A. S. Tolba, A. H. El-Baz, and A. A. El-Harby, "Faces Recognition: A Literature Review",Int. Journal of Signal Processing, Vol. 2, No. 1, PP. 88 - 103, 2005.
[5] X. Lu, Y. Wang, and Anil K. Jain, "Combining Classifiers for Faces Recognition", Proc. of ICME, July 2003.
[6] Y. C. Feng, P. C. Yuen, and A. K. Jain, "A Hybrid Approach for Face Template Protection", Proc. of SPIE Defense and Security Symposium, Florida, 2008.
[7] A. Vetro and N. Memon, "Biometric System Security", Tutorial presented at 2nd Int. Conf. on Biometrics, Seoul, South Korea, 2007.
[8] Y. Sutcu, T. Sencar, and N. Memon, "A Secure Biometric Authentication Scheme Based on Robust Hashing", ACM MM-SEC Workshop, 2005.
[9] A. Lumini and L. Nanni, " An Improved BioHashing for Human Authentication", Pattern Recognition, Vol.40, PP. 1057 - 1065, 2007.
[10] F. Hao, R. Anderson, and J. Daugman, "Combining cryptography with biometrics effectively", Tech. Rep. UCAM-CL-TR-640, Cambridge University, 2005.

[11] U. Uludag, S. Pankanti, S. Prabhakar, and A.K. Jain, "Biometric Cryptosystems: Issues and Challenges", Proceedings of the IEEE, Vol. 92, No.6, June 2004.
[12] the ORL (AT&T) Face Database, Available at http://www.uk.research.att.com/facedatabase.html.
[13] A.B. Teoh, D.C. Ngo and A. Goh, "Personalised Cryptographic Key Generation Based on FaceHashing", Computers and Security, Vol.23, PP. 606 - 614, 2004.
[14] F. Monrose, M.K. Reiter, Q. Li and S. Wetzel, "Using Voice to Generate Cryptographic Keys", Speech Recognition Workshop, Greece, 2001.
[15] S.M. Hussain and H. Al-Bahadili," A Password-Base Key Deriviation Algorithm Using the KBRP Method", Journal of Computer Science, Vol. 5, PP.777 - 782, 2008.
[16] S.M. Hussain,and N.M. A-Ajloni," Key Base Random Permutation (KBRP)", Journal of Computer Science, Vol. 2, No. 5, PP. 419 - 421, 2006
[17] S. S. Kozat, R. Venkastesan and M. K. Mihçak, "Robust Perceptual Image hashing Via Matrix Invariants" , Vol. 5, PP. 3443 - 3446, 2004.
[18] J. Daemen, and V. Rijmen, "The Block Cipher Rijndael", Proceeding of the 3rd Int. Conf. on Smart Card Research and Applicationa, Berlin, PP. 277 - 284, 2000.