A Secure Face Recognition System

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

Abstract— **In this paper, a secure face recognition system is presented, in which face detection is performed with skin color detection followed by light normalization and normalized cross correlation. Principal component analysis (PCA) is used for face verification. Due to the rising concern about the security and privacy of the biometric system, we offer a secure storage for user records by encrypting them using Advanced Encryption Standard (AES). A different encryption/ decryption key is used for each user, and that key is not stored in the database, it is extracted by expanding the submitted user identification (ID). A simulation of AES algorithm using field programmable gate array (FPGA) and the very high speed integrated circuit hardware description language (VHDL) is performed.**

I. Introduction

Today, technology driven society faces many security and privacy issues and one of them is reliable user authentication. Although, in most of the cases password-based authentication systems may be considered secure enough, the level of security is limited to the relatively weak human memory. An alternative approach is to use biometrics (Biometrics is the science of using unique human tangible parameters both in biological and behavioral for person authentication) instead of passwords for authentication. Higher entropy and uniqueness of biometrics makes them favorable in many applications that require high level of security. Recent developments in biometric technology enable widespread use of biometrics-based authentication systems [1], [2], [3].

Face recognition has become an intensive field of research since the early nineties, together with other biometrics verification methods (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]. One of the most popular techniques is the principal component analysis (eigenfaces). The eigenfaces technique attempts to capture the variation between facial images in an orthogonal basis set of vectors, referred to as *eigenfaces*. In other words, the eigenfaces are the image vectors which map the most significant variations between faces. It has been shown that any face image could be represented by a linear combination of eigenfaces, i.e. by a weight vector. The recognition is done by comparing the weight vectors of an image to the weight vectors in a database and finding the closest match. Several advantages made eigenfaces a suitable technique for the proposed system:

- *eigenfaces* have been shown to produce 96 % of correct classification under different lighting conditions [6].
- *eigenfaces* tolerate small variations in scale, rotation, and expression [6].
- *eigenfaces* have already been used in real-time systems, hence have an acceptable computational complexity [7].

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 sample (face image), and all users important information is saved. Recently, biometric template protection became one of the important issues in deploying a practical biometric system, a number of biometric template security algorithms have been reported [8], [9], [11], [12], [13].

In this work, higher template security is attempted by incorporating biometrics and cryptography [14] to reap the benefit of both into a more reliable and convenient authentication system. The Advanced Encryption Standard (AES) technique is used for encrypting the user records in the database.

AES can be programmed in software or built in hardware. However software implementation offers limited physical security. This work performs, a simulation of AES algorithm using field programmable gate array (FPGA) and the very high speed integrated circuit hardware description language (VHDL). FPGAdv. and ModelSim 6.2 software is used for compilation and simulation of the VHDL code. After verification of the functionality, the design is synthesized into a Xilinx spartan3 FPGA by using Mentor Graphics precision RTL synthesis.

This paper is organized as follows: Section II describes the proposed system architecture and all the steps evolved. Face detection is discussed in Section III. In Section IV face verification technique is explained. The biometric template

978-1-4244-2116-9/08/$25.00 ©2008 IEEE

Security of the proposed system is investigated in Section V. Section VI is devoted to conclusion and future work.

II. System Architecture

During the enrolment stage, each user offers 8 characters ID and a face image (face94 database [15] is used which consists of 153 individual, 20 image each, stored in 24-bit RGB JPEG format, 180X200 pixel each).

The user ID is matched against the one stored in the database, if it is correct the face image corresponding to that ID in the database is retrieved and face detection stage begins (section III). In the face detection stage, skin color detection followed by light normalization and Normalized Cross Correlation (NCC) are used for finding the exact location of the face in the image and extract it (the face box is 80x80 pixel image).

When the face is correctly detected the face verification stage starts (section IV). PCA technique is used for face recognition and because it has a high computational load, wavelet compression was performed to reduce the image size to 20 X 20 pixel before PCA face recognition. This dramatically reduce the computational load of PCA. The user will be offered three trials to give a correct ID and face image, if he fails an illegal attempt will be recorded.

If the user is who he claims to be, a correct access is recorded and the 8 character ID is expanded as described in section IV-B to form the 16 byte encryption/decryption key.

All the above stages ID match, face detection, face verification and ID expansion are programmed using Matlab7 as a software language. Finally, the the Matlab program will temporarily saves the 16 byte key to a file from which the FPGADV encryption/decryption VHDL program will read it. User record (name, address, bank account information,....etc) will be read from database and decrypted. Complete system architecture is shown in Fig.1.

III. Face detection.

Face detection is the process of determining the location and the size of one or more faces in an image. The face detection problem has been extensively surveyed in [16], [17] and a number of different techniques have been reported in [10], [18], [19], [20], [21], [22]. In this work we use the face94 database [15], and the face detection problem is limited to localize a single face in the image.

The proposed face detection system consists of four steps. The first step is to classify each pixel in the given image (180x200 pixel) as a skin or non-skin pixel. The second step is to perform some morphological operations (dilation and erosion) on the detected skin regions in order to form a candidate face region. The third step is to perform light normalization on the input image so that all images can be regarded as taken under the same lighting conditions. Finally, NCC is performed between the average face image (computed by averaging the faces in the database) and the skin detected image to get a face box (80x80 pixel).

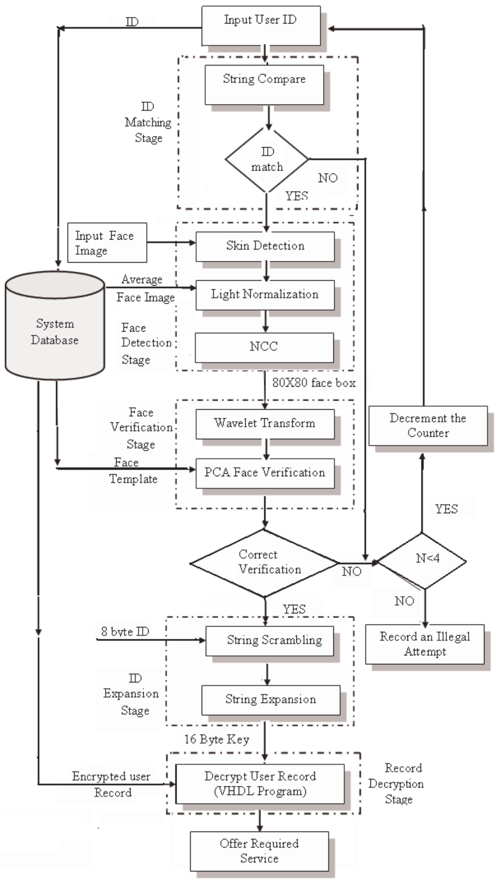


Fig. 1. System Architecture.

A. Skin Detection.

In the skin detection stage, the color content of each pixel is analyzed and determined to be either skin or non-skin. Different color spaces used in skin detection previously include HSV, normalized RGB, YCrCb, and YIQ. We adopted the popular HSV color space for our experiments as, according to Zarit et al. [23], HSV gives the best performance for skin pixel detection. Furthermore, the HSV space was reduced to its HS subspace by ignoring the V component, which contributes very little to the discrimination of skin tone [24]. The conversion from RGB to HSV is as follows,

$$H = \begin{cases} H_1, & \text{if } B \leq G \\ 360^o - H_1, & \text{if } B > G \end{cases}$$

where

$$H_1 = \arccos\left(\frac{\frac{1}{2}[R-G]+[R-B]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right),$$

$$S = \frac{max(R,G,B) - min(R,G,B)}{max(R,G,B)}$$

and

$$V = \frac{max(R,G,B)}{255} \tag{1}$$

B. Morphological Operations.

It is common during the color segmentation to return values that are close to skin but are actually non-skin, or other skin colored regions that are not part of the face. These noisy error values are generally isolated pixels or group of pixels that are significantly smaller than the total face regions, which would be represented by a big connected region in the binary image. Inclusion of these noisy pixels would result in a box that is much larger than intended and defeats the purpose of segmentation.

Since these spurious errors are much smaller than the face region itself, morphological techniques such as erosion and dilations are good tools to eliminate these pixels. For an improved bounded box, erosion followed by dilation is used in the system after the color segmentation to clean up the binary mapping prior to extracting the skinned region.

C. Light Normalization.

Light normalization involves adjusting the illumination of the image so that all images can be regarded as taken under the same lighting conditions. Careful attention needs to be paid in conserving the same amount of illuminance on all images such that no particular bias is placed upon lighting difference. the total amount of lighting can be normalized by observing the energy embedded in the image. The average face image is used as the standardized image, all input images should normalize their total energy so that it matches the total energy of the average face.

D. Normalized Cross Correlation (NCC).

NCC involves finding the best match between a template (the average face) and a sequence of windows. The basic principle behind NCC is to compare two windows of the same size and measure their correlation. A maximized correlation value means that the two windows under examination, have each of their corresponding pixels matching the closest among all the other windows under test.

Considering $W_L(x,y)$ as the window to be matched in image L of dimension N-by-N centered on the point (x,y), and $W_R(x+u,y+v)$ as the window in image R that is displaced from $W_L(x,y)$ by (u,v), the cross correlation of the two windows is defined as

$$CC = \sum_{i,j=-N/2}^{N/2} W_L(x+i,y+j)W_R(x+i+u,y+j+v) \quad (2)$$

To improve the plain CC method, which can be oversensitive to local characteristics, NCC is used, which divides the correlation by the standard deviations of the signals in the windows $\sigma_L\sigma_R$ [22].

$$NCC = \frac{CC}{\sigma_L\sigma_R} \quad (3)$$

IV. Face Verification

In the face verification stage, the input face image (after face detection stage) is compared against the one stored in the database corresponding to the offered user ID. PCA technique is used for face verification, but due to its high computational load, wavelet compression is used to reduce the image size to 20X20 pixel before PCA face recognition is performed. This work only deals with a subset of Face94 database [15]. The number of individuals is 50, with 20 images each, 15 images are used in the training stage and 5 images are used for testing.

A. Wavelet Analysis.

Wavelet decomposition provides local information in both space and frequency domains. The wavelet transform (WT) [28] is one of the methods that have been investigated to compensate for the high computational load in finding the eigenvectors. In the proposed system, each face image is 180x200 pixel, and after the face detection stage we have a face box (80x80 pixels). The computational complexity to find the eigenvectors of such image is estimated to be $O(d^3)$, where d is the total number of pixels. From matrix theory, in the case where the number of training images N is smaller than d, the complexity will be reduced to $O(N^3)$. But if we are adding regular users to the system, N becomes large. Recomputing every time the eigenspace will be done with a load increase of cubic order, thereby leading to undesired computation cost. PCA on WT has been shown to address this issue. In particular, it has been possible to select a 16x16 sub-band, and still get excellent recognition rates and discrimination power.

B. Eigenface Calculation.

Eigenfaces is one of the most thoroughly investigated approaches to face recognition. It is also known as Karhunen Loève expansion, eigenpicture, eigenvector, and principal component. The PCA has been used efficiently to represent faces. In Refs. [26], [27] it was shown that, any face image could be approximately reconstructed by a small collection of weights for each face and a standard face picture (eigenpicture). The weights describing each face are obtained by projecting the face image onto the eigenpicture.

In mathematical terms, eigenfaces are the principle components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images. The eigenvectors are ordered to represent different amount of variation among the faces [4]. Each 2D face image is considered as a vector, by concatenating each row (or column) in the image. Let $X = (X_1, X_2, ..., X_i, ..., X_N)$ represent the n x N data matrix, where each X_i is a face vector of dimension n, concatenated from p x p face image, where p x $p = n$. Here n represents the total number of pixels in the face image and N is the number of face images in the training set. The main vector of the training images $\mu = \sum_{i=1}^{N} X_i$ is subtracted from each image vector. PCA can be considered as a linear transformation from the original image vector space to projection feature vector space, i.e.

$$Y = W^T X \quad (4)$$

where Y is the d x N feature vector matrix, d is the dimension of the feature vector, and W is the transformation matrix. Note that $d << n$. PCA basis vectors are defined as the eigenvectors of the matrix S_T,

$$S_T = \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T \quad (5)$$

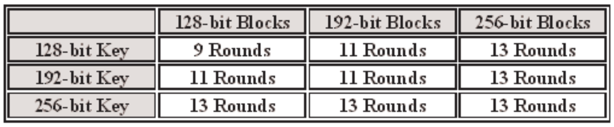
The transformation W is composed of the eigenvectors corresponding to the d largest eigenvalues. After applying the projection, the input vector (face) in an n-dimensional space is reduced to a feature vector in a d-dimensional subspace [5].

V. Biometric Template Security.

There is a rising concern about the security and privacy of the biometric data itself [11]. Face recognition systems, like most biometric recognition systems, need a template database. Saving user information in a database is considered the weakest link in the system. In the proposed system, stored templates are secured by encryption to increase security. Even if the attacker could access the database, he will not be able to understand the saved information. The AES algorithms is used for encryption and decryption. For more security (due to the drawbacks of software encryption systems) AES VHDL program is simulated (using Modelsim simulator) and after verification of the functionality the core is synthesized into a Xilinx FPGA using Mentor Graphics precision RTL synthesis.

A. Advanced Encryption Standard (AES).

Rijndael is a block cipher developed by Joen Daemen and Vincent Rijmen. 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 merely 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 (see Fig. 2). For full encryption, the data is passed through Nr rounds, the number of rounds is a function of the key and block lengths as shown in table I .



	128-bit Blocks	192-bit Blocks	256-bit Blocks
128-bit Key	9 Rounds	11 Rounds	13 Rounds
192-bit Key	11 Rounds	11 Rounds	13 Rounds
256-bit Key	13 Rounds	13 Rounds	13 Rounds

TABLE I
Number of rounds in Rijndael as a function of block length and key length.

These rounds are governed by the following transformations:

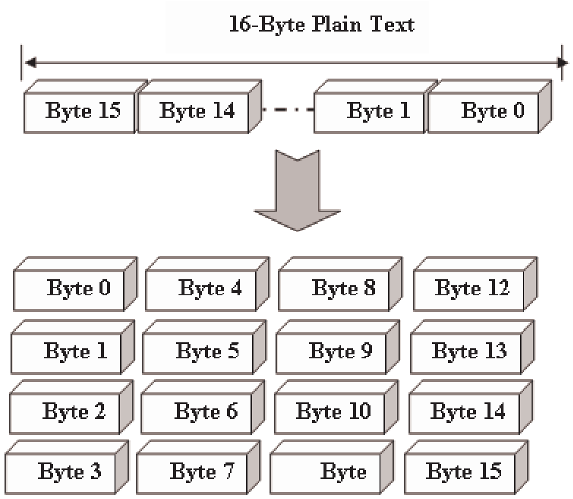


Fig. 2. Two Dimensional State Matrix.

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 a substation table (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 matrix output from the shiftrow operation is multiplied by a fixed matrix,

$$\begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}$$

It should be noted that the bytes are treated as polynomials rather than numbers.

In a 128-bit block Rijndael encryption, plain text and cipher text are processed in blocks of 128 bits. As shown in Fig. 2, intermediate values of Rijndael are represented as $4Nb$ state matrix of bytes, Nb = (block length / 32). Similarly, the input and round keys are represented as a $4Nk$ matrix of bytes, Nk = (key length / 32). In our hardware implementation we set the block length and key length to 128 bits. Hence $Nb = Nk = 4$.

The encryption procedure consists of several steps as shown in Fig. 3. After the initial addroundkey, a round function is applied to the data block (consisting of Subbyte, Shiftrows, Mixcolumns, and Addroundkey transformation, respectively). It is performed iteratively Nr times depending on the key length. For a 128 bit block length and key length, 11 roundkeys are needed (1 for the initial round, 9 for the standard rounds, and 1 for the final round). The roundkeys is generated recursively. The cipherkey is described as a matrix;

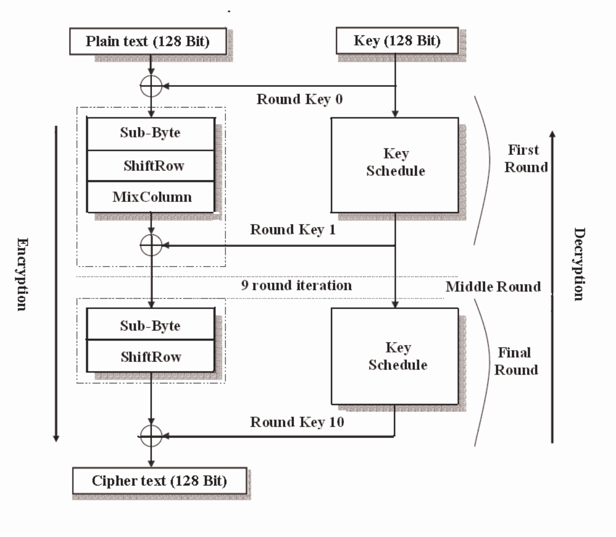


Fig. 3. Encryption and Decryption Flow of AES.

$$K = \begin{pmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \end{pmatrix}$$

The i-th column of K is denoted by W_i. The key schedule is a method to extend K with more columns (for 11 rounds we need 44 column). In our case 128 bit key and block length, the key expansion equations is as follows [29], [30]:

$$SW = SubWord(S_1(W_{i-1}))$$

$$W_i = \begin{cases} W_{i-N_k} \oplus SW \oplus rcon(\frac{i}{N_k}), & \text{if } i \bmod N_k = 0 \\ \\ W_{i-N_k} \oplus W_{i-1}, & \text{if } i \bmod N_k \neq 0 \end{cases} \quad (6)$$

where the function $S_1(W_{i-1})$ is a cyclic shift of the elements in W_{i-1}. The function $SubWord$ is a Subbyte operation on each byte of the word vector separately. $rcon(\frac{i}{N_k})$ is a vector, that is defined as $rcon(i) = [x^{i-1},' 00',' 00',' 00']$, with x^{i-1} being powers of x in the Galois Field $GF(2^8)$.

The decryption structure has exactly the same sequence of transformation as in encryption, by using Inv-subbyte, Inv-mixcolumn, Inv-shiftrow, and AddroundKey allow the form of key shedulling to be identical for encryption and decryption.

B. ID Expansion.

Each user in the system has his own ID, which consists of eight characters (its ASCII representation in hex. is 8 bytes). An extension rule is used to expand the ID to a 16 byte encryption/decryption key. The ID expansion is done in two steps: First, perform a string scrambling technique (convert all the characters to lower case then, arrange them alphabetically in descending order). Then, perform a string expansion technique using a simple pseudo random generator.

The generated encryption key is used to encrypt each user record that contain all the important user information (name, bank account number, employment, ..etc.). Extending the ID to form an encryption key gives us four main benefits;

1) We are asking each user to memorize only his ID (eight characters) instead of asking him to remember the entire sixteen byte encryption key.
2) There is an ID for each user, therefore each user will have his/her own encryption/decryption key. If an attacker was able to decrypt one record in the database he will not be able to decrypt all the other users records.
3) We do not need to save the encryption/decryption key in the database. For an attacker, even if he access the database, he needs to know a valid ID and a correct ID expansion rule to be able to decrypt a single user record.
4) A variety of techniques could be used to expand the 8 byte ID to 16 byte encryption key. Increasing the complexity of the expansion technique will increase the difficulties an attacker will face.

C. Hardware Encryption.

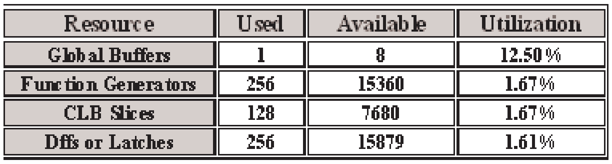
Rijndael has been implemented in software using C, C++, Java, Matlab, and assembly [30]. Software offer limited physical security especially with respect to key storage. The proposed system simulates a field programmable gate array (FPGA) implementation of Rijndael. Such a hardware implementation cannot be easily modified and hence it is physically secure against attacks. For various software and hardware implementations of Rijndael see Refs. [31], [32], [33]. While pipelined ASIC implementations are the fastest, assembly language implementation on an MC 6805 is the slowest.

In this work, design and implementation of two configurable and flexible cores of AES algorithm are done: an encryptor and a decryptor. The two cores are designed using Electronic Code Block (ECB) mode meaning that every single data block is encrypted and decrypted independently from other data blocks. Since ECB is the basic element of all other main modes such as Cipher Block Chainning (CBC), Cipher Feedback (CFB), and output Feedback (OFB), it is easy to extend the design and implement the other modes.

All the modules in these flexible cores are created using very high speed integrated circuit hardware description language (VHDL) [34]. Some modules are designed using behavioral style and some are designed fully RTL.

Firstly a common package is created to set the global values and functions used by all of the modules and sub-modules. Secondly, Sub-modules Byte Substitution, Row Shifter, Column Mixer, Round Key Adder and key Expander for the encryptor, inverse of all these modules, except Round Key Adder since it is identical for both, for the decryptor are designed. All sub-modles are designed using behavioral VHDL. Thirdly, a controller to control these sub-modules and round operations are designed for the encryptor and for the decryptor by using RTL style.

Simulations for the two cores are run by using Modelsim simulator (using the test vectors provided by AES submission package [30]). After verification of the functionality, the cores are synthesized into a Xilinx spartan-3 FPGA using the Mentor Graphics Precision RTL synthesis which is an integrated development tool. The Device Utilization for Spartan-3 (3S1000ft256) for the encryptor is as shown in the table 2.



Resource	Used	Available	Utilization
Global Buffers	1	8	12.50%
Function Generators	256	15360	1.67%
CLB Slices	128	7680	1.67%
Dffs or Latches	256	15879	1.61%

TABLE II
DEVICE UTILIZATION FOR SPARTAN-3.

VI. CONCLUSION AND FUTURE WORK

This work presents a secure face recognition system, which performs face verification based on user ID and PCA face recognition technique. An accurate face box detection was accomplished by skin color detection followed by NCC and between the two stages light normalization was performed. The proposed system offered a correct recognition rate of 96%.

To overcome the non-secure template storage problem, incorporate biometrics and cryptography to reap the benefit of both into a more reliable and convenient authentication system is used. The AES technique is used to encrypt the user records before storing it in the system database, and decrypt it again after the user is correctly recognized. Different encryption/decryption key is used for each user. As software implementation of AES offers limited physical security, a simulation and synthesization of AES algorithm was performed.

As each biometric has its own advantages and disadvantages, no single biometric can be considered optimal. Therefore future work will focus on using multiple biometrics for personal authentication, and investigating the best fusion approach.

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 Transactions on Information Forensics and Security*, Vol. 1, No. 2, pp. 125143, 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] M. TURK, A. Pentland, ”Face Recognition Using Eigenfaces”. *In proc. IEEE CVPR*, Maui, pp. 586-591, June 1991.

[7] A. Pentland, B. Moghaddam, and T. Starner, ”View Based and Modular Eigenfaces for Face Recognition”. *Proc. IEEE conf. Computer Vision and Pattern Recognition*, Seattle, pp. 84-91, June 1994.

[8] A. K. Jain, A. Ross, and U. Uludag, ”Biometric Template Security: Challenges and Solutions”. *Proceedings of European Signal Processing Conference (EUSIPCO)*, Antalya, Turkey, September 2005.

[9] Y. C. Feng, Pong C. Yuen, and Anil K. Jain, ”A Hybrid Approach for Face Template Protection”. *Proceedings of SPIE Defense and Security Symposium*, Orlando, Florida, 2008 .

[10] K. Sandeep and A.N. Rajagopalan,”Human Face Detection in Cluttered Color Images using Skin Color and Edge Information”. *Proc. Indian Conference on Computer Vision, Graphics and Image Processing*, Dec. 2002.

[11] S. Prabhakar, S. Pankanti, and A. K. Jain ”Biometric Recognition Security and Privacy Concerns”. *IEEE Security and Privacy Magazine*, Vol. 1, No. 2, pp. 33-42, March-April 2003.

[12] P. Tuyls, A. H. M. Akkermans, T. A. M. Kevenaar, G.-J. Schrijen, A. M. Bazen, and R. N. J. Veldhuis, ”Practical Biometric Authentication with Template Protection”. *Proceedings of Fifth International Conference on Audio- and Video-based Biometric Person Authentication”*, Rye Town, USA, pp. 436 - 446, July 2005.

[13] A. Vetro and N. Memon, ”Biometric System Security”. *Tutorial presented at Second International Conference on Biometrics*, Seoul, South Korea, August 2007.

[14] U. Uludag, S. Pankanti, S. Prabhakar, and A.K. Jain, ”Biometric Cryptosystems: Issues and Challenges”. *In Proceedings of the IEEE,* Vol. 92, June 2004.

[15] Face Recognition Data, University of Essex,Uk. *Computer Vision Science Research Projects*, http://cswww.essex.ac.uk/mv/allfaces/index.html.

[16] Ming-Hsuan Yang, D. J. Kriegman, and N. Ahuja, ”Detecting Faces in Image: A Survey”. *IEEE Trans. on Pattern Analysis and Machine Intelligence* , Vol.24, no. 1, pp. 34-58, January 2002.

[17] E. Hjelmas, and B. K. low, ”Face Detection: A Survey”.*Computer vision and Image Understanding*, Vol.83, No. 3, pp. 236-274, 2001.

[18] R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain,”Face Detection in Color Images”. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.24, No. 5, pp. 696-706, May 2002.

[19] P. Viola, and M. J. Jones, ”Robust Real-time Face Detection”. *Int. J. Computer Vision* , Vol.57, No. 2, pp. 137 154, 2004.

[20] K. P. Saeed, F. Karim, and F. Hajati, ”Face Detection Based on Central Geometrical Moments of Face Components”. *IEEE conference on systems, Man.*, pp. 4225-4230, Oct. 2006.

[21] C. Lin, ”Face Detection by Colour and Multilayer Feedforward Neural Network”. *IProc. IEEE International Conference on Information Acquisition* , pp. 518-523, June-July 2005.

[22] Chin-Chuan Han, Hong-Yuan M. Liao, Gwo-Jong Yu, and Liang-Hua Chen, ”Face Detection Via Morphology Based Pre-processing”. *Pattern Recognition*, pp.1701-1712, 2000.

[23] B. D. Zarit, B. J. Super, and F.K.H. Quck, ”Comparison of Five Color Models in Skin Pixel Clasification”. *ICCV'99 Int. Workshop on Recogniotion, Analysis and Tracking of Faces and Gestures in Real Time Systems*, 1999.

[24] Qiang Zhu, Kwang-Ting Cheng, Ching-Tung Wu, and Yi-Leh Wu, ”Adaptive Learning of an Accurate Skin-Color Model”. *IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pp. 37-42, May 2004.

[25] L. Sirovich and M. Kirly, ”Low-Dimensional Procedure for the Characterisation of Human Faces”. *J. Optical Soc. of Am.*, Vol. 4, pp. 519-524, 1987.

[26] M. Kirly and L. Sirovich, ”Application of the Karhunen Loève Procedure for the Characterisation of Human Faces”. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 831-835, Dec. 1990.

[27] M. Turk and A. Pentland, ”Eigenfaces For Recognition”. *J. Cognitive Neuroscience*, Vol. 3, pp. 71-86, 1991.

[28] A. Graps, ”An Introduction to Wavelets”. *IEEE Computer Science And Engineering,* Vol. 2, Num. 2, 1995.

[29] J. Daemen, and V. Rijmen, ”The Block Cipher Rijndael”. *Proceeding of the 3rd Int. Conf. on Smart Card Research and Applications*, Lecture Notes in Vomputer Science, Vol. 1820, pp. 277-284, 2000.

[30] J. Daemen, and V. Rijmen, ”The Rijndael Block Cipher ”. *AES Proposal*, Ver. 2, March 1999.

[31] K. Aoki, and H. Lipmaa, ”Fast Implementation of AES candidates ”. *3rd AES Conference*, pp. 106-120, April 2000.

[32] G. Keating, ”Performance Analysis of AES Candidates on the 6805 CPU Core”, http://www.ozemail.com.au / geoffk/aes-6805.

[33] B. Weeks, M. Bean, T. Rozylowicz and C. Ficke, ”Hardware Performance Simulations of Round 2 Advanced Encryption Standard Algorithms”. *3rd AES Conference*, pp. 286-304, April 2000.

[34] Peter J. Ashenden ”The Designer's Guide to VHDL”. *2nd Edition, San Francisco, CA, Morgan Kaufmann*, 2002.