Journal of Engineering and Applied Sciences 13 (21): 9269-9276, 2018
ISSN: 1816-949X

New Data Security Method Based on Biometrics

[1]Sally Antoin Jerjees and [2]Tarik Zeyad Ismaeel
[1]Department of Computer Engineering,
[2]Department of Electrical Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq

Abstract: Merging biometrics with cryptography has become more familiar and a great scientific field was born for researchers. Biometrics adds distinctive property to the security systems, due biometrics is unique and individual features for every person. In this study, a new method is presented for ciphering data based on fingerprint features. This research is done by addressing plaintext message based on positions of extracted minutiae from fingerprint into a generated random text file regardless the size of data. The proposed method can be explained in three scenarios. In the first scenario the message was used inside random text directly at positions of minutiae in the second scenario the message was encrypted with a choosen word before ciphering inside random text. In the third scenario the encryption process insures a correct restoration of original message. Experimental results show that the proposed cryptosystem works well and secure due to the huge number of fingerprints may be used by attacker to attempt message extraction where all fingerprints but one will give incorrect results and the message will not represent original plain-text, also this method ensures that any intended tamper or simple damage will be discovered due to failure in extracting proper message even if the correct fingerprint are used.

Key words: Cryptosystem, ciphering, fingerprint minutiae and random text, represent original, discovered, extracting proper

INTRODUCTION

Enumerous types of data security methods were invented since ancient civilization of Greece, Rome and Egypt. These methods were varied and sequenced with time from simple to complex. Two well-known categories of cryptography are commonly used these are symmetric and asymmetric encryption algorithms. In symmetric a secret key is used to convert message (plaintext) to unreadable message (cipher text). While asymmetric algorithm uses two keys, public key which is announced to every person used in the encryption and the other is the secret key which is used for decryption and derived from the public key which is known by recipient only. The algorithms are different, each has its own characteristics, the strength of each encryption depends upon the key management, type of cryptography, number of keys and number of bits used in a key (Bhanot and Hans, 2015). Experimental results of Singh and Supriya (2013) are done to show that Advanced Encryption Standard (AES) is most efficient in terms of speed, time, throughput and avalanche effect. By Soni *et al*. (2012) a comparison is done for Data Encryption Standard (DES) and AES algorithms, based on the text files used and the experimental result. It was inferred that AES algorithm consumes less encryption and decryption time as compared to DES algorithm. Athena and Sumathy (2017) presented a brief review of major cryptographic techniques such as Rivest Shamir and Adleman (RSA), and Dffie Hellman and the Elliptic Curve Cryptography (ECC). The results of these approaches were compared in terms of key size, key generation time, signature generation time and signature verification time. The initialization of random prime numbers and the key computation based on the points on the elliptic curve assured the high-security compared to the existing schemes with the minimum time consumption and sizes in cloud-based applications. By Sony *et al*. (2015) a new technique is deployed to improve asymmetric key encryption techniques by using multiple keys and Chinese Remainder Theorem (CRT) with original RSA algorithm. A comparison was done among original RSA and proposed RSA with CRT and modified RSA with respect to computational time to prove that their method was good when security and less computational time requirements were needed together. Now, encryption of data exceeds traditional methods and new innovations were added to the cryptography an example is Biometric technologies which promise many benefits, including stron ger user authentication and security improvements.

By Cavoukian and Stoianov (2007), the researchers focused on the privacy and security advantages of Biometric Encryption (BE). By considering the merits of

Corresponding Author: Sally Antoin Jerjees, Department of Computer Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq

Biometric Encryption for verifying identity, protecting privacy and ensuring security. Biometrics is individual elements which can be physical or behavioral measured to check personality. Encryption based biometric was used with traditional security algorithms to enhanced security due to increasing the use of images in various fields (Saranya and Prabhu, 2016). It became more convenient to meet cryptographic requirements using fingerprint image to generate a crypto-biometric key (Bhagya and Mahesh, 2014).

Minutiae extraction of fingerprint: Biometrics is classified into physical and behavioral. Physical biometrics includes (Fingerprint, Iris and Facial) and behavioral biometrics includes (Signature, Voice and Handwriting). The oldest and most common types of biometrics are fingerprints which are classified into three levels the first level depends on the shape and they are (whorls, loops and arches) and the second level represent by details called minutiae and the third level which depends on precision of the scanners is pore (Maddala and Tangellapally, 2010). Most systems that use fingerprint depend on minutiae of the fingerprint image Minutiae are natural and distinctive properties of human. The popular minutiae types used are ridge bifurcation and ridge ending (Shrivastava and Sharma, 2012). The other types of minutiae are combinations of both of them (Chaudhari *et al*., 2014) as shown in Fig. 1.

The fingerprint minutiae are found at the feature extraction stage of image processing. Numerous minutiae are always being found due to the noise. The minutiae are easily being found when thinned image become ready to use where endings can be found at termination points of thin lines but bifurcations are found at the intersection of three thin lines (Chaudhari *et al*., 2014). Many techniques were invented to use fingerprint image in different applications such as verification and identification (Kaur *et al*., 2008). A study was done to present a comparison of classifications of minutiae extraction techniques (Bansal *et al*., 2011). The image must pass through many processes before it can be valid to be used, so, Afsar *et al*. (2004) was proposed a system using fingerprint image and explained the stages of processing that image to get detailed features for its importance in providing high degree of security in verification and identification systems. More than one type of biological features can be used to enhance security (Jagadeesan and Duraiswamy, 2010) proposed a cryptographic key of 256-bit from (Iris and fingerprint) templates to prepare image before using it. Several steps must be done on gray image to convert it to binary image that shown in Fig. 2ab represent the thinned image after using morphological method in MATLAB. Crossing Number concept (CN) (Sudiro and Yuwono, 2012) will be used to extract minutiae

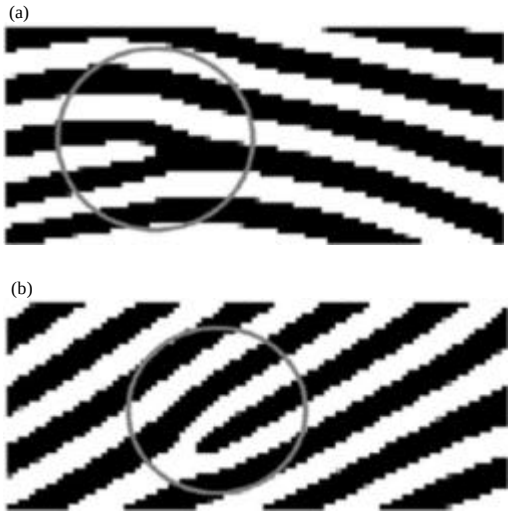

Fig. 1: Coventional types of minutiae: a) Ending ridge and b) Bifurcation ridge

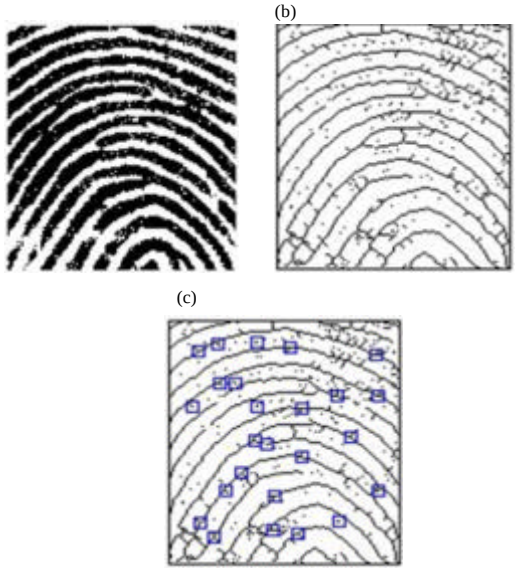

Fig. 2: Steps for minutiae extraction: a) Binary image; b) Thinned image and c) Minutiae of image

from the thinned image as in Fig. 2c. The concept is performed by comparing pixel of thinned image with eight neighboring pixels by a window of 3×3 where value of p equals to one and value of p_i is either 0 or 1, the following equation will be applied to find CN:

$$CN = \frac{1}{2}\sum_{i=1}^{8}|p_i - p_{i+1}| \quad (1)$$

If CN = 1Then it is ridge ending but if it is equal to 3 then it is bifurcation.

MATERIALS AND METHODS

Proposed security system: Feature extraction process of fingerprint image outcomes an important informaton can be used in verification and cryptography purposes (Daramola *et al*., 2013). Most of those features are used as a key for standard encryption algoritms (Fatangare and Honwadkar, 2011). The proposed method suggests to benefit from these features neither for key generation nor creation of algorithm but as effective particularity for ciphering. The proposed method can be carried out in three scenarios as:

Scenario 1: The proposed cryptosystem in scenario 1 enhance security level where data and key size are not a matter. The scenario is demonstrated in the following steps:

Step 1: Enter data message such as (Good morning).

Step 2: Add fixed word at end of message such as (eof) which is known to both sides, transmitter and recipient, this suffix increases authentication and prevention damage of the text. So, the message becomes: (Good morningeof).

Step 3: Extract (ending and bifurcation) minutiae from fingerprint, the position of these minutiae will be used as shown in Table 1.

Step 4: Generate random text used into cryptosystem to carry the message through its contents the size of the text must equal to the size of used fingerprint image.

Step 5: Insert the message from step 2 into generated random text at position of minutiae points from the step 3 in the same order as in the Table 1 at this point text handles message and it is ready to be sent to the recipient. Figure 3 shows the text that contains message.

Step 6: At receiver side, the recipient has a same copy of fingerprint and fixed word that was added to the message which is used to find location of included message inside text that he received.

Scenario 2: In scenario 1 the message was readable this mean if the attacker has the right fingerprint, the message can be extracted with several attempts to specify the number of used minutiae as well as their order in ciphering. So, additional security stage can be used to terminate any opportunity for reconstracting message by unauthorized person as in the following steps:

Step 1: Enter data message such as (Good morning).

Table 1: Minutiae point's positions

Number of minutiae points	X position of minutiae	Y position of minutiae
1	49	13
2	36	35
3	13	46
4	49	46
5	70	51
6	113	88
7	18	16
8	114	17
9	122	24
10	14	26
11	36	27
12	96	30
13	86	38
14	68	45
15	118	54
16	99	55
17	16	63
18	120	67
19	50	69
20	77	69
21	43	87
22	66	94
23	20	107
24	43	108
25	96	108

Table 2: Bit by bit XORing between the letters of key and message

Letters message	Binary representation	Letters of key	Binary representation	XOR result	Result in hexa
G	01000111	s	01110011	00110100	34
o	01101111	a	01100001	00001110	0E
o	01101111	l	01101100	00000011	03
d	01100100	l	01101100	00001000	08
Space	00100000	y	01111001	01011001	59
m	01101101	s	01110011	00011110	1E
o	01101111	a	01100001	00001110	0E
r	01110010	l	01101100	00011110	1E
n	01101110	l	01101100	00000010	02
i	01101001	y	01111001	00010000	10
n	01101110	s	01110011	00011101	1D
g	01100111	a	01100001	00000110	06
e	01100101	l	01101100	00001001	09
o	01101111	l	01101100	00000011	03
f	01100110	y	01111001	00011111	1F

Step 2: Add fixed word at end of message such as (eof) for same reason mensioned in scenario 1 step 2.

Step 3: Encrypt message from previous step by XORing it with key like (sally) bit by bit order to become unreadable the hexa values for this result is (34 0E 03 08 59 1E 0E 1E 02 10 1D 06 09 03 1F) due to it represents unseen symbols if its converted to character type as shown in Table 2.

Step 4: Extract (ending and bifurcation) minutiae from fingerprint, the position of these minutiae will be used as shown in Table 1.

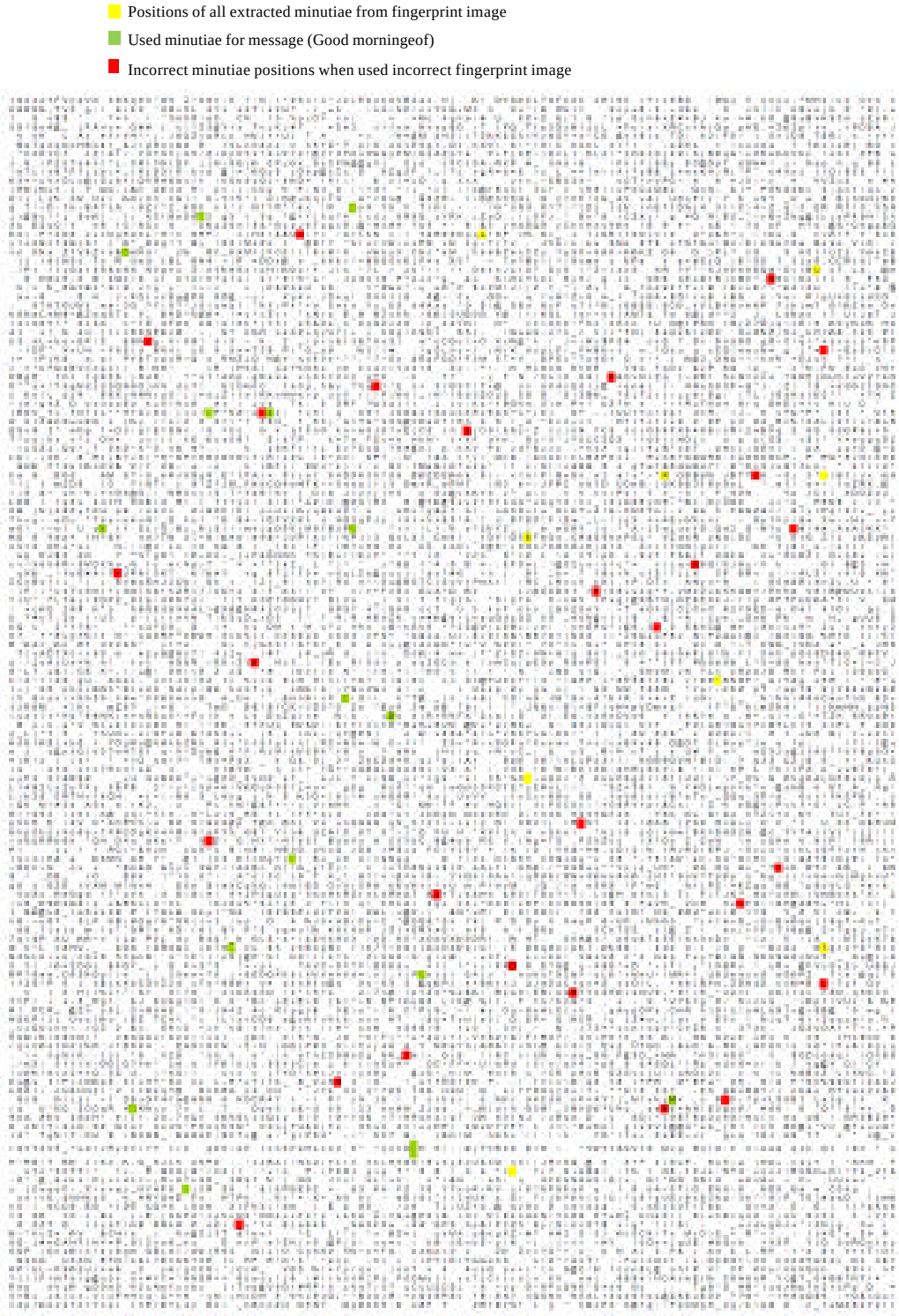

Fig. 3: Position of minutiae on the random text for scenario (1)

Step 5: Insert the message from step 3 into generated random text at position of minutiae points from previous step at this point text handle ciphered message and it is ready to be sent to the recipient as shown in Fig. 4.

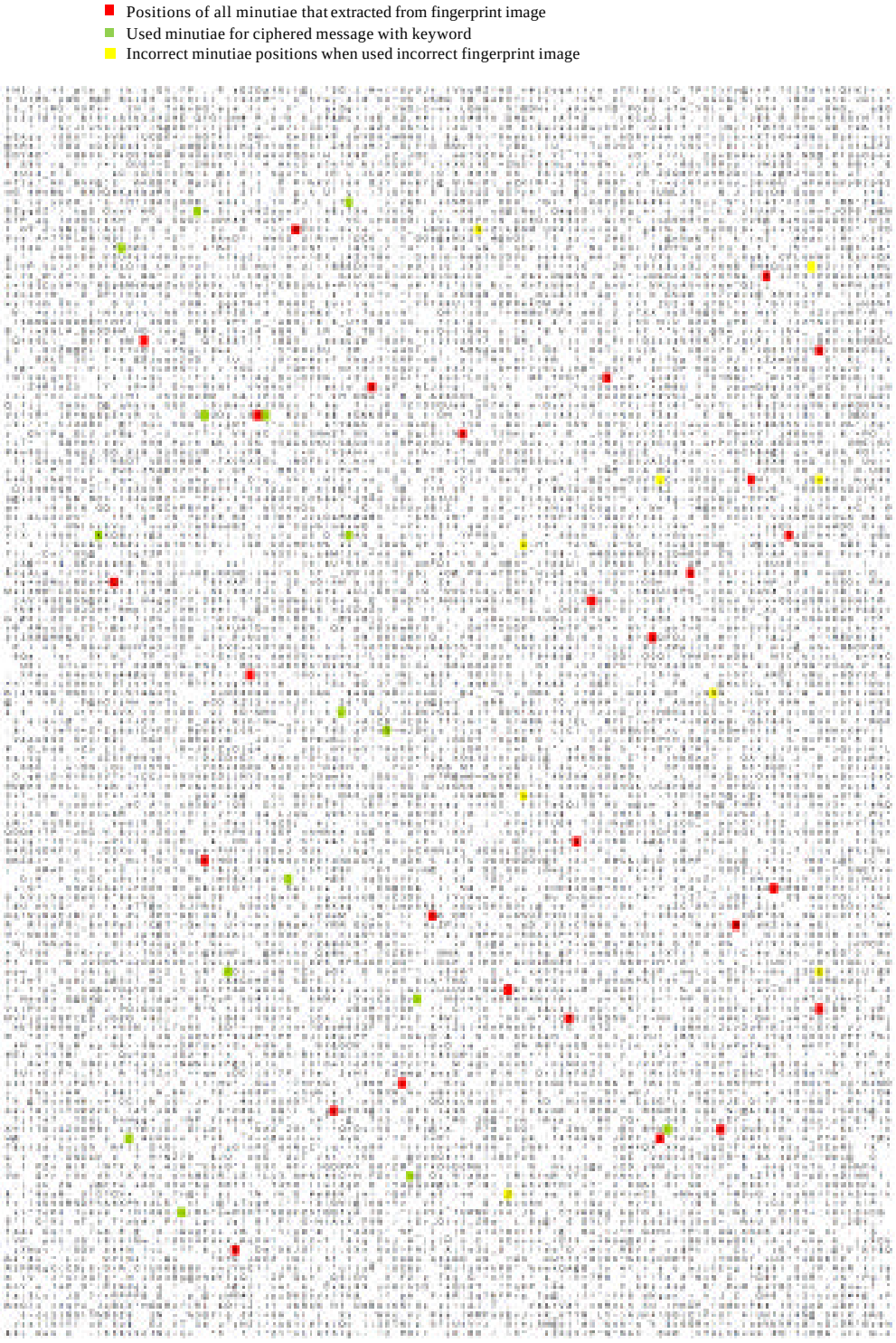

Fig. 4: Position of minutiae on the random text for scenario (2)

Step 6: At receiver side, the recipient has a same copy of fingerprint which is used to find location of ciphered encrypted message inside text that he received and then decrypt the message with same key word to retrieve original message.

Scenario 3: In some cases the aim of attacker is to prevent recipient from decipher the right message, so, opponent may replace some letters of the ciphertext and wrong information will be received. Scenario 3 suggest a new method to withstand this type of attacks as in the following steps:

Step 1: Enter data message such as (Good morning).

Step 2: Add fixed word at end of message such as (eof) and it will be (Good morning eof) for same reason mensioned in scenario 1 step 2.

Step 3: Encryption in this scenario do not use any key, message itself become a key after cyclic shift where the message will be (fGood morningeo) then XORing the new form with previous one bit by bit in order to become (21, 28, 00, 0B, 44, 4D, 02, 1D, 1C, 07, 07, 09, 02, 0A, 09) in hexadecimal form due to it represents unseen symbols if its converted to character type.

Step 4: Extract (ending and bifurcation) minutiae from fingerprint, the position of these minutiae will be used as shown in Table 2.

Step 5: Insert the message from step 3 into generated random text at position of minutiae points from the previous step at this point text handle ciphered message and it is ready to be sent to the recipient as shown in Fig. 5.

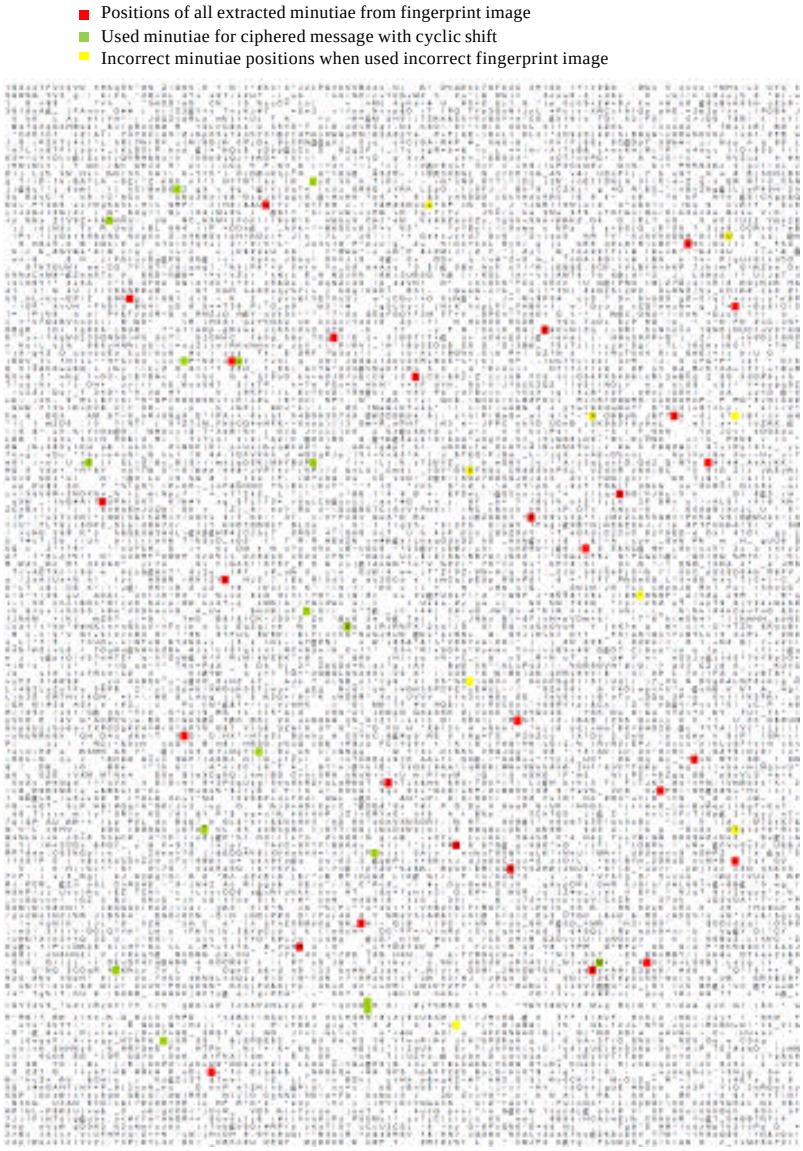

Fig. 5: Position of minutiae on the random text for scenario (3)

Step 6: At receiver side, the recipient has the same copy of fingerprint which is used to find location of ciphered encrypted message inside text that received and then decrypt the message to retrieve original message by XORing first letter of ciphered extracted message with letter f the resultant letter will be XORed with next letter of ciphered message as shown:

RESULTS AND DISCUSSION

The random text generated has the same size of fingerprint image using MATLAB 2017b. In the first scenario, the message inserted in the text directly after adding (eof) as shown in Fig. 3 in position of minutiae listed in the Table 2. In the second scenario the message encrypted with key word (sally) repeated as long as message and the encrypted form inserted in the text as shown in the Fig. 4. In the third scenario there is no key used, the message with its cyclic shift form will XORed together and the result will be inserted in the text as shown in the Fig. 5. When recipient received random text the same algorithm for all scenarios will be applied in reverse order to obtain original message which is (Good morning). When unauthorized person estimates the algorithm and tries another fingerprint (not fingerprint used in ciphering) the incorrect result will appear as shown in Fig. 3, Table 3 reviewed faulty messages that will be extracted when using incorrect fingerprint for all scenarios.

When recipient was using the proper fingerprint on a damaged or tampered text incorrect message will be extracted which does not represent plaintext in scenarios 1 and 2 a damage was happened only on the letters at minutiae positions during tampering but in scenario 3 if tampering infect one letter that was positioned by minutiae all subsequent letters were destroyed, big tampering increase the probability that letter (G) distorted, so that, the message will completely break. Table 4 showed the extracted messages for different amount of noise and different letters were destroyed.

The strength of any security system is experienced by type of attacker who can break the system either by estimating the key or the encryption process in our proposed system, the key is fingerprint, so that, there is huge number of them but we suppose that the attacker must know the fingerprint, the number of minutiae and their types and order of them to start his attack. So that, we can calculate the brute force attack to estimate the time needed by the attacker to break the system. We assume

Table 3: Reconstructed messages when incorrect fingerprint used for deciphering

Scenario	Reconstructed message
1	'A)qFyNf MNGEPJH 8/,t*krn!'
2	'R5!X4L \S.VLJV@VC8'
3	'F-d-R Ri6X7Ç6hFh8sS2DwWw'

Table 4: Destroying letters when random noise added to the text of scenario 3

Destroyed letter	Reconstructed message
Letter (n)	'Good mor fFpP#\('
Letter (d)	Goo o"QLLPWP#PZs6 Rx-^'
Letter (o)	'G4GEGZz z zs 6dN = -I'
Letter (G)	'f f f f} u uC0b D7D'

that the attacker has a device can perform 2^{56} decryption per second and the average number of minutiae of fingerprint is 50 as in the following equation:

$$BFT = {}^{(\text{Number of minutiae})}\!/_{2^{56}} \text{ second}$$

The time needed by the brute force attacker is $1.338407952246545132504908051458 \times e^{+40}$ years. This time is just to obtain unreadable message because of XORing with another sequence as in scenarios 2 and 3.

CONCLUSION

The simulation results showed that the proposed method enhanced the security. The huge number of fingerprint minutiae added strength to the proposed method. Mixing more than one type of minutiae and with the use of different order of them can add more confusion and diffusion to the cryptosystem. Scenario 2 was introduced when scenario 1 cannot with stand when the correct fingerprint is discovered. In the second scenario there is more diffusion of message inside random text due to encryption of message where any traditional encryption method can be used. The random text used to handle message differs every time the sender tries to send a new message this gives strength to the system because if the same text is sent twice the opponent can compare two copies and extract the message. In scenario 3 when some attackers tampered with text for some reasons, recipient can discover this damage when original message cannot be obtained from true fingerprint then the message is discarded and the sender is indicated to resend message.

REFERENCES

Afsar, F.A., M. Arif and M. Hussain, 2004. Fingerprint verification system using minutiae matching. National Conf. Emerging Technol., 2004: 141-146.

Athena, J. and V. Sumathy, 2017. Survey on public key cryptography scheme for securing data in cloud computing. Circuits Syst., 8: 77-92.

Bansal, R., P. Sehgal and P. Bedi, 2011. Minutiae extraction from fingerprint images: A review. Int. J. Comput. Sci., 8: 74-85.

Bhagya, P. and P.K. Mahesh, 2016. Generating an efficient crypto-biometric key using fingerprint and fuzzy vault. Imperial J. Interdisciplinary Res., 2: 442-448.

Bhanot, R. and R. Hans, 2015. A review and comparative analysis of various encryption algorithms. Intl. J. Secur. Appl., 9: 289-306.

Cavoukian, A. and A. Stoianov, 2007. Biometric encryption: Technology for strong authentication, security and privacy. Biometric Technol. Today, 15: 1-11.

Chaudhari, A.S., G.K. Patnaik and S.S. Patil, 2014. Implementation of minutiae based fingerprint identification system using crossing number concept. Inf. Econ., 18: 17-26.

Daramola, S.A., T. Sokunbi and A.U. Adoghe, 2013. Fingerprint verification system using support vector machine. Intl. J. Comput. Sci. Eng., 5: 678-683.

Fatangare, M. and K.N. Honwadkar, 2011. A biometric solution to cryptographic key management problem using Iris based fuzzy vault. Intl. J. Comput. Appl., 15: 42-46.

Jagadeesan, A. and K. Duraiswamy, 2010. Secured cryptographic key generation from multimodal biometrics: Feature level fusion of fingerprint and iris. Int. J. Comput. Sci. Inform. Secur., 7: 296-305.

Kaur, M., M. Singh, A. Girdhar and P.S. Sandhu, 2008. Fingerprint verification system using minutiae extraction technique. World Acad. Sci. Eng. Technol. Intl. J. Comput. Inf. Eng., 2: 3405-3410.

Maddala, S. and S.R. Tangellapally, 2010. Implementation and evaluation of INST biometic image software for fingerprint recognition. MSc Thesis, Blekinge Institute of Technology, Karlskrona, Sweden.

Saranya, R. and S. Prabhu, 2016. Image encryption using RSA algorithm with biometric recognition. Intl. J. Eng. Comput. Sci., 5: 19149-19154.

Shrivastava, V. and S. Sharma, 2012. Data compression of fingerprint minutiae. Intl. J. Eng. Sci. Technol., 4: 401-405.

Singh, G., 2013. A study of encryption algorithms (RSA, DES, 3DES and AES) for information security. Intl. J. Comput. Appl., 67: 33-38.

Soni, S., H. Agrawal and M. Sharma, 2012. Analysis and comparison between AES and DES cryptographic algorithm. Intl. J. Eng. Innov. Technol., 2: 362-365.

Sony, K., D. Shaik, B.D. Sri and G. Anitha, 2015. Improvised asymmetric key encryption algorithm using MATLAB. IOSR. J. Electron. Commun. Eng., 10: 31-36.

Sudiro, S.A. and R.T. Yuwono, 2012. Adaptable fingerprint minutiae extraction algorithm based-on crossing number method for hardware implementation using FPGA device. Intl. J. Comput. Sci., Eng. Inf. Technol., 2: 1-30.